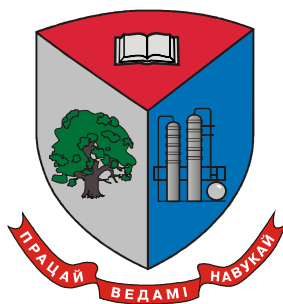


МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ»



**70-я НАУЧНО-ТЕХНИЧЕСКАЯ
КОНФЕРЕНЦИЯ УЧАЩИХСЯ,
СТУДЕНТОВ И МАГИСТРАНТОВ**

15–20 апреля 2019 г.

СБОРНИК НАУЧНЫХ РАБОТ

В 4-х частях

Часть 4

Минск 2019

УДК 005.745:378.6](476)(06)

ББК 66.75

70-я научно-техническая конференция учащихся, студентов и магистрантов: сб. науч. работ : в 4-х ч. – Минск, 15–20 апреля 2019 г. [Электронный ресурс] – Минск : БГТУ, 2019. – Ч. 4. – 496 с.

Сборник научных работ студентов и магистрантов университета составлен по итогам 70-й студенческой научно-технической конференции Белорусского государственного технологического университета, проведенной с 15 по 20 апреля 2019 г. На конференции было заслушано 1826 докладов, лучшие из них представлены в данном сборнике. Материалы содержат результаты научных исследований студентов, посвященные актуальным вопросам лесоводства, лесоустройства, лесозащиты, технологии и техники лесной промышленности, химической технологии и техники, неорганических веществ, вяжущих материалов, технологии стекла и керамики, технологии электрохимических производств, экологии, синтеза новых органических и неорганических материалов, экономики, издательского дела и полиграфии, информационных технологий. Также представлены доклады, посвященные энергосбережению, безопасности технологических процессов и производств, контролю качества и безопасности веществ, материалов и изделий.

Сборник предназначен для использования специалистами соответствующих отраслей народного хозяйства, научными работниками и преподавателями, а также студентами и магистрантами.

Рецензенты:

декан факультета ИТ, доцент, канд. техн. наук
зам. декана ИЭФ, доцент, канд. техн. наук

Д.В. Шиман
В.П. Демидовец

Редакционная коллегия:

зав. кафедрой ИиВД, канд. техн. наук
зав. кафедрой физики, д-р физ.-мат. наук
доц. кафедры ИБиП, канд. историч. наук
зав. кафедрой ПИ, доц., канд. техн. наук
зав. кафедрой ЭиУП, канд. эконом. наук
доц. кафедры ВМ, канд. физ.-мат. наук
методист учебно-методического сектора

Д.М. Романенко
Н.Н. Крук
П.С. Крючек
Н.В. Пацей
А.В. Ледницкий
Л.Д. Яроцкая
О.И. Ставровская

© УО «Белорусский государственный
технологический университет», 2019

ОРГАНИЗАЦИОННЫЙ КОМИТЕТ

Дормешкин О.Б.	проректор по научной работе (председатель)
Сакович А.А.	проректор по учебной работе (зам. председателя)
Каврус И.В.	начальник НИЧ
Черник Е.О.	зав. сектором ИВОНД
Сильванович А.В.	инженер сектора ИВОНД
Шиман Д.В.	ответственный за НИРС факультета ЛХ, канд. с/х. наук, доцент кафедры лесоводства
Симанович В. А.	ответственный за НИРС факультета ТТЛП, канд. техн. наук, доцент кафедры лесных машин, дорог и технологий лесопромышленного производства
Стасевич О.В.	ответственная за НИРС факультета ТОВ, доцент кафедры физико-химических методов сертификации продукции
Залыгина О.С.	ответственная за НИРС факультета ХТиТ, канд. техн. наук, доцент кафедры промышленной экологии
Усевич В.А.	ответственная за НИРС факультета ИЭ, старший преподаватель кафедры экономической теории и маркетинга
Медяк Д.М.	ответственная за НИРС факультета ПиМ, канд. техн. наук, доцент кафедры полиграфических производств
Яроцкая Л.Д.	ответственная за НИРС факультета ИТ, канд. физ.-мат. наук, доцент кафедры высшей математики
Тишкевич И.А.	методист учебно-методического отдела, ответственная по работе с колледжами

**Секция
ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

**ОСОБЕННОСТИ РАЗРАБОТКИ ИНТЕРАКТИВНОГО
УЧЕБНИКА ПО ДИСЦИПЛИНЕ
«ОБРАБОТКА ИЗОБРАЗИТЕЛЬНОЙ ИНФОРМАЦИИ.
РАЗДЕЛ «КОРРЕКЦИЯ ИЗОБРАЖЕНИЙ»»**

Сегодня электронная книга приходит на смену бумажной. Новые технологии дают новые возможности. Электронный учебник – это учебное издание в электронном виде, которое содержит структурированный и систематизированный материал, используемый студентами в учебном процессе для освоения новых знаний и умений [1]. Электронный учебник имеет ряд преимуществ по сравнению с печатным аналогом:

- простота и удобство обращения;
- возможность обновления ресурса электронного учебника;
- автоматизация учебного процесса;
- полнота передаваемой информации.

Для представления учебного контента в виде электронного интерактивного учебника существует множество приложений, большинство из которых доступны по лицензии типа opensource. В таблице представлены некоторые приложения, которые могут быть использованы для создания интерактивных учебников [2-4].

Таблица – Приложения для разработки электронных книг

Название приложения	Операционная система, под управлением которой работает приложение	Разработчик	URL для доступа к приложению
iBooks Author	Mac OS iOS	Apple	www.apple.com/ibooks-author
Course Builder	Android iOS	Google	code.google.com/p/course-builder
Magic Premium Author	Любая	Samsung	magicinfoshop.com/ru/soft
Kindle Textbook Creator	Любая	Amazon	www.amazon.com/gp/feature.html?ie=UTF8&docId=1002998671

Каждое из представленных в таблице приложений позволяет работать с несколькими интерактивными элементами.

Однако приложение iBooksAuthor обладает наибольшим функционалом для разработки электронных учебников: управление контен-

том при помощи тезауруса/словаря; представление содержимого в виде текста, видео, диаграмм, галереи графических рисунков; размещение 3D-моделей, java-тегов; поддержка тестирования и др.

Программа iBooks Author решает множество задач, требуя от автора книги минимальных общекомпьютерных знаний. Преимуществом разработки на базе данной технологии является наличие бесплатной и интуитивно понятной среды разработки учебника: преподаватель сам может создать учебник без привлечения специалистов [2].

iBooks Author помогает автору не только создать книгу, но и самостоятельно опубликовать её в Интернете. Учебник можно опубликовать в интернет-магазине iTunesStore в разделе AppStore (в открытом или платном доступе). Второй вариант размещения файлов типа .ibooks – самостоятельно, например, при помощи электронной почты для дальнейшей загрузки на iPad с помощью iTunes. В этом случае, тестируемая книга отображается в списке книг приложения iBooks с пометкой Proof.

Программа совместима с файлами из офисных пакетов Apple и Microsoft и поддерживает виджеты на JavaScript и HTML5. Книга создается на основе встроенных шаблонов iBooks Author. Шаблоны для книг можно создавать и самостоятельно. Созданную книгу можно сохранить как файл формата *.iba для дальнейшей работы и редактирования. Законченную книгу можно экспортировать в формат *.ibooks для чтения iPad. При желании можно экспортировать книгу в формат *.pdf или *.txt.

После первого запуска мы увидим интерфейс, в котором можно выбрать любой из предложенных шаблонов, или открыть новый проект книги. Нам предлагают взять один из предложенных шаблонов, так как в нем мы сможем изменить все, что сочтете нужным. Процесс создания книги довольно прост: вам нужно лишь переместить различные объекты к нужным частям страницы.

Для создания интерактивных элементов на страницу нужно поместить один из так называемых виджетов (Widgets), которые вызываются кнопкой меню на верхней панели. Существует несколько типов виджетов:

Галерея (Gallery), Медиабраузер (Media), Упражнение (Review), Интерактивное изображение (Interactive Image), Трехмерный объект (3D), Презентация (Keynote), Прокручивающееся боковое меню, Всплывающее окно (Pop-Over), HTML

В процессе работы над книгой, автор сможет проверить, насколько ее внешний вид соответствует его замыслам. Для этого нужно подключить к компьютеру Mac планшет iPad, запустить на нем программу iBooks, а затем в программе iBooks Author нажать кнопку Preview (просмотр). Программа автоматически распознает подклю-

ченный планшет и загрузит на него текущую версию вашей книги. После этого Вы сможете оценить внешний вид Вашей книги и проверить работу всех интерактивных элементов.

После этого ее можно опубликовать через iTunes Producer прямо в iBookstore, сделав ее доступной для широкой публики. Если интерактивность в книге не обязательна, после верстки ее можно экспортировать в PDF и распространять в печатном виде. Ограничение на размер публикуемой книги составляет 2 ГБ, но Apple рекомендует не делать электронных книг размером больше 1ГБ и отмечает, что у читателей в 3G-сетях могут быть ограничения на загрузку, не больше 20МБ.

Цель проекта – создание уникального учебного пособия по дисциплине «Обработка изобразительной информации» Раздел «Коррекция изображения», в электронном интерактивном варианте для использования студентами, учащими на специальности «Дизайн электронных и веб-изданий» для подготовки к практическим занятиям, зачетам и экзаменам. А также для повышения интереса учащихся в изучении данной дисциплины.

Для разработки электронного учебника используется программа iBooks и будут использоваться следующие виджеты:

1. Галерея (Gallery)

Интерактивная галерея фотографий или изображений, которые можно будет просматривать в окне на определенной странице книги. Вместо того, чтобы просматривать по одному изображению на странице, читатель может пролистывать коллекцию интерактивных фотографий.

2. Медиабраузер (Media)

С помощью этого виджета на страницу книги можно поместить видео- или аудиофрагменты. Вместо обычного чтения и просмотра изображений читатель может познакомиться с содержанием вживую. Видеоролики могут проигрываться автоматически, и любой кадр в iBooks Author можно выбрать в качестве кадра-постера.

3. Упражнение (Review)

С помощью такого виджета удобно проверять усвоение учебного материала: фактически он представляет собой интерактивное тестовое задание в стиле «вопрос-ответ» с определенным количеством возможных ответов (рисунок 1). Автор может предложить несколько вариантов ответа на каждый вопрос.

4. Интерактивное изображение (Interactive Image)

Любая картинка намного интереснее, когда она интерактивна. Этот виджет поможет «оживить» иллюстрации – добавить к ним подпись, а также использовать автоматическое увеличение и уменьшение, позволяющее сделать акцент на нужном фрагменте (рисунок 3). Вы-

носки, инструменты панорамирования и изменения масштаба делают чтение гораздо увлекательнее.

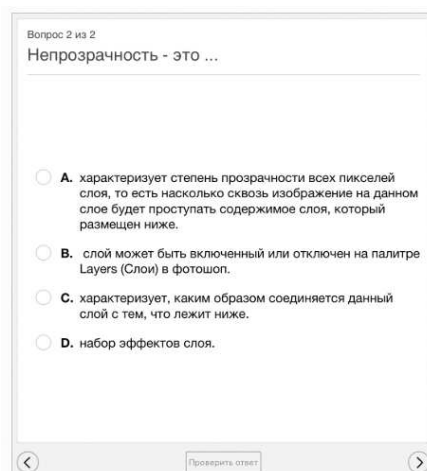


Рисунок 1 – Пример виджета «Упражнение»

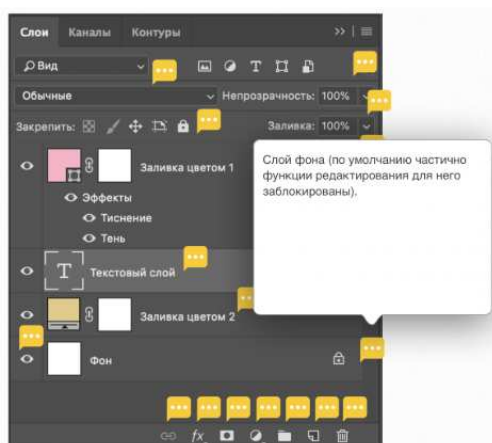


Рисунок 2– Пример виджета «Интерактивное изображение» в учебнике

4. HTML

Создавайте собственные HTML-виджеты, используя ваш любимый редактор текста или приложение сторонних разработчиков, и добавляйте их в книгу вместе с текстом. Динамически обновляемые веб-данные поддерживают актуальные примеры.

5. Всплывающее окно (Pop-Over)

Изображение может заменить тысячу слов, но иногда к нему стоит добавить пару фраз. Окна Pop-Over позволяют читателям, коснувшись изображения, отобразить окно с дополнительной информацией, изображениями или другими данными для более полного контекста.

Прототип разрабатываемого учебника «Обработка изобразительной информации. Раздел «коррекция изображений»» представлен на рисунке 3. Весь материал разбивается на 5 глав, в которых используются авторские видеоролики, 3D-модели, авторские изображения с интерактивным описанием элементов обработки и т.д.

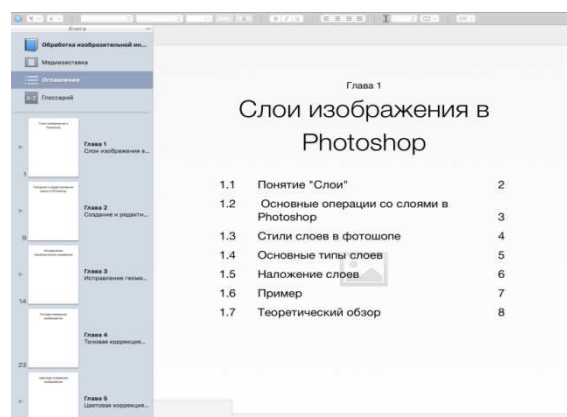


Рисунок 3 – Прототип учебника

В заключении, можно сказать, что применение электронного учебника усиливает активность каждого студента, повышает его заинтересованность в изучении материала. Самостоятельная работа с электронным учебником или работа в учебной аудитории позволяет научить студента ориентироваться в информационном пространстве, изучить новый материал и находить решение поставленным задачам. Использование современных методов обучения студентов в вузах, а именно – электронных учебников, улучшит качество знаний, ускорит процесс получение информации. Рациональное применение электронных средств образования при изучении дисциплины «Обработка изобразительной информации» будет способствовать увеличению интереса к предмету, позволит студентам лучше усвоить данный материал и повысит мотивацию к получению новых навыков и знаний.

ЛИТЕРАТУРА

- 1.Семенова Н.Г. Теоретические основы создания и применения мультимедийных обучающих систем лекционных курсов электротехнических дисциплин: монография. Дата доступа: 28.03.2019.
- 2.iBooksAuthor [Электронный ресурс]. – Режим доступа: <https://www.apple.com/ru/ibooks-author/> – Дата доступа: 28.03.2019
- 3.MagicInfo Premium Author [Электронный ресурс]. – Режим доступа: <http://www.magicinfosamsung.ru/authorI/> – Дата доступа: 28.03.2019
4. Kindle Textbook Creator [Электронный ресурс]. – Режим доступа: <https://in30minutes.com/kindle-textbook-creator-vs-ibooks-author-no-contest-yet/> – Дата доступа: 28.03.2019.

ГЕНЕРИРОВАНИЕ ФИКТИВНЫХ ДАННЫХ ПРИ ПОМОЩИ FAKER.JS

Для тестирования приложений, использующих определенные наборы данных, требуется формирование наборов реальных или фиктивных данных.

Фиктивные данные – это данные, которые содержат информацию, не соответствующую действительности.

Для удобства тестирования веб-интерфейса используется библиотека `Faker.js`.

`Faker` – это библиотека, которая позволяет генерировать случайные данные. С ее помощью можно заполнить таблицы в базе данных, построить корректные XML-документы, сформировать JSON-ответы для REST.

Особенности:

- наличие множества групп форматированных данных;
- поддержка возврата локализованных данных, отличных от `en_US`;
- возможность работы с готовыми наборами данных;
- возможность создавать собственные шаблоны данных
- возможность загрузки `faker.js` из CDN;
- наличие обертки для работы с CLI.

Подключить библиотеку возможно несколькими путями. Первое, используя «Node.js», что предоставляет возможность взаимодействия с консолью и с браузером. Для работы через «Node» необходимо в консоль ввести следующие команды:

```
npm init
npm install faker -s
nodeapp
```

В файле `app.js`. (рисунок 1), приводится код инициализации библиотеки. После этого останется создать переменные, которым и присваиваются значения из библиотеки.

```
//npm init
//npm install faker -s
//node app
var faker = require('faker');
var randomName = faker.name.findName();

var http = require('http');
const PORT = 3000;
```

Рисунок 1– Работа с Node.js

Второе, подключение через сетевую инфраструктуру «CDN». Для этого необходимо подключить ссылку (рисунок 2) в скрипте JavaScript. Документация по работе с библиотекой Faker находится по адресу «<https://github.com/marak/Faker.js>». Библиотека располагает богатым набором методов для генерации случайных данных. Например, она содержит набор следующих полей: name, address, phone, internet, company, image, date, random и т.д. Каждый из пунктов содержит свои подпункты, которые и являются методами извлекаемых данных.

```
<!-- FAKER.JS -->  
<script src="https://cdnjs.cloudflare.com/ajax/libs/Faker/3.1.0/faker.js"></script>
```

Рисунок 2 – Подключение через CDN

Для демонстрации возможности «Faker.js» создается форма вида (рисунок 3). После чего необходимо написать код, обращение к элементам формы может происходить как при помощи JS, так и при помощи «jQuery».



Рисунок 3– Форма с использованием библиотеки «Faker.js»

На рисунке 4 представлена функция генерации псевдослучайных данных, для формы, где значениям блоков присваиваются данные библиотеки, после чего данная функция выполняется при нажатии на кнопку «GenerateNew».

Данная форма является тестовым объектом, содержащим несколько полей, как текстовых, так и графических, в блоке «locality» демонстрируется работа локализации библиотеки, в результате чего генерация данных меняется относительно выбранного языка (рисунок 5).

```
function generateName () {  
  var email= "email: " + faker.internet.email();  
  var jobTitle = "jobTitle: " + faker.name.jobTitle();  
  var fullName = faker.name.findName();  
  var phoneNumber = "phone: " + faker.phone.phoneNumber();  
  
  var background = 'url("' + faker.image.nature() + "')';  
  $('<div class="cardheader">').css({'background': background,  
    'background-size': 'cover', 'height': '135px'});  
  
  document.getElementById('name').innerHTML=email;  
  document.getElementById('fullName').innerHTML = fullName;  
  document.getElementById('jobTitle').innerHTML = jobTitle;  
  document.getElementById('phoneNumber').innerHTML = phoneNumber;  
  
  $('#imageAvatar').attr("src", faker.internet.avatar());  
}
```

Рисунок 4 – Демонстрация функции генерации данных

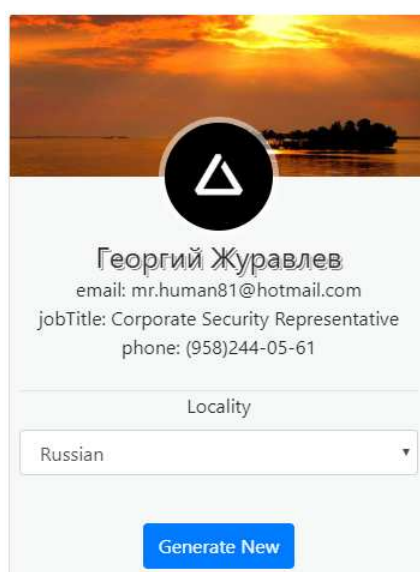


Рисунок 5 – Демонстрация функции локализации данных формы

Создание формы и вывод значений демонстрирует лишь малый аспект задач, решаемых при помощи faker.js. Например, существует возможность работы при помощи РНР. Данная библиотека подойдет для большинства задач, связанных с генерацией случайных данных, таких как формы, базы и т.д., а функционал библиотеки способен, при создании своих поставщиков ихэлперов. Данная библиотека отлично подходит для тестирования и разработки качественных интерфейсов или просто структур, включающих данные.

ЛИТЕРАТУРА

1. Faker.js [Электронный ресурс] / github.com. – 2019. – Режим доступа: <https://github.com/marak/Faker.js>. – Дата доступа: 11.04.2019.
2. Генерирование фейковых данных для вашего JavaScript-приложения [Электронный ресурс] / habr.com. – 2019. – Режим доступа: <https://habr.com/ru/post/248999/>. – Дата доступа: 11.04.2019.

УДК 004.9

Студ. О. М. Карпач
Науч. рук. ст. преп. Р. В. Азарчик
(кафедра информатики и веб-дизайна, БГТУ)

ОСОБЕННОСТИ ДИЗАЙНА И РАЗРАБОТКИ ИГРОВОГО ПОРТАЛА «ТАЙНЫ ЗАБЫТОГО ОСТРОВА»

Игровой портал «Тайны забытого острова» посвящён игре и предназначен для получения более детальной информации о ней, общения пользователей и разработчиков, а так же прохождения захватывающих миссий (рисунок 1).

Данный продукт разрабатывался для облегчения популяризации приложения посредством быстрого добавления новых статей, реализации пользовательского кабинета, а также организации браузерного веб-приложения с удобным интерфейсом и захватывающей игровой реальностью.

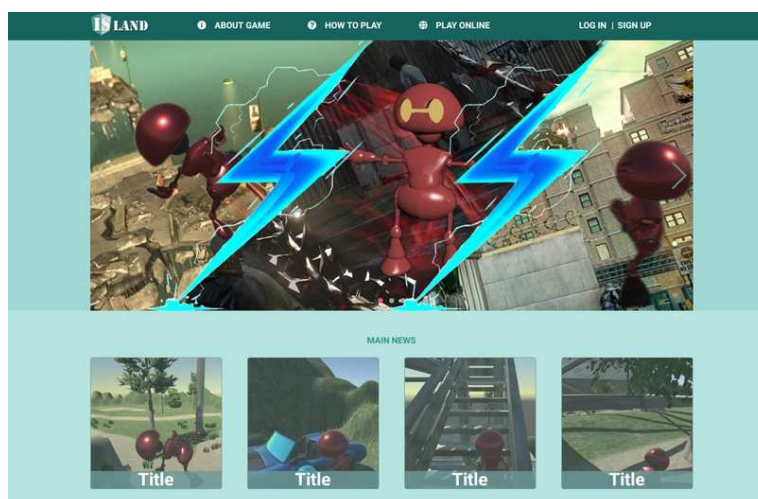


Рисунок 1– Игровой портал «Тайны забытого острова»

Была поставлена задача организации игрового портала, вследствие чего было принято решение составления плана работ:

- организация базы данных;
- написания WEB API сервера;
- разработка swagger-документации;
- создание прототипа;
- разработку клиента с помощью фреймворка Angular7.

Для организации базы данных была использована свободная, объектно-реляционная система управления базами данных – PostgreSQL.

Для реализации WEB API сервера была выбрана программная платформа, основанная на движке V8 – Node.js. Пакетная система, ко-

торая называется – NPM и является самой большой в мире экосистемой библиотек с открытым исходным кодом.

Для интеграции данного ресурса с другими игровыми платформами в будущем, а так же для тестирования имеющихся API маршрутов и серверной валидации, использовалась технология – Swagger (рисунок 2).

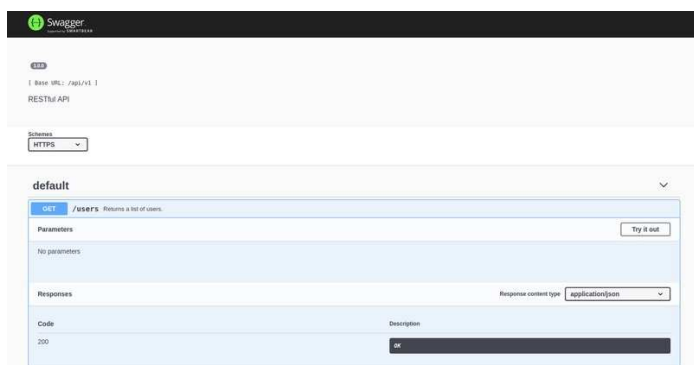


Рисунок 2– Панель управления маршрутами «Swagger»

Добавление новостей реализовано при помощи системы управления контентом.

Для разработки дизайна было проанализировано несколько существующих веб-ресурсов.

Для описания стилей было принято решение использовать технологию SASS, которая значительно упрощает написание CSS-стилей, являющихся основой и главной составляющей практически каждого игрового портала и сайта в целом (рисунок 3).

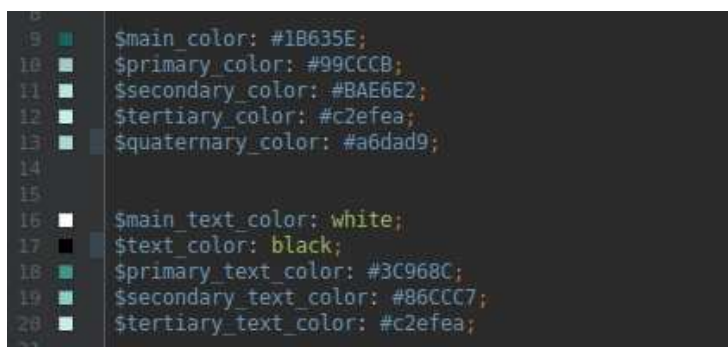


Рисунок 3– Структура кода «SASS»

Для реализации клиента использовался фреймворк – Angular7, который является одним из самых популярных фреймворков при разработке профессиональных веб-приложений (рисунок 4). Именно этот фреймворк позволяет реализовывать “SinglePage” приложения, особенностью которых является однократная загрузка компонентов при перманентной подгрузке необходимого контента [1].

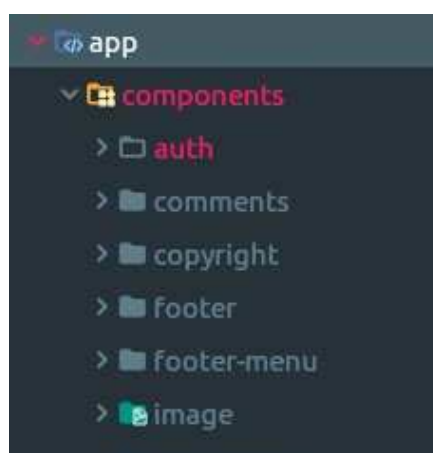


Рисунок 4– Компоненты SinglePageApplications

Для создания цветового оформления игрового портала (рисунок 5). Был использован онлайн генератор цветовых гамм при помощи родительского цвета (#BAE6E1).

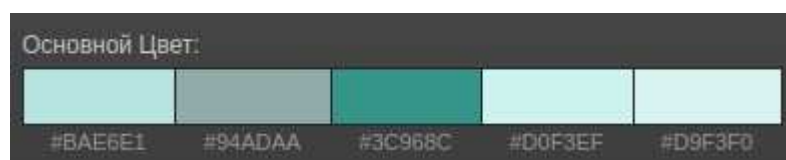


Рисунок 5–Цветовое оформление веб-ресурса

Таким образом, использование данного игрового портала позволит создать хорошую платформу для общения пользователей, разработчиков и администрации сайта. Решит вопрос с доступностью приложения с разных устройств, а так же предоставит хороший функционал как пользователям, так и администрации данного игрового портала.

ЛИТЕРАТУРА

1. Руководство по Angular 7[Электронный ресурс] / Сайт МЕТАНИТ. –2012–2019. – Режим доступа: <http://metanit.com/web/angular2>.– Дата доступа: 11.04.2019.

УДК 004.9

Студ. Е. А. Ахремчик
Науч. рук. ст. преп. Р. В. Азарчик
(кафедра информатики и веб-дизайна, БГТУ)

ОСОБЕННОСТИ ДИЗАЙНА И РАЗРАБОТКИ ВЕБ-САЙТА «YUOLLA»

Студия мебели «YUOLLA» занимается разработкой кухонь, шкафов, ванных комнат, гостиных и т. д. разной сложности, а так же заниматься производством мебели по немецкой технологии «Фолдинг».

Веб-сайт «YUOLLA» разрабатывался для улучшения имиджа компании, создан удобный и легкий в использовании интерфейс для пользователей сайта.

В ходе работы были поставлены цели:

- разработка логотипа веб-сайта;
- разработка айдентики компании;
- выбор технологий для реализации приложения;
- программная реализация продукта;
- подключение дополнительных сервисов к веб-сайту;
- оптимизация веб-сайта под различные виды устройств;
- тестирование веб-сайта в различных условиях;
- проведение рекламной компании веб-сайта.

При разработке логотипа у заказчика было пожелание создать шрифтовой логотип (рисунок 1).



Рисунок 1 –Разработанные форматы логотипов

При разработке дизайна было принято единоголосное решение в стилистике сайта – это классицизм и минимализм. Акценты были расставлены на вытянутые и прямые формы. Нашим предложением при разработке дизайна было широкоформатное разрешение, т.е. главные страницы сайта должны располагаться на всю ширину экрана (рисунок 2). Для программной реализации был использован язык программирования php, после чего сайт был интегрирован с CMS«Wordpress» [1] по пожеланию заказчика. Для реализации адаптивности сайта была выбрана технология Bootstrap 4.

Главным преимуществом сайта является уникальность, были созданы воронки и точки захвата лида и достижение цели в понимании и передачи атмосферы мебельной тематики, а также удобный ин-

терфейс добавления информации на сайт самим заказчиком без написания кода (рисунок 3).

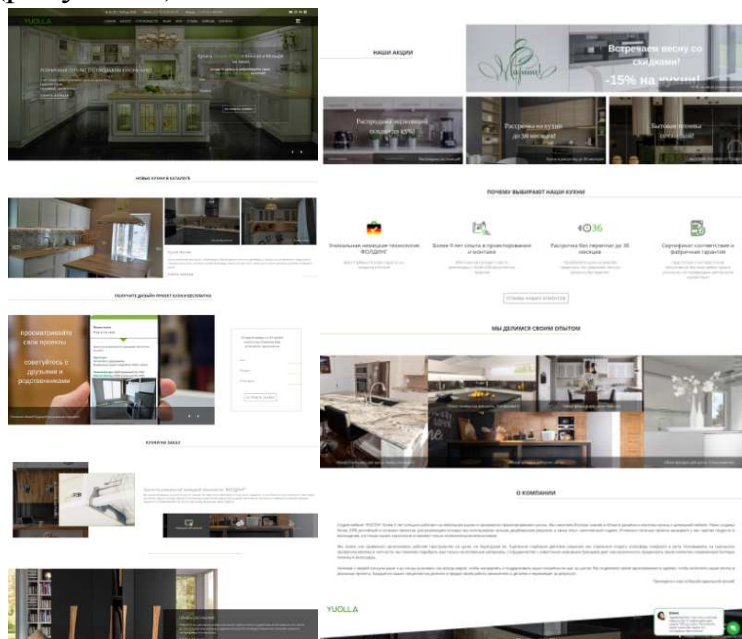


Рисунок 2 – Интерфейс главной страницы

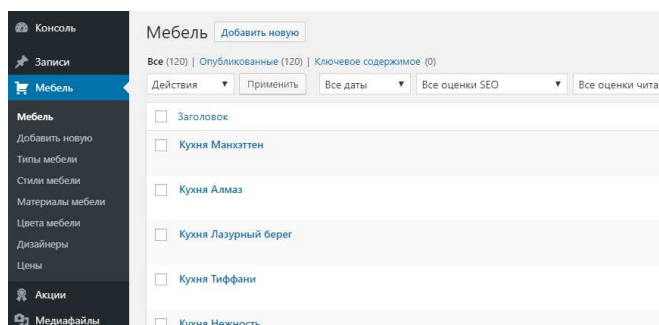


Рисунок 3 – Интерфейс админ панели при добавлении мебели

CMS «WordPress» предлагает большой набор расширений:

- 1) SEO (рисунок 4).
- 2) визуальный редактор для административной части (рисунок 5).

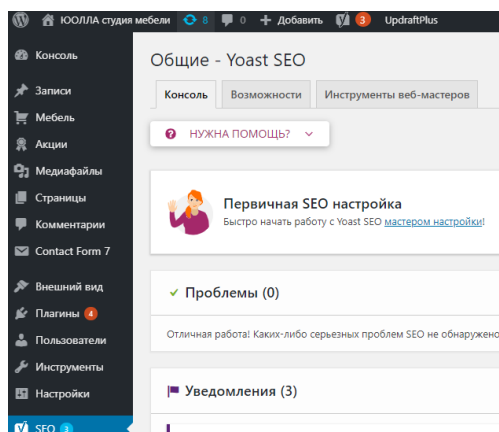


Рисунок 4 – Интерфейс плагина для SEO

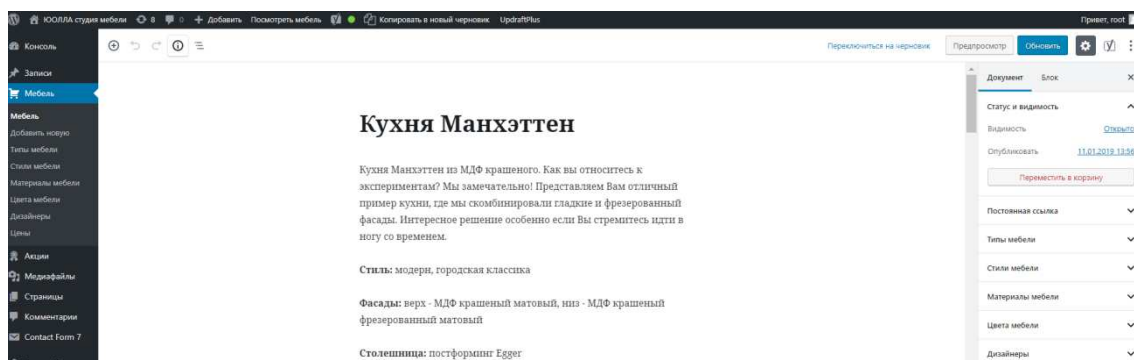


Рисунок 5 – Визуальный редактор для административной части

В качестве цветового оформления веб-ресурса были выбраны следующие цвета: белый, зелёный (#69b221), чёрный (рисунок 6).

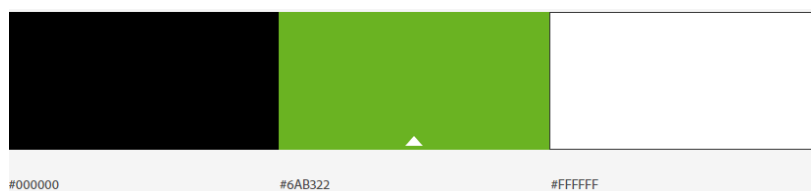


Рисунок 6– Цветовая палитра

ЛИТЕРАТУРА

1. Wordpress документация – [Электронный ресурс]. – 2015–2018. – Режим доступа: <https://codex.wordpress.org/>. – Дата доступа: 10.04.2018.

УДК 004.9

Студ. А. В. Харланович
 Науч. рук. ст. преп. Р. В. Азарчик
 (кафедра информатики и веб-дизайна, БГТУ)

ОСОБЕННОСТИ ДИЗАЙНА И РАЗРАБОТКИ ВЕБ-САЙТА «АМАН ТРЕЙДИНГ»

Каждый сайт в сети интернет выполняет свою функцию. Сайты могут быть новостными, рекламными, личными блогами, закрытыми ресурсами, обучающими, детскими. Также немаловажную роль в разработке сайтов занимает возможность просматривать его на различных устройствах: мобильных телефонах, планшетах, компьютерах с обычным и маленьким монитором.

Перед разработкой веб-сайта проводится анализ различных аналогов. Это позволяет определить все плюсы и минусы других проектов, что необходимо на ранних этапах разработки концепции элек-

тронного продукта и позволяет избежать аналогичных ошибок в дальнейшей работе. Также при разработке визуальной составляющей сайта учитывались принципы дизайна и оформления. Оформление было создано на основе трендов дизайна в 2019 года.

Главной целью проекта является создание веб-сайта компании «Аман Трейдинг» для информирования потенциальных и постоянных клиентов о товарах и предлагаемых услугах, а также, продвижению компании на рынке Беларуси по распространению кондитерского и хлебобулочного сырья.

Задача сайта – оптимально быстро, удобно и грамотно предоставить информацию для пользователя сайта. Для этого необходимо выяснить и учесть требования посетителя и спроектировать удобный интерфейс для пользователей. Также основной и не мало важной ситуацией является, предоставить пользователям качественный, современный и популярный контент.

Одной из перспективных задач развития хлебопекарной отрасли является расширение ассортимента хлеба и хлебобулочных изделий путем освоения новых технологий и рецептур. В целях совершенствования ассортимента хлебобулочной продукции необходимо отслеживать изменение потребностей населения с целью их более полного удовлетворения. Так как работа предусматривает создание сайта-каталога продукции, то перспектива развития проекта является полного и удобного информирования о товаре, смотря как того хочет видеть посетитель.

Для того чтобы проект был успешным, необходимо на начальном этапе правильно определить целевую аудиторию. Сначала нужно ответить на вопрос: «Кому будет интересен мой веб-сайт?». Веб-сайт будет интересен юридическим лицам и предпринимателям, занимающимся производством хлебобулочных и кондитерских изделий, а также различным мелким и крупным компаниям, которым будет требоваться предлагаемая продукция. Так как в компаниях выбором товаров, ее закупкой, использованием в производстве будут заниматься разные сотрудники, то сайт должен быть удобен каждому, кто его будет посещать, и предоставлять требуемую для них информацию.

Любой сайт еще на начальном этапе своей разработки должен быть тщательно продуман с точки зрения будущей функциональности. Функции должны полностью соответствовать целям и задачам ресурса. Сайт должен быть максимально простым (без лишних элементов), но и достаточно функциональным. Для составления функциональной схемы были выделены следующие разделы сайта:

– главная (главная страница, которая загружается при открытии сайта, на ней находится меню и ссылки на другие разделы);

- каталог (каталог продуктов, выполняет роль сортировки);
- новости (блог с новостями компании);
- контакты (контактная информация, возможность обратной связи);
- о компании (раздел, в котором есть краткое описание компании);
- рецептура (страница с рецептурами, поделенными на группы);
- админ-панель (возможность входа для администратора и других лиц компании для управления информацией на сайте).

Диаграмма вариантов использования (сценариев поведения, прецедентов) является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Данная диаграмма (рисунок 1) состоит из актеров, вариантов использования и отношений между ними. При построении диаграммы могут использоваться также общие элементы нотации: примечания и механизмы расширения.

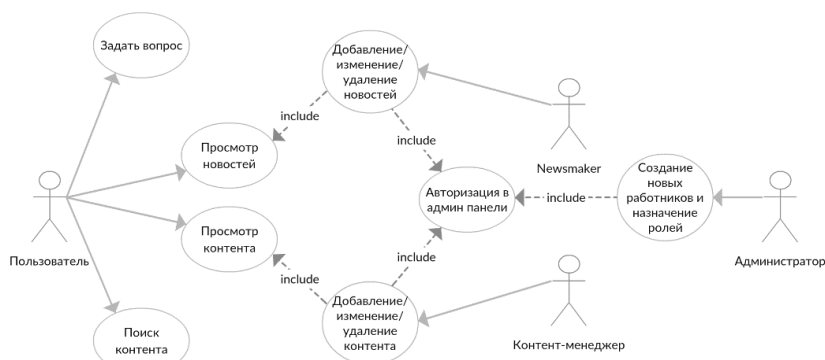


Рисунок 1– Диаграмма использования

На диаграмме (рисунок 1) использования показано, что пользователь может сделать на сайте и какие действия могут быть со стороны администратора сайта.

При выборе дизайна сайта дипломного проекта, необходимо было учесть его функциональность. Если сделать дизайн с большим количеством ярких картинок, кнопок, баннеров, то это не поможет привлечь пользователя. Не нужно добавлять на сайт много анимации. Лучше отдавать предпочтение функциональности и эффективности содержимого, а не эффектности формы.

Для реализации программной составляющей проекта используется язык разметки HTML5 с CSS, язык программирования JavaScript, PHP и базу данных MySQL в phpMyAdmin. Так же для графического контента и макетирования веб-ресурса используются следующие программы: Adobe Photoshop, Adobe Illustrator, Axure.

После проведенного анализа аналогов веб-сайтов и стилей дизайна, было принято решение использовать для проекта минималистичный цветной стандартный стиль с элементами классического и журнального стилей. Сайт в минималистичном стиле сконцентрирован на каком-либо главном элементе и состоит из одной или двух колонок. Структура сайта обязательно должна соответствовать разрешению монитора, на котором его будет просматривать пользователь. Самым популярным разрешением является 1280×1024 пикселей, поэтому необходимо в первую очередь подстраивать дизайн сайта под это разрешение.

Структура интерфейса веб-площадки понятна для обычного пользователя, в тоже время содержит необходимые функциональные средства, обеспечивающие удобство работы с набором объявлений, с поиском нужных объявлений и статей (рисунки 2–3).

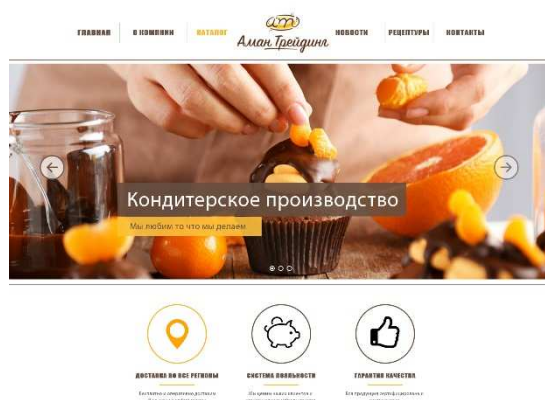


Рисунок 2– Макет главной страницы



Рисунок 3 – Макет страницы описания товара

В дизайне были разработаны несколько вариантов тематического оформления сайта, например, светлая, темная и новогодняя.

Таким образом, использование данного сайта позволит пользователям быстро и удобно находить требуемую им информацию, а также просматривать новостные статьи компании.

ЛИТЕРАТУРА

1. AdobePhotoshop [Электронный ресурс] / Сайт ADOBE. – Режим доступа: https://www.adobe.com/cis_ru/products/photoshop.htm – Дата доступа: 29.03.2019.
2. Кантелон, М. JavaScript в действии / Майк Кантелон, Марк Хартер, Натан Райлих. – СПб: Питер, 2015. – 448 с.
3. Видео уроки PHP [Электронный ресурс] / Канал YOUTUBE. – Электрон. дан. – Режим доступа: <https://www.youtube.com/user/ilia-kan>. – Дата доступа: 27.03.2019.

УДК 004.9

Студ. А. М. Михайлова

Науч. рук. ст. преп. Р. В. Азарчик

(кафедра информатики и веб-дизайна, БГТУ)

РАЗРАБОТКА ВЕБ-САЙТА «УП "БЕЛЛЕСЭКСПОРТ"»

Веб-сайт УП «Беллесэкспорт» разрабатывается на основе уже имеющегося сайта экспортно-производственного республиканского унитарного предприятия. Разработка нового сайта позволит улучшить и структурировать, а также привнести новый стиль в создание сайтов государственного масштаба. Сайт даст возможность пользователям в несколько кликов получить желаемый результат.

Данный продукт разрабатывается с целью лёгкого, удобного и интуитивного понятного дизайна показать пользователям необходимую информацию.

Задачей данной работы является реализация веб-сайта для размещения новостей и каталога экспортных и внутренних лесопроductций, оказание услуг по сервисному обслуживанию лесозаготовительных машин, оптовая и розничная торговля товарами, используя PHP, базу данных MySQL в phpMyAdmin.

Актуальность выбранной темы определяется тем, что имеющийся сайт организации, морально устарел, поскольку был создан в 2015 году. Для того чтобы привлекать новых клиентов необходим редизайн сайта. Так при создании нового сайта не будет использоваться система управления сайтами 1С- Битрикс, что значительно облегчит управление контентом сайта.

Для данного сайта сделан новый способ добавления контента, при условии разделения прав пользователей и их доступа к возможностям разных отделов. Главную роль выполняет администратор и все основные функции доступны только ему, остальные же пользователи

разделены по отделам и могут добавлять новую информацию в специально отведенном для них месте на сайте.

Веб-сайт разрабатывается на основе языка PHP.

Данный язык был выбран, так как он обеспечивает:

- автоматическое извлечение POST и GET-параметров;
- взаимодействие с большим количеством различных систем управления базами;
- работа с cookies и сессиями;
- обработку файлов, загружаемых на сервер.

Структура интерфейса веб-сайта понятна для обычного пользователя, в то же время содержит необходимые функциональные средства, обеспечивающие удобство работы с каталогами, с поиском нужных продуктов. Для этого все лесопродукции разделены в пункте меню на экспортный и внутренний рынок (рисунок 1).

Поскольку у каждого каталога есть определенный тип, нужно позаботиться о том, чтобы пользователь мог просматривать продукцию только нужного ему типа. Это реализовано с помощью простого выбора в панели справа.

Веб-сайт предоставляет пользователям возможность обратиться к администрации сайта посредством заполнения формы обратной связи как для физических лиц, так и для юридических. Все полученные данные сохраняются в базу данных для дальнейшей обработки их администратором или конкретным отделом.

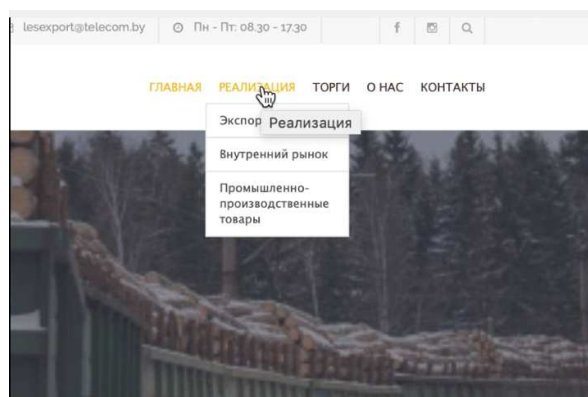


Рисунок 1– Пункт меню для выбора продукции рынка

Для создания административной части веб-сайта были написаны несколько страниц PHP с использованием шаблона Bootstrap [2].

При разработке дизайна сайта были использованы фирменные стили организации, такие как цвет и гарнитура, представленные на изображениях (рисунок 2–3).

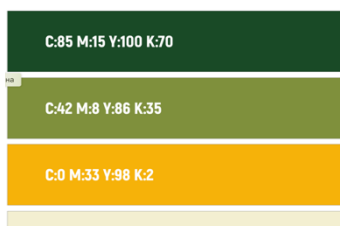


Рисунок 2 – Фирменные цвета организации

Akrobat Black
 абгдєжзиклмнопрстѹфхцщщььэюя
 АБВГДЕЖЗИКЛМНОПРСТУФХЦЩЩЬЬЪЭЮЯ
 1234567890!> %<_{}+
 abcdefghijklmnopqrstuvwxyz
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 1234567890!@#%&'*{}~|_+>

Akrobat
 абгдєжзиклмнопрстѹфхцщщььэюя
 АБВГДЕЖЗИКЛМНОПРСТУФХЦЩЩЬЬЪЭЮЯ
 1234567890!> %<_{}+
 abcdefghijklmnopqrstuvwxyz
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 1234567890!@#%&'*{}~|_+>

Akrobat Light
 абгдєжзиклмнопрстѹфхцщщььэюя
 АБВГДЕЖЗИКЛМНОПРСТУФХЦЩЩЬЬЪЭЮЯ
 1234567890!> %<_{}+
 abcdefghijklmnopqrstuvwxyz
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 1234567890!@#%&'*{}~|_+>

Рисунок 3 –Фирменная гарнитура организации

Так для нового сайта была переделана структура сайта на основе уже имеющейся, поскольку старая версия была интуитивна непонятно новым пользователям сайта (рисунок 4).

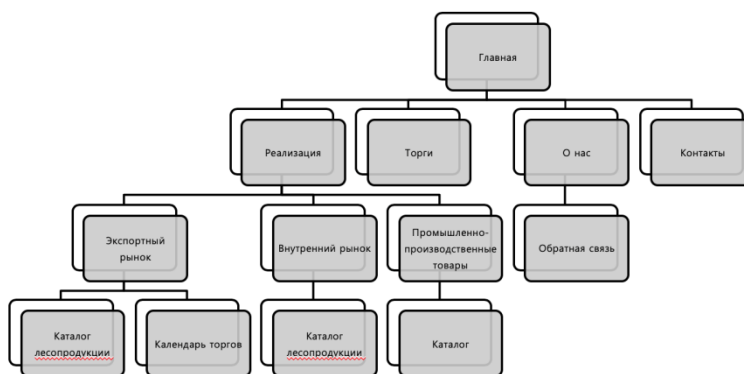


Рисунок 4– Структура сайта

Таким образом, использование данного сайта позволит пользователям быстро и удобно найти необходимую информацию, что повысит уровень продаж товаров и оказание услуг.

ЛИТЕРАТУРА

1. Официальная документация УП «Беллесэкспорт» [Электронный ресурс] / Сайт УП «Беллесэкспорт». – 2015. – Режим доступа: <http://www.bellesexport.by/>. – Дата доступа: 07.04.2019.

2. Официальная документация [Электронный ресурс]/ Уроки.NET.–2005. – Режим доступа: <http://www.uroki.net/>.–Дата доступа: 14.04.2019.

3. Начало работы с php [Электронный ресурс]/ HABR. – 2018. – Режим доступа: <https://habr.com/ru/sandbox/67436/>. – Дата доступа: 14.04.2019.

УДК 004.582

Студ. А. М. Карпач

Науч. рук. ст. преп. Т.П. Брусенцова
(кафедра информатики и веб-дизайна, БГТУ)

РАЗРАБОТКА ФУНКЦИОНАЛЬНОЙ СТРУКТУРЫ СОЦИАЛЬНОЙ СЕТИ «SOCIAL EXCHANGE»

Начальным этапом разработки функциональной структуры социальной сети будет рассмотрение аудитории, для этого нужно ознакомиться с некоторыми человеческими факторами.

При разработке пользовательских интерфейсов для различных сегментов потребительских рынков, учитываются географические, демографические, психографические, поведенческие факторы, влияющие на психосоциальные особенности личности, а также учитываются:

- основные темпераменты: сангвиник, холерик, меланхолик, флегматик;
- основные типы личности: интроверт, экстраверт;
- типы поколений: поколение Y, поколение X, поколение беби-бум.

При разработке интерфейса, например, при темпераменте сангвиник кнопки, иконки, шрифт желательно выбрать крупный, пространство лучше не загромождать, использовать точное разделение тонов букв, фона. В целом, интерфейс должен быть достаточно прост и легок, так как человек подвижный и данные особенности интерфейса помогут ему поскорее найти то, что его интересует, не теряя интерес. При темпераменте флегматик, следует создавать подсказки в интерфейсе, например, иконки, точно дающие понять то, что же будет за ней. В целом, любые особенности должны учитываться, группироваться, отбрасываться, анализироваться и приводиться в действие.

Экстраверсия проявляется в дружелюбном, разговорчивом, энергичном поведении, в то время как *интроверсия* проявляется в более замкнутом и уединённом поведении. Поэтому при рассмотрении типов личности для интерфейса весьма актуально использовать функцию добавления в друзья, а также при нежелании, можно просто не добавляться, что является гениальным решением, помогающим соблюсти потребности, опираясь на данную типизацию.

В процессе проектирования социальной сети были выделены следующие группы пользователей:

- миллениалы или поколение Y (рожденные после 1981 и до 2000-х);
- поколение X (рожденные с 1965–1979 гг);
- поколение бэби-бум (рожденные с 1965–1979 гг).

Также важно и необходимо сформулировать цели создания сети. Любая социальная сеть ставит главной целью создание комфортных условий для построения взаимоотношений пользователей и их коммуникации. Это, а также возможности современных веб-технологий, позволили создавать социальные сети различного типа и направленности. Однако их универсальной общепринятой классификации не существует.

Следующим шагом является проектирование самого интерфейса, для этого нужно четко понимать, что это такое, а также знать порядок и сами шаги проектирования, для достижения наиболее продуктивных результатов. *Интерфейс* – только половина во взаимодействии с системой, другая половина – человек, пользователь.

После выявления нужных параметров, в соответствии с психологическими особенностями проанализированной аудитории разрабатывается прототип проекта, который представляет набор блоков, полей и кнопок. Таким образом, прототипируются все страницы разрабатываемого ресурса. После чего строится ментальная карта проекта (рисунок 1). Затем кодируется интерфейс в соответствии с прототипом, а далее внутренний функционал. При необходимости можно изменять список следования и добавлять незапланированные элементы интерфейса.

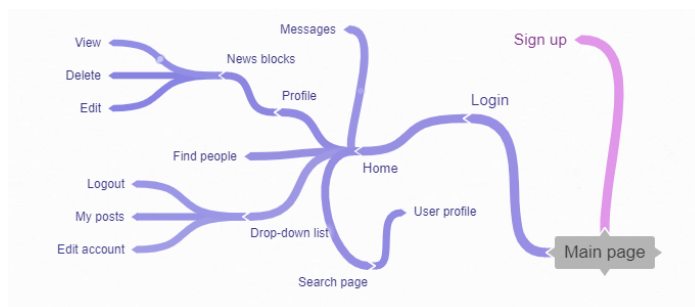


Рисунок 1 – Ментальная карта

Началом древовидной системы является «Main page», от которой отходят две ветви: «Sign up» и «Login». Первая в свою очередь отвечает за регистрацию пользователя, включает в себя колонки для заполнения нужной информацией. Вторая же представляет целую ветку, которая отвечает за все страницы после аутентификации и является в некотором плане стартовой точкой использования веб-сайта. Далее, пользователь переходит на страницу «Home», которая имеет следующий функционал (рисунок 2).

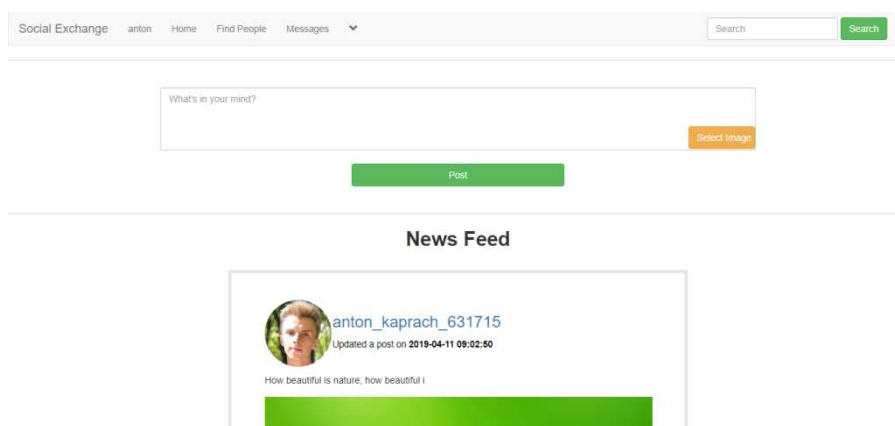


Рисунок 2 – Вид страницы «Home»

Пользователь может видеть меню и некоторое представление ленты новостей. Строка меню в свою очередь имеет заголовки, функционал которых соответствует названию. При переходе на страницу пользователя видим ту же строку меню, она статична на протяжении всей аутентификации пользователя, появляется главный план, изображение пользователя, а также информации, которую он ввел при авторизации, и список новостей. В целом можно сказать, что категории данной социальной сети представлены таким образом, чтобы облегчить функционал для удобства пользователей. Что касается согласованности элементов интерфейса, ее мы добивались:

- использованием пустого пространства, что помогает подчеркнуть элементы интерфейса. В случае проекта, например, это отступы справа и слева.

- применением модульной сетки. В качестве модульной сетки используем фреймворк «Bootstrap», который помогает сэкономить время для поиска нового элемента в макете.

Помимо всего, важно соблюдать согласованность при формировании файлов самого проекта, для этого создаются папки, каждый блок, модуль, документ необходимо хранить в папках с соответствующим названием.

Страница регистрации будет иметь вид (рисунок 3). Что касается информации при вводе, она минимизируется таким образом, чтобы пользователь как можно скорее мог приступить к использованию ресурса. Поля имеют крупный размер, а возле каждого существует пиктограмма, которая в некотором смысле раскрывает значение поля, выпадающие списки представлены таким образом, чтобы было все понятно какой параметр выбрать.

Рисунок 3 – Вид страницы регистрации

Необходимо заметить, что при разработке структуры интерфейса, каждая кнопочка имеет вид, согласованный с предпочтениями пользователей, которые были проанализированы выше. Например, на странице пользователя, кнопки имеют разные цвета, что помогает при дифференциации элементов и говорит, что влияние на интерфейс будет различное (рисунок 4)

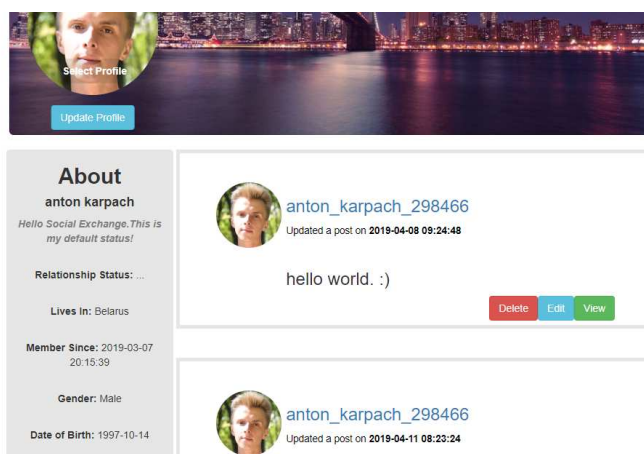


Рисунок 4 – Интерфейс страницы пользователя

В заключении можно сделать вывод, что грамотное проектирование интерфейса, должно способствовать совмещению пользователя и ресурса. Удобное расположение элементов играет важную роль в понимании пользователем большинства процессов, которые будут происходить на данном ресурсе. Важно сопоставлять категории, создавать списки наследования, соблюдать чистоту кода, которая так же оказывает влияние на интерфейс. Необходимо заранее анализировать аудиторию и в последующем программировать интерфейс в соответствии с выявленными задачами. При соблюдении вышеперечисленных факторов и не только, проект способен сохранять аудиторию и доставлять только удовольствие от общения с интернет ресурсом.

УДК 004.5

Студ. Л.С. Юхновец, А.С. Леонова
Научн. рук. ст. преп. Н.И. Потапенко
(кафедра информатики и веб-дизайна, БГТУ)

АНАЛИЗ ЭЛЕКТРОННЫХ ИЗДАНИЙ СМИ НА «ОТЗЫВЧИВОСТЬ»

1. Понятия «Адаптивный» и «Отзывчивый» веб-дизайн. Их сравнительная характеристика.

Наиболее старым и простым типом html-верстки сайта является так называемая фиксированная верстка. Данная технология фиксирует заданную ширину блоков веб-страниц, несмотря на разрешения экрана устройства. К примеру, если шаблон сверстан под разрешение экрана 1024×768 px, то при просмотре на экране 800×600 px и меньше будет появляться горизонтальная прокрутка, а при просмотре в разрешении 1280×1024 px и больше будет виден фон страницы с краев. Развитие мобильных технологий, появление различных планшетов и смартфонов со своими браузерами, увеличение трафика посещаемости Интернет-ресурсов, привело к появлению отзывчивого и адаптивного веб-дизайна.

Основное сходство между этими двумя способами заключается в том, что они позволяют приспособливать сайты к мобильным устройствам с различными расширениями экрана, в конечном итоге обеспечивая удобное отображение сайта, независимо используемого устройства. В тоже время, эти методы отличаются по своей структуре.

При создании Responsive (отзывчивого) веб-сайта используется гибкая структура страницы – «резиновый макет» (fluid grid), пропорции и размеры элементов которой задаются в процентах. Помимо «резиновой» структуры, используются гибкие изображения (fluid images), «резиновые» текстовые блоки. При уменьшении ширины такой страницы весь контент плавно сожмется, структурные элементы уменьшатся относительно друг друга. Так, например, если веб-сайт имел трех колончатую структуру, то на более узком экране он будет иметь две или одну колонку контента. Для создания отзывчивого веб-сайта используются медиа запросы (media queries) – блоки кода с указанием параметров устройств вывода. Медиа запросы позволяют применять определенные правила (стили) для вывода различных контентных блоков в различном порядке и пропорциях в зависимости от ширины экрана и возможностей устройства, на котором происходит просмотр веб-сайта.

Адаптивные веб-сайты (ADV) внешне могут быть похожи на отзывчивые. Ключевым различием между ними является то, что для адаптивного веб-сайта заранее создается набор различных вариантов шаблонов (разметки) для устройств с различными разрешениями экрана,

а также определяется функционал, которым может обладать определенная версия для определённых типов устройств. Например, при создании сайта проектируется дизайн страниц нескольких типовых размеров по ширине: 320 px для смартфонов, 760 px и 1024 px для планшетов, 1680 px для десктоп браузеров. Когда пользователь заходит на адаптивный сайт, сервер определяет, при помощи какого устройства осуществляется просмотр, и показывает пользователю нужный контент и шаблон.

Таким образом, отзывчивый сайт имеет универсальный дизайн, который подстраивается под размеры дисплея, для адаптивного сайта создается несколько шаблонов, оптимизированных и уникальных для каждого класса устройств (рисунок 1).

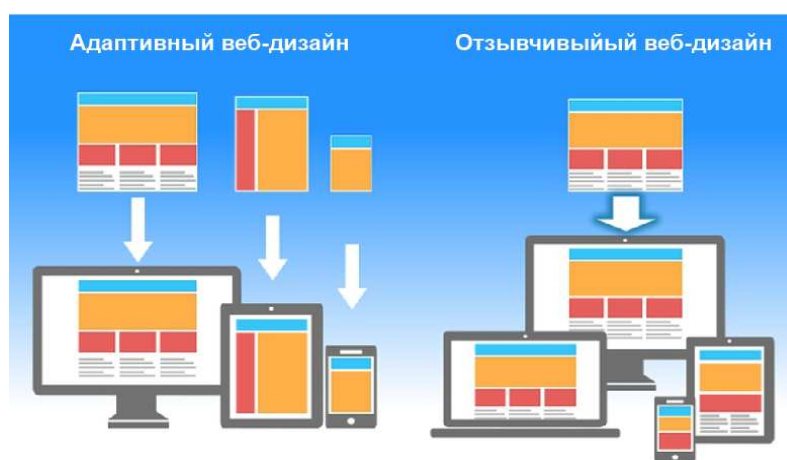


Рисунок 1 – Отображение адаптивного и отзывчивого дизайна

2. Методика проведения исследования электронных изданий СМИ на «отзывчивость».

Для тестирования веб-сайтов на отзывчивость использовались следующие инструменты:

– Am I Responsive (<http://ami.responsivedesign.is>), позволяет посмотреть отображение сайта на четырех устройствах. Доступные размеры: настольный монитор – 1600×992 px; ноутбук – 1280×802 px; планшет – 768×1024 px; мобильный телефон – 320×480 px.

– Screenfly (<http://quirktools.com/screenfly>), позволяет посмотреть отображение сайта на множестве устройств – нетбуках, мониторах, планшетах, телефонах, телевизионных разрешениях, также можно указать произвольное разрешение. Инструмент имитирует строку агента пользователя, но не поведение этих устройств.

В коде страницы просматривается:

1. Наличие адаптации для просмотра страницы на разных устройствах. Для этого страницы должны содержать в контейнере head метатеги viewport. Он сообщает браузеру, каким образом необходимо контролировать размеры и масштаб страницы. Например: `<meta name="viewport"`

content="width=device-width, initial-scale=1">. В данном случае <initial-scale=1> обеспечивает соотношение 1:1 между пикселями CSS и независимыми пикселями устройства. Параметр <width=device-width> позволяет адаптировать ширину окна просмотра к экрану устройства.

2. Использование атрибута media позволяющего выбрать стиль на основе характеристик устройства, а также медиазапросов – это простые фильтры, которые можно применять к стилям CSS. Они позволяют изменять стили на основании характеристик устройства, связанных с отображением контента, включая тип, ширину, высоту, ориентацию, разрешение экрана.

3. Использование относительных единиц. Основной принцип отзывчивого сайта (главное отличие от макетов с фиксированной версткой) – подвижность и пропорциональность. Использование относительных размеров для текстовых областей, изображений помогает упростить макет и предотвратить случайное создание компонентов, не вмещающихся в область просмотра. Ширина колонок задаётся в процентах или сочетанием процентов и пикселей, благодаря чему используется вся эффективная область страницы.

3. Группы электронных изданий для исследования:

1. Информационные порталы

Электронный адрес портала	Верстка
tut.by	фиксированная
Belta.by	фиксированная
360.by	фиксированная
Relax.by	отзывчивая
21.by	фиксированная
B2b.by	фиксированная
Av.by	отзывчивая
onliner.by	фиксированная

2. Информационно-поисковые порталы для специалистов данной области:

Электронный адрес портала	Верстка
Kadrovik.by	Фиксированная
Normativka.by	фиксированная
Jurisconsult.by	фиксированная

3. Популярные газеты:

Название газеты	Электронный адрес	Верстка
7 дней	http://7dney.by/ru	фиксированная
Belarus Today	www.belarustoday.info	фиксированная
Автогазета	http://www.autogazeta.by	фиксированная
АиФ в Беларуси	http://www.aif.by	отзывчивая
Беларускі час	www.belchas.by	фиксированная

БелГазета	http://www.belgazeta.by	фиксированная
Белорусы и рынок	http://www.belmarket.by	отзывчивая
Ва-банкъ	www.vabank.by	фиксированная
Вечерний Минск	www.vminsk.by	фиксированная
Гастроном	http://gastronom.by/	отзывчивая
Звезда	http://www.zviazda.by/be	отзывчивая
Компьютерная газета	www.nestor.minsk.by/kg	фиксированная
Компьютерные вести	https://www.kv.by	отзывчивая
КП в Белоруссии	www.kp.by	фиксированная
Культура	http://www.kimpress.by	фиксированная
Наша Ніва	www.nn.by	фиксированная
Народная Воля	http://www.nv-online.info	фиксированная
Советская Белоруссия	www.sb.by	фиксированная
Экономическая газета	https://neg.by	фиксированная
Экспресс новости	http://www.expressnews.by	отзывчивая

4. Журналы:

Название журнала	Электронный адрес	Верстка
Автожурнал	https://adt.by	отзывчивая
Архитектура и строительство	http://www.ais.by	отзывчивая
Вестник ценообразования	http://www.profmedia.by	фиксированная
Вяселка	http://www.veselka.by	фиксированная
Главный бухгалтер	https://www.gb.by	фиксированная
Готовый бизнес Беларуси	http://www.bagb.by	фиксированная
Дело	http://delo.by	фиксированная
Женский журнал	http://ladys.by	отзывчивая
Индивидуальный предприниматель	http://predprinimatel.by	фиксированная
Изобретатель	http://izobretatel.by	отзывчивая
Класс	http://classmedia.by	отзывчивая
Медицинские новости	http://www.mednovosti.by	фиксированная
Мой Интернет	http://www.myinternet.by	фиксированная
Отдел кадров	http://otdelkadrov.by	фиксированная
Юрист	http://jurist.by	фиксированная

4. Отзывчивый дизайн на примере газеты «Звезда».

Газета «Звезда» – ежедневная общественно-политическая газета на белорусском языке (рисунок 2–5).

```
<script src="/sites/all/themes/zviazda_theme/js/modernizr.js"></script>
<link type="text/css" rel="stylesheet" href="http://www.zviazda.by/sites/default/files/css/css_ZUkqzCG7feKKen844492TQWkCk6-tG0H3Fc6C50Y2K4.css" media="all" />
<link type="text/css" rel="stylesheet" href="http://www.zviazda.by/sites/default/files/css/css_Wu8npAzy16WmnnhKxpexfgsArvo1GGaXGy03GMA5bU.css" media="all" />
<link type="text/css" rel="stylesheet" href="//cdn.jsdelivr.net/bootstrap/3.3.5/css/bootstrap.min.css" media="all" />
<link type="text/css" rel="stylesheet" href="http://www.zviazda.by/sites/default/files/css/css_PMzHkckBrae2crS41vROR0dXXdQH2-j8631TaI1jAKwk.css" media="all" />
<script src="http://www.zviazda.by/sites/default/files/js/js_MpKfeLsTh5JIVGZL170sAuT1rqAC38MLLlkjqj0IX_k.js"></script>
```

Рисунок 2 – Использование атрибута media в коде страницы

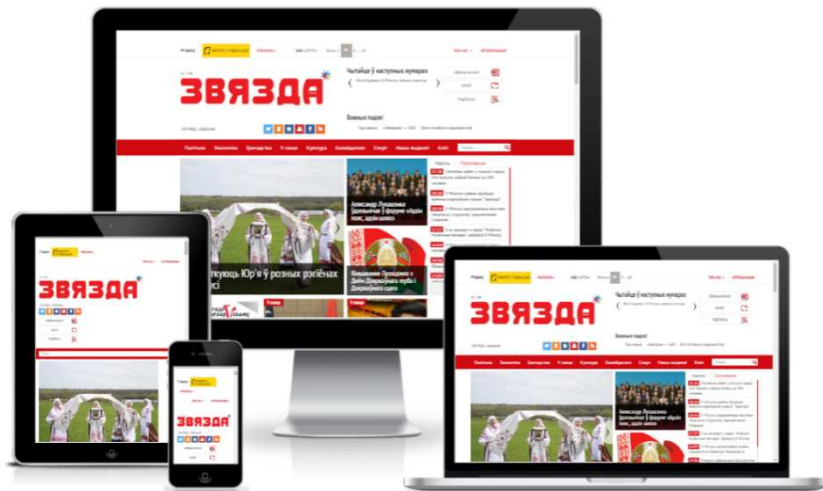


Рисунок 3 – Отображение отзывчивого дизайна с помощью инструмента Am I Responsive

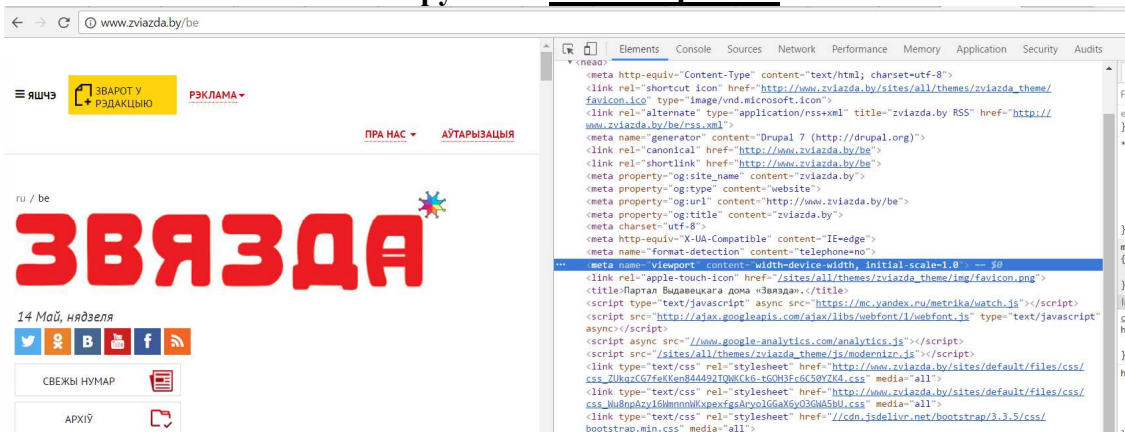


Рисунок 4 – Использование метатега viewport в коде страницы

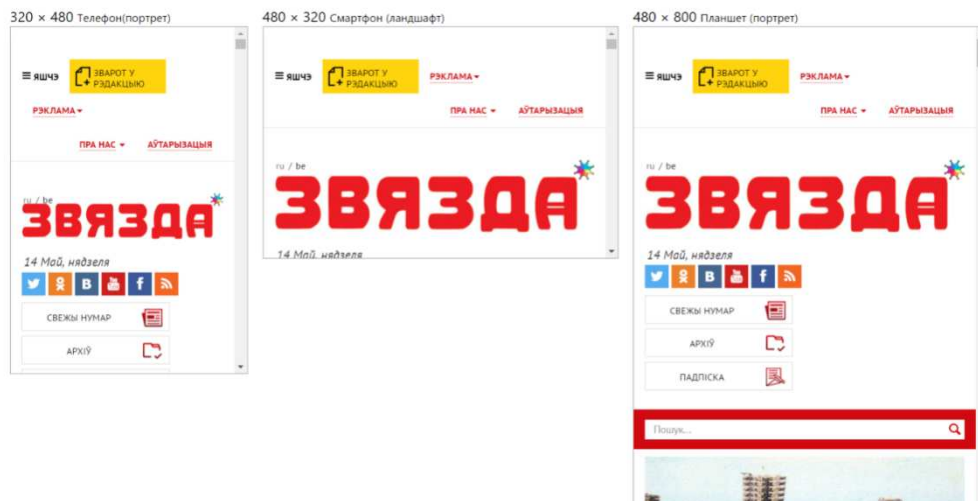


Рисунок 5 – Отображение отзывчивого дизайна с помощью инструмента Screenfly

5. Отображение сайтов с фиксированной версткой (рисунок 6–7)

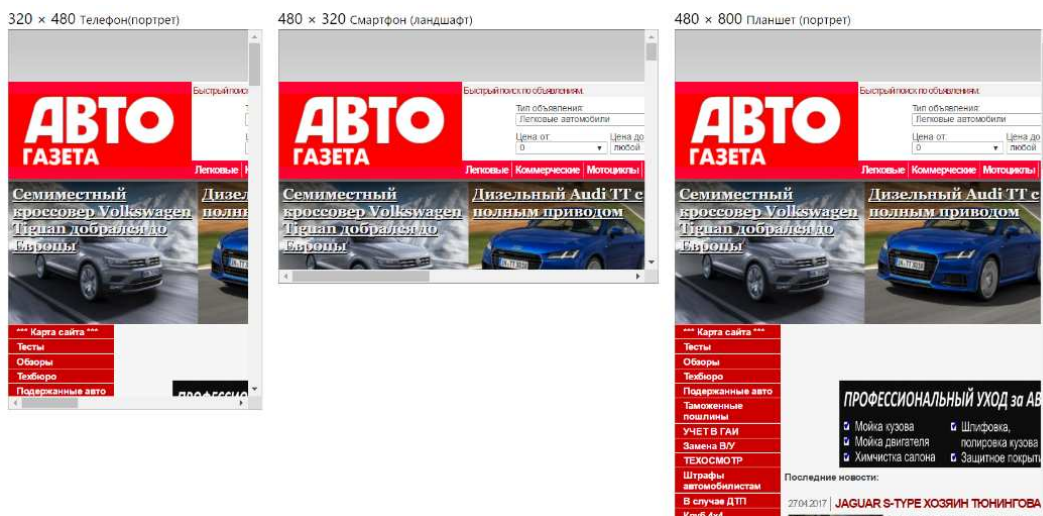


Рисунок 6 – Газета «Автогазета»

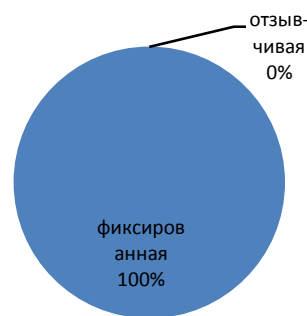


Рисунок 7 – Журнал «Главный бухгалтер»

6. Результаты по группам электронных изданий:



Информационные порталы



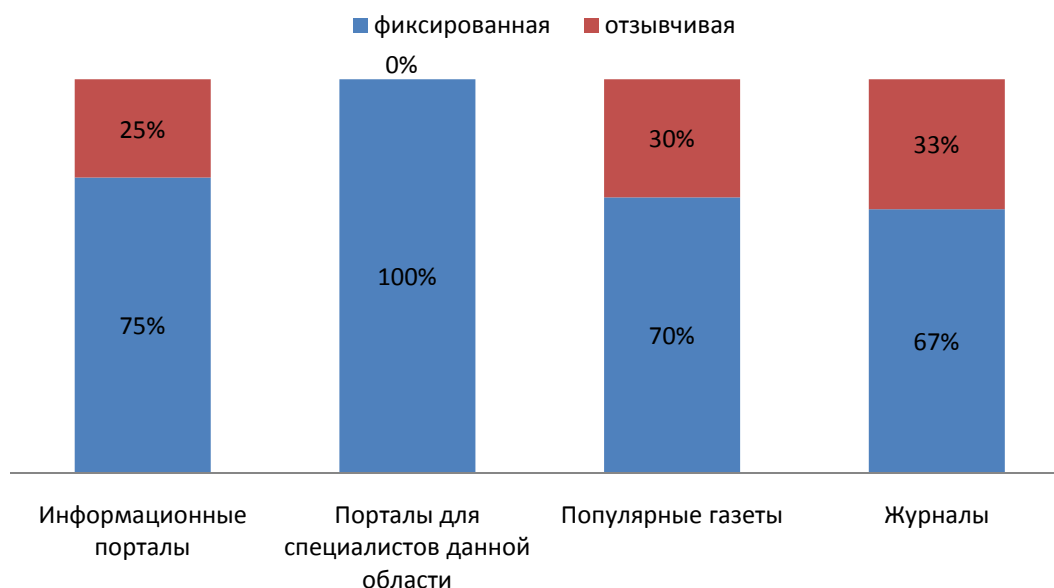
Порталы для специалистов данной области



Популярные газеты



Журналы



Итоговая гистограмма

7. Выводы

Таким образом можно сделать вывод, что «отзывчивость» на белорусских веб-страницах встречается значительно реже, чем привычная фиксированная верстка. Особенно не востребована отзывчивая верстка для специализированных информационно-поисковых порталов. Возможно, это вызвано тем, что большинство из исследуемых веб-изданий появилось в период всеобщего использования фиксированной верстки, переход сайта к отзывчивому дизайну требует переработки существующего дизайна, что требует временных и денежных затрат.

Также разработка отзывчивого дизайна может быть нецелесообразна, когда информационный ресурс предполагает работу на большом экране (например, журнал «Вестник ценообразования», портал для специалистов Normativka.by). Стоит отметить, что некоторые «фиксированные» страницы имеют мобильную версию (интернет-порталы Tut.by, Belta.by, B2b.by, Normativka.by; газеты «Беларускі час», «КП в Белоруссии», «Наша Ніва», «Советская Белоруссия»).

УДК 004.9

Студ. А. С. Леонова, Л. С. Юхновец
Науч. рук. ассист. С.А. Осоко
(кафедра информатики и веб-дизайна, БГТУ)

ИСПОЛЬЗОВАНИЕ СТЕРЕО (АНАГЛИФНЫХ) ИЗОБРАЖЕНИЙ ПРИ ИЗУЧЕНИИ ТЕХНИЧЕСКИХ ДИСЦИПЛИН

В современном мире все большее распространение получает применение стереоэффектов в различных сферах жизни, таких как кино, фотография, образование и т.д. Цель работы – проанализировать возможности использования стерео изображений при подготовке учебной литературы на примере анаглифных изображений.

Актуальность темы заключается в том, что постепенное внедрение 3D-технологий в повседневную реальность заставляет уделять данному аспекту все больше внимания.

Одним из часто встречающихся стереоэффектов являются анаглифные изображения. Это метод получения стереоэффекта для стереопары обычных изображений при помощи цветового кодирования изображений, предназначенных для левого и правого глаза. Сущность анаглифной технологии заключается в окрашивании изображения стереопары в дополнительные цвета, при этом оба кадра стереопары формируют одно изображение. Разделение левого и правого кадра происходит с помощью 3D-очков, состоящих из двух фильтров: красного и синего (бирюзового). В некоторых случаях фильтры могут быть красного и зеленого цветов. Данного рода изображения могут отображаться через мультимедиа-проекторы, в печатных изданиях. Этот метод работает практически на любых цветных телевизорах и мониторах, так как в этих устройствах используется аддитивный способ формирования цвета. Однако полиграфическая печать анаглифных изображений является более сложным процессом. Так как принцип формирования цвета меняется на субтрактивный, а однозначного соответствия между аддитивной и субтрактивной цветовыми моделями не существует.

Суть анаглифного метода заключается в том, что изображения воспринимаются объемными, подлежат детальному рассмотрению, то есть привлекают внимание. Благодаря этому их можно благополучно внедрять при построении моделей различных элементов, приборов, компонентов и т.д., в различных технических дисциплинах.

Основным недостатком метода анаглифов+ является их неполная цветопередача. То есть, формируемое изображение благодаря эффекту бинокулярного смешения цветов воспринимается однотонным или, при определенном соотношении яркостей, ахроматическим.

Адаптация наблюдателя к специфическим условиям происходит достаточно быстро. Однако при пребывании в 3D-очках более 15 минут у наблюдателя на продолжительное время (порядка получаса) снижается цветовая чувствительность и возникает ощущение дискомфорта от восприятия окружающего мира.

Рассмотрим подробный процесс создания подобного рода изображения в программе Adobe Photoshop. Подготовленный снимок (рисунок 1) открывается с помощью выбранной программы, с последующей разблокировкой слоя (двойной щелчок по слою, нажатие на кнопку «ОК»).



Рисунок 1 – Исходное изображение

Исходный слой дублируется, для этого в командной строке выполняется цепочка команд «Слой – Создать дубликат слоя», или горячими клавишами Ctrl + J.

При двойном щелчке на выбранном слое, появляется окно Стиля слоя. В открытом окне находятся дополнительные параметры, в которых необходимо выключить некоторые каналы. Для одного из слоев, необходимо выключить канал R (red). Получим результат, представленный на рисунке 2.



Рисунок 2 – Отключение канала R

Для оставшегося слоя необходимо отключить каналы B (blue), G (green), в ходе чего изображение будет окрашено в красный цвет, как на рисунке 3.



Рисунок 3 – Отключение каналов В и G

С помощью инструмента «Перемещение» сдвигаем слои и получаем готовый вариант анаглифного изображения (рисунок 4).



Рисунок 4 – Анаглифное изображение

В результате мы можем от простой картинке прийти к объемной. Это позволяет нам развить пространственное мышление, без которого невозможно подготовить качественного специалиста по техническим дисциплинам.

ЛИТЕРАТУРА

3. Что такое анаглиф?[Электронный ресурс] / Сайт ISPRING. – Режим доступа: <https://www.ispring.ru/elearning-insights/anaglif>. – Дата доступа: 02.04.2019 г.

4. Виртуальная реальность: стереоскопическое изображение, дисплейные системы. [Электронный ресурс] / Сайт OSP. – Режим доступа: <https://www.osp.ru/pcworld/2008/06/5371423/>. – Дата доступа: 04.04.2019 г.

5. Что такое анаглифные 3D-очки. [Электронный ресурс] / Сайт TECHNO-GUIDE. – Режим доступа: <http://techno-guide.ru/informatsionnye-tekhnologii/3d-tekhnologii/что-такое-anaglifnye-3d-ochki.html>. – Дата доступа: 07.04.2019 г.

6. Анаглиф эффект. Photoshop. [Электронный ресурс] / Сайт TAKE-FOTO. – Режим доступа: https://www.takefoto.ru/articles/obrabotka_fotografii/1730_anaglif_effekt_Photoshop. – Дата доступа: 09.04.2019 г.

Студ. Е. Р. Смирнов
 Науч. рук. А. Н. Щербакова
 (кафедра информатики и веб-дизайна, БГТУ)

РАЗРАБОТКА НА REACT/REDUX С ИСПОЛЬЗОВАНИЕМ RAMDA

React – это инструмент для создания пользовательских интерфейсов. Его главная задача – обеспечение вывода на экран того, что можно видеть на веб-страницах. React значительно облегчает создание интерфейсов благодаря разбиению каждой страницы на небольшие фрагменты, называемые компонентами.

Компонент React – это участок кода, который представляет часть веб-страницы. Каждый компонент – это JavaScript-функция, которая возвращает кусок кода, представляющего фрагмент страницы [1].

Компоненты делятся на:

- умные;
- глупые.

Глупые служат для отображения статического контента страницы (рисунок 1). Умные служат для хранения и работы с состояниями (рисунок 2). Redux – это инструмент управления как состоянием данных, так и состоянием интерфейса в JavaScript-приложениях. Он подходит для одностраничных приложений, в которых управление состоянием может со временем становиться сложным. Redux не связан с каким-то определенным фреймворком, и, хотя разрабатывался для React, также может использоваться с Angular.

```
const Layout = ({ children }) => {
  return (
    <div className='view-container'>
      <Header />
      <div className='container'>
        <div className='row'>
          <div className='col-lg-3 col-xl-3 col-sm-12 col-md-3'>
            <Sidebar />
          </div>
          <div className='col-lg-9 col-xl-9 col-md-9 col-sm-12'>
            {children}
          </div>
        </div>
      </div>
      <Footer />
    </div>
  )
}
```

Рисунок 1 – Глупый компонент

```

class Search extends Component {
  constructor(props) {
    super(props)
    this.state = {
      value: ''
    }

    this.handleChange = this.handleChange.bind(this)
    this.handleSubmit = this.handleSubmit.bind(this)
  }

  handleChange(event) {
    this.setState({
      value: event.target.value
    })
  }

  handleSubmit(event) {
    event.preventDefault();
    this.props.searchWatch(this.state.value)
  }

  render() {
    return (
      <div className='search'>
        <form onSubmit={this.handleSubmit}>
          <input
            type="text"
            onChange={this.handleChange}
            className='searchInput'
            placeholder='Поиск товара'
          />
          <button className='searchButton'></button>
        </form>
      </div>
    )
  }
}

```

Рисунок 2 – Умный компонент

Redux предлагает хранить все состояние приложения в одном месте, называемом «store» («хранилище»). Компоненты «отправляют» изменение состояния в хранилище, а не напрямую другим компонентам. Компоненты, которые должны быть в курсе этих изменений, «подписываются» на хранилище.

С Redux все компоненты получают свое состояние из хранилища. Также ясно, куда компонент должен отправить информацию об изменении состояния – опять же в хранилище. Компонент только инициирует изменение и не заботится об остальных компонентах, которые должны получить это изменение. Таким образом, Redux делает поток данных более понятным.

Ключевые моменты Redux:

- хранилище (store): хранит состояние приложения.
- действия (actions): некоторый набор информации, который исходит от приложения к хранилищу и который указывает, что именно нужно сделать. Для передачи этой информации у хранилища вызывается метод `dispatch()`.

Reducer: функция (или несколько функций), которая получает действие и в соответствии с этим действием изменяет состояние хранилища [2].

Общая схема взаимодействия элементов архитектуры Redux представлена на рисунке 3.

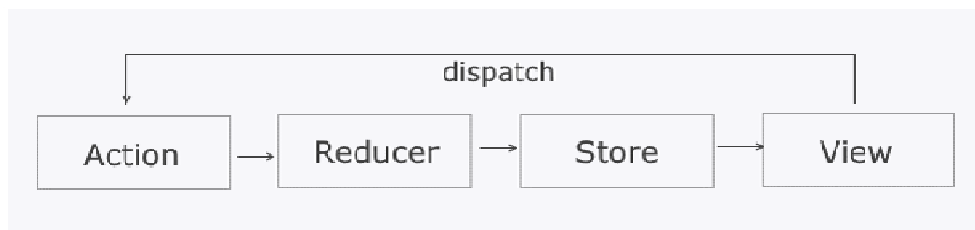


Рисунок 3–Схема взаимодействия элементов архитектуры Redux

Из View (то есть из компонентов React) мы посылаем действие, это действие получает функция reducer, которая в соответствии с действием обновляет состояние хранилища. Затем компоненты React применяют обновленное состояние из хранилища.

Ramda предлагает другой стиль написания кода, стиль, который был позаимствован из чисто функциональных языков программирования. Здесь механизм создание сложной функциональной логики реализуется с помощью композиции. Однако, любая библиотека предоставляет возможность реализовать функциональную композицию. Но в отличие от других, тут это «делается с легкостью» [3].

Пример использования данных технологий на разработанном проекте интернет-магазина часов.

Структура подключаемых пакетов представлена на рисунке 4.

```
import React from 'react'
import ReactDOM from 'react-dom'
import { createStore, applyMiddleware } from 'redux'
import { composeWithDevTools } from 'redux-devtools-extension'
import thunk from 'redux-thunk'
import { syncHistoryWithStore } from 'react-router-redux'
import { Router, Route, browserHistory } from 'react-router'
import { Provider } from 'react-redux'
```

Рисунок 4–Структура подключаемых пакетов

Главная страница сайта представлена на рисунке 5.

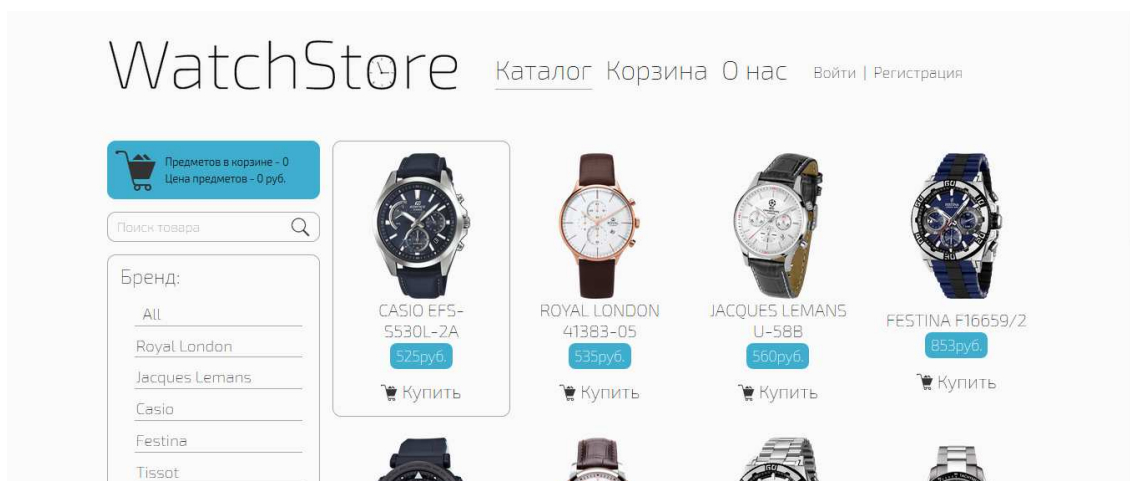


Рисунок 5–Главная страница веб-сайта

Фрагмент реализации поиска на веб-сайте представлен на рисунке 6.

```
const applySearch = item => R.contains(
  state.watchesPage.search.toUpperCase(),
  R.prop('name', item)
)
```

Рисунок 6–Фрагмент реализации поиска на веб-сайте

`R.contains`получает значение `bool`, если есть вхождение подстроки в строку, `R.prop`получает значение по ключу.

Таким образом, можно сделать вывод, что, используя `react/redux`, можно очень просто обрабатывать данные внутри компонента независимо от других компонентов и передавать данные между ними. Также использование `ramda` за счет набора своих функций весьма упрощает работу с обработкой данных.

ЛИТЕРАТУРА

1. Основы React– [Электронный ресурс]. – 2006-2019. – Режим доступа: <https://habr.com/ru/company/ruvds/blog/343022/>. – Дата доступа: 20.04.2018.

2. Redux– [Электронный ресурс]. – 2014-2018. – Режим доступа: <https://getinstance.info/articles/react/learning-react-redux/>. – Дата доступа: 20.04.2018.

3. Почему Ramda? – [Электронный ресурс]. – 2014-2018. – Режим доступа: <https://getinstance.info/articles/react/learning-react-redux/>. – Дата доступа: 20.04.2018.

УДК 004.42

А. К. Пустовалов

Науч. рук. доц., к.т.н. А. П. Лашенко
(кафедра информатики и веб-дизайна, БГТУ)

РАЗРАБОТКА САЙТА ДЛЯ АНАЛИЗА ХОЗЯЙСТВЕННОЙ ДЕЯТЕЛЬНОСТИ ПРЕДПРИЯТИЯ

В последние десятилетия экономика Беларуси переживает период глубоких преобразований, которые осуществляются на всех уровнях производства. Основой процесса преобразования является адаптация всех экономических субъектов к рыночным условиям хозяйствования. В этой ситуации предприятия и организации должны прорабатывать и внедрять новые принципы организации производства продукции, повышать эффективность производства за счет производимой продукции.

Для управления производством нужно иметь полную и правдивую информацию о ходе производственного процесса, о результатах хозяйственной деятельности, о тенденциях и характере происходящих изменений в экономике предприятия. Такую информацию можно получить с помощью анализа хозяйственной деятельности предприятия.

На основе результатов анализа разрабатываются и обосновываются управленческие решения.

Основные расчеты в настоящее время производятся на компьютере с использованием приложения Excel. Работа довольно трудоемкая, требующая внимательности. При этом легко допустить ошибки, которые удалось бы избежать, если этот процесс автоматизировать.

В данной работе представлен сайт, с помощью которого можно не только рассчитать основные показатели, но и произвести анализ хозяйственной деятельности предприятия. На первой странице сайта (рисунок 1) расположены гиперссылки.

При выборе соответствующих гиперссылок открываются страницы (одна из которых приведена на рисунке 2), содержащие таблицы для ввода исходных данных, кнопки, нажимая на которые можно произвести расчеты, а также на странице приводится текст анализа экономических показателей.



Переход к рыночной экономике требует от предприятий повышения эффективности производства, конкурентоспособности продукции и услуг на основе внедрения достижений научно-технического прогресса. Важная роль в реализации этой задачи отводится экономическому анализу производственной деятельности субъектов хозяйствования. С его помощью вырабатывается стратегия и тактика развития предприятия, обосновываются планы и управленческие решения.



- [1. Анализ показателей производства и реализации продукции.](#)
- [2. Анализ трудовых показателей.](#)
- [3. Анализ затрат и себестоимости продукции.](#)
- [4. Анализ показателей эффективности функционирования организации.](#)
- [5. Анализ показателей финансового состояния.](#)

Рисунок 1– Первая страница разработанного сайта



Введите исходные данные

Объем производства продукции, тыс. руб.		Выручка от реализации продукции, тыс. руб., тыс. руб.	
<i>за предыдущий год, тыс. руб.</i>	175690	<i>за предыдущий год, тыс. руб.</i>	205814
<i>по плану на отчетный год, тыс. руб.</i>	192239	<i>по плану на отчетный год, тыс. руб.</i>	215834
<i>фактически за отчетный год, тыс. руб.</i>	193627	<i>фактически за отчетный год, тыс. руб.</i>	211988
Начисленный налог на добавленную стоимость, тыс. руб.		Прочие налоги и сборы, тыс. руб., тыс. руб.	
<i>за предыдущий год, тыс. руб.</i>	35297	<i>за предыдущий год, тыс. руб.</i>	
<i>по плану на отчетный год, тыс. руб.</i>	37015	<i>по плану на отчетный год, тыс. руб.</i>	
<i>фактически за отчетный год, тыс. руб.</i>	36356	<i>фактически за отчетный год, тыс. руб.</i>	

Рисунок 2– Страница сайта для анализа показателей производства и реализации продукции

В процессе выполнения анализа хозяйственной деятельности предприятия осуществляется формирование информационного массива и расчет основных экономических показателей организации, к которым относятся:

- показатели производства и реализации продукции;
- трудовые показатели;
- показатели затрат и себестоимости продукции;
- показатели эффективности функционирования организации;
- показатели финансового состояния организации.

На странице сайта после ввода исходных данных надо нажать на кнопку «Выполнить расчеты» и в тексте анализа в нужных полях появятся вычисленные значения (рисунок 3). Исходный код программы (рисунок 4).

фактически за отчетный год, тыс. руб.	27514	фактически за отчетный год, тыс. руб.	45499
---------------------------------------	-------	---------------------------------------	-------

Выполнить расчеты

Анализ

В отчетном году произведено продукции на сумму 193627 тыс. руб., что на 1388 тыс. руб. больше запланированного объема производства, т. е. на 0.71684%.

По сравнению с предыдущим годом произведено продукции на 17937 тыс. руб. больше, т. е. на 10.2094%.

Выручка от реализации за вычетом налогов при этом составила 175632 тыс. руб., что больше на 2.91233% выручки предыдущего периода. План по выручке недовыполнен на -1.814%.

Коэффициент реализации в предыдущем периоде был равен 0.970556, а в отчетном 0.907063.

Рисунок 3– Страница сайта с текстом анализа показателей

```
<html>
<head>
<title>Анализ хозяйственной деятельности</title>
<script language="JavaScript">
function analiz() {
fl.an1.value=fl.oop3.value;
var tmp=fl.oop3.value-fl.oop2.value;
if(tmp>0)
{ fl.an2.value =tmp+" тыс. руб. больше" }
else
{ fl.an2.value =tmp+"тыс. руб. меньше" };
fl.an3.value=parseFloat(fl.an2.value)*100/parseFloat(fl.an1.value);
var tmp=fl.oop3.value-fl.oop1.value;
if(tmp>0)
{ fl.an4.value =tmp+" тыс. руб. больше" }
else
{ fl.an4.value =tmp+"тыс. руб. меньше" };
fl.an5.value=parseFloat(fl.an4.value)*100/parseFloat(fl.oop1.value);
fl.an6.value=fl.vip3.value-fl.nds3.value;
tmp2=fl.vip1.value-fl.nds1.value;
tmp1=parseFloat(fl.an6.value-tmp2)*100/parseFloat(fl.an6.value);
```

Рисунок4– Исходный код

Сайт разработан на языке разметкиHTML-5 [1, 2]. В состав HTML-документов включены программы, написанные на языке JavaScript, с помощью которых осуществляются расчеты экономических показателей и результаты помещаются в текстовые поля, включенные в текст анализа.

ЛИТЕРАТУРА

1. Фрейн Бен. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. – СПб: Питер, 2014. – 298 с.
2. Евсеев Д.А. Web-дизайн в примерах и задачах, уч. пособие для студентов высших учебных заведений – М.:Кнорус, 2016 – 263 с.

УДК 004.42

Студ. А. В. Буйко

Науч.рук. доц., к.т.н. А. П. Лащенко
(кафедра информатики и веб-дизайна, БГТУ)

ИСПОЛЬЗОВАНИЕ СЦЕНАРИЕВ JAVASCRIPT В HTML-ДОКУМЕНТАХ

Одним из наиболее популярных видов деятельности в IT-индустрии является создание и разработка web-страниц. В наши дни, существует множество возможностей для создания сайтов пользователям, с абсолютно разным уровнем знаний в программировании.

Как правило, составляющим любого HTML-документа является JavaScript, – язык сценариев, или скриптов. Скрипт представляет собой программный код – набор инструкций, который не требует предварительной обработки (например, компиляции) перед запуском. Код JavaScript интерпретируется движком браузера во время загрузки веб-страницы. Интерпретатор браузера выполняет построчный анализ, обработку и выполнение исходной программы или запроса.

С помощью этого языка можно:

- изменять содержимое страницы, т. е. дописывать или убирать текст, менять стили и теги;
- реагировать на какие-то события (например, на щелчок мыши) и выполнять заданную функцию, и пр.

Перед тем как использовать JavaScript, его необходимо добавить в HTML документ. Сделать это можно с помощью элемента `<script>` двумя способами:

- 1) определить встроенный сценарий, который располагается непосредственно между парой тегов `<script>` и `</script>`;
- 2) подключить внешний файл с JavaScript-кодом.

Для иллюстрации вышесказанного был разработан сайт, на веб-странице которого размещен фотоальбом с использованием JavaScript. Это реализовано в три шага.

Первый шаг – размещение в заголовке HTML-документа сценариев:

```
<script type="text/javascript"
src="js/prototype.js"></script>
<script type="text/javascript"
src="js/scriptaculous.js?load=effects"></script>
<script type="text/javascript"
src="js/lightbox.js"></script>
```

Второй шаг:

- добавление `rel="lightbox"` ко всем ссылкам активации изображений, например:
``

HTMLКодирование

;

– добавлении rel="lightbox" ко всем изображениям, например:

```
<a rel="lightbox[roadtrip]" href="image.jpg">  
  
</a>;
```

Для набора связанных изображений:

```
<a href="images/all1.jpg" rel="lightbox[roadtrip]">Фото1</a>  
<a href="images/all2.jpg"  
rel="lightbox[roadtrip]">Фото2</a>ит.д.
```

Количество загружаемых изображений не имеет значения.

Третий и заключительный шаг создания фотоальбома представляет собой загрузку рисунков в нужную папку. Результаты проведённой работы можно видеть на рисунке 1.

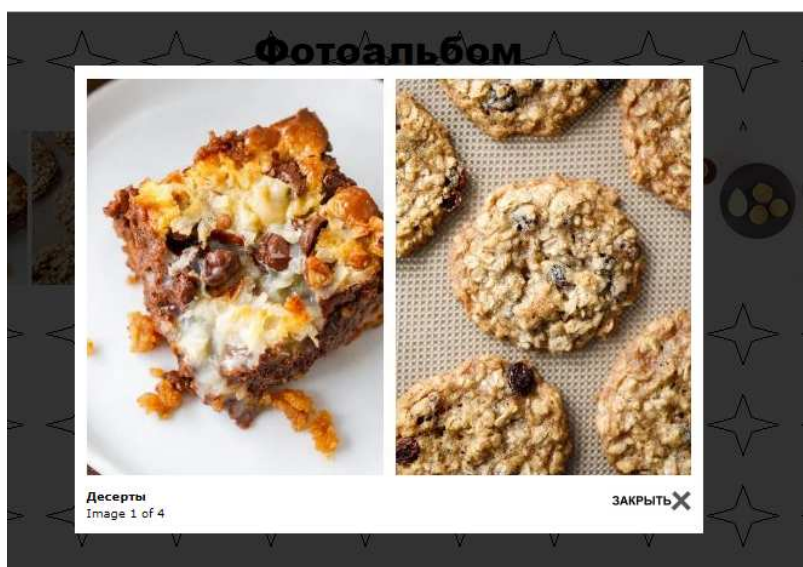


Рисунок 1– Демонстрация выбранного изображения

ЛИТЕРАТУРА

1. Фрейн Бен HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. Питер, 2014 – 298 с.
2. Евсеев Д.А. Web-дизайн в примерах и задачах, уч.пособие для студентов высших учебных заведений – М.:Кнорус, 2016 – 263 с.

УДК 004.42

Студ. К. Н. Кажушко

Науч. рук. доц., к.т.н. А. П. Лащенко
(кафедра информатики и веб-дизайна, БГТУ)

РАЗРАБОТКА САЙТОВ С ИСПОЛЬЗОВАНИЕМ ФРЕЙМОВ

С каждым днем IT-сфера становится шире. Благодаря чему появляются новые способы для создания сайтов и из разметки.

Одним из таких способов является разметка страниц с помощью фреймов. Этот способ требует создания для каждого фрейма отдельного HTML-файла, а также HTML-файла для всей веб-страницы. Отличить в браузере веб-страницу, размеченную фреймами, от страницы, созданной с другими технологиями, довольно трудно. Они различаются лишь специфическими разделительными линиями, которые могут и не отображаться. Наряду с такими линиями, фреймы могут содержать полосы прокрутки для просмотра информации. Размеры создаваемой веб-страницы (например, страницы, содержащей фотографию) не должны превышать размеры фрейма. В противном случае содержимое просто не поместится внутри фрейма, появятся полосы прокрутки.

Чтобы разбить страницу на фреймы, воспользуйтесь тегом `<frameset>`. Всякий раз после применения тега `<frameset>` на веб-странице будет появляться линия, разделяющая ее на две части. Разделительная линия может быть либо горизонтальной, либо вертикальной, горизонтальная линия определяется атрибутом `rows` указанного тега, а вертикальная атрибутом `cols`. В коде страницы с двумя фреймами каждому из указанных атрибутов необходимо присвоить по два значения, определив тем самым ширину `cols` и высоту `rows` фреймов. Значения в коде указываются через пятую.

Но с помощью фреймов веб-страницу можно разделить не только на вертикальные и горизонтальные ячейки. Эти ячейки также можно вкладывать друг в друга. Благодаря чему создаются вложенные фреймы. Пример такой разбивки страницы представлен на рисунке 1.

Код для страницы такого вида будет выглядеть так:

```
<frameset rows="10%, 70%, *">  
<frame src="a1.html">  
<frameset cols="200, *">  
<frame src="a2.html">  
<frame src="a3.html">  
</frameset>  
<frame src="a4.html">  
</frameset>
```

В приведенном коде видно, что сразу задается ширина каждой из строк. Следующим действием вставляется веб-страница в первую

строку при помощи тега <frame>. И затем, вновь используется тег <frameset> для того, чтобы разбить вторую строку на два столбца. Таким образом и создаются вложенные фреймы.

10%	
80%	20%
20%	

Рисунок 1 – Вложенные фреймы

Особое внимание нужно обратить на то, как в коде задана высота верхнего фрейма страницы, которая равна 10%, второго фрейма 70%. Запись, определяющая высоту третьего фрейма, имеет вид «*». Это значит, что высота данного фрейма не фиксируется, и поэтому он будет занимать оставшуюся часть экрана. Такой вариант записи значений атрибута rows в коде считается наиболее удобным. Если указать фиксированное значение высоты, то при различных разрешениях монитора фрейм либо не будет помешаться полностью на экране, либо будет содержать свободное пространство.

Использование фреймов сильно упрощает работу пользователя, поскольку с их помощью можно одновременно просматривать на экране две и более страниц. Это делает использование обычной веб-страницы гораздо удобнее.

УДК 004.42

Студ. Р. В. Раковец

Науч.рук. доц., к.т.н. А. П. Лащенко
(кафедра информатики и веб-дизайна, БГТУ)

ИСПОЛЬЗОВАНИЕ ТАБЛИЦ CSS В HTML-ДОКУМЕНТАХ

Интернет не стоит на месте, это одна из наиболее динамичных областей экономики, и, соответственно развиваются также и стандарты, принятые в этой области.

В настоящее время в веб-разработке CSS практически полностью заменены устаревшие теги, использовавшиеся в первых четырёх спецификациях языка гипертекстовой разметки. Сегодня таблицы стилей позволяют не только отделить контент сайта от его оформления, но и дают возможность применять различные эффекты, которые раньше на сайтах можно было реализовать только при помощи графики.

CSS (CascadingStyleSheets) – язык таблиц стилей, который позволяет прикреплять стиль (например, шрифты и цвет) к структурированным документам (например, документам HTML и приложениям

XML). Обычно CSS-стили используются для создания и изменения стиля элементов веб-страниц и пользовательских интерфейсов, написанных на языках HTML и XHTML, но также могут быть применены к любому виду XML-документа

CSS работает со шрифтами, цветом, полями, строками, высотой, шириной, фоновыми изображениями, позиционированием элементов и многими другими вещами.

С помощью CSS можно:

- управлять отображением практически любого содержимого HTML-документа;

- задать фиксированные размеры какого-нибудь блока на странице;

- задать фиксированные координаты положения блока на странице;

- наложить один блок на другой;

- настраивать параметры отображения содержимого документа,

недоступные при использовании HTML-элементов, и пр.

Для браузеров, не поддерживающих CSS вообще или поддерживающих лишь отдельные правила стилевого оформления, описание шаблонов заключается между символами комментариев.

Код данного элемента выглядит следующим образом.

```
<head>
<title>Ter style</title>
<style type="text/css">
h1{font-size:180%;font-family: Arial;color: #333366;}
</style>
</head>
<body>
<h1>
ВГТУ-2019!
</h1>
</body>
```

Наглядно это будет выглядеть следующим образом (рисунок 1).

Отображено изменение шрифта, цвета и размера. Данная конструкция должна присутствовать в разделе head.

CSS поддерживает таблицы стилей для конкретных носителей, поэтому авторы могут адаптировать представление своих документов к визуальным браузерам, слуховым устройствам, принтерам, брайлевским устройствам, карманным устройствам и т.д.

Каскадные таблицы служат не только для воплощения дизайна, но и кардинально меняют подход к сайтостроению, упрощая труд разработчиков и обеспечивая гибкость реализации.

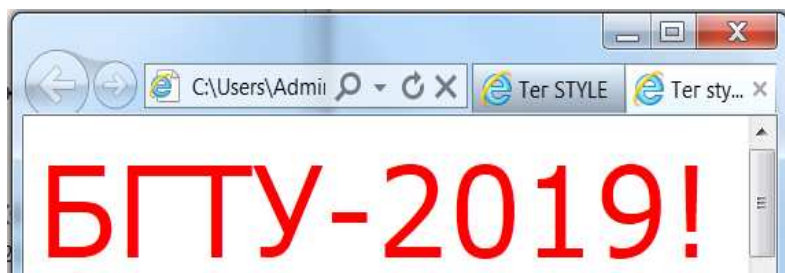


Рисунок 1 –Окно браузера

Основной целью разработки CSS является разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

УДК 004.41

Студ. В.В. Сорока, А.А. Синькевич
Науч. рук. ст. преп. И. А. Миронов
(кафедра информатики и веб-дизайна, БГТУ)

ЭЛЕКТРОННЫЙ КАТАЛОГ РЕЗЮМЕ ВЫПУСНИКОВ

Система «Электронный университет» предназначена для автоматизации управления образовательным процессом, научно-исследовательской деятельностью, финансово-хозяйственными процессами обеспечения жизнедеятельности вуза. А главное обеспечивает их полную интегрированность в информационно-коммуникационную образовательную среду университета. Одним из модулей электронного университета является электронный каталог резюме выпускников. Это интернет-платформа, на которой размещена база резюме старшекурсников вуза. Ее функционирование позволит оптимизировать поиск первого рабочего места молодыми специалистами, упростить процесс взаимодействия «наниматель – кандидат» путем предоставления работодателям детальных данных о потенциальных сотрудниках.

Данный модуль позволит студентам наполнять каталог резюме собственными лабораторными или курсовыми работами на протяжении всего времени обучения.

Целью данной работы является разработка макетов основных страниц и элементов дизайна электронного каталога резюме выпускников.

Задачами проекта является анализ аналогичных решений, разработка структурной схемы, разработка элементов дизайна и макетов основных страниц электронного каталога резюме выпускников.

Для того, чтобы определить какой интерфейс будет максимально удобным, необходимо рассмотреть различные веб-ресурсы для создания резюме. Были рассмотрены три аналогичных решения: ResumeBILD, Craft-CV, CV2you.

Проанализировав данные веб-ресурсы, в качестве основных были выделены следующие разделы: базовая информация, образование, опыт работы, достижения, курсы, языковые навыки, профессиональные навыки, ваш выбор, хобби, а также страницы просмотра резюме и личный кабинет. Также необходимы главная страница и страницы входа и регистрации.

Незарегистрированный пользователь будет иметь возможность регистрации и после нее автоматически становится зарегистрированным пользователем. Зарегистрированный пользователь будет иметь возможность входа в систему, после входа у него появится доступ к личному кабинету и возможность создать резюме. Для регистрации кроме электронного адреса и пароля необходимо будет ввести номер студенческого билета и номер пропуска (рисунок 1). Для входа в систему достаточно будет только электронного адреса и пароля.

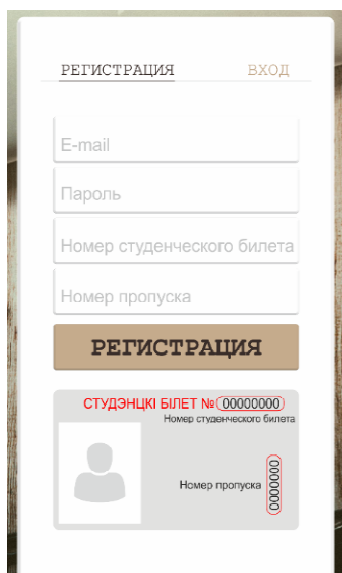
The image shows a registration form with a white background and a thin border. At the top, there are two tabs: 'РЕГИСТРАЦИЯ' (Registration) and 'ВХОД' (Login). Below the tabs are four input fields: 'E-mail', 'Пароль' (Password), 'Номер студенческого билета' (Student ID), and 'Номер пропуска' (Access Pass). Below these fields is a prominent brown button with the text 'РЕГИСТРАЦИЯ'. At the bottom of the form, there is a visual representation of a student ID card. The card is white with a red header that reads 'СТУДЕНЦІ БИЛЕТ № 00000000' and 'Номер студенческого билета'. It features a grey silhouette of a person on the left and the text 'Номер пропуска' followed by a vertical red bar containing the number '000000' on the right.

Рисунок 1– Окно регистрации

При попадании на главную страницу, появляется возможность создать резюме, а также зайти в личный кабинет (рисунок 2).

Создать резюме

ru en

Базовая информация

Имя

Фамилия

E-mail

Мобильный

Город

Дата рождения

Желаемая должность

О себе

Добавить фото

Сохранить

Рисунок 2– Макет «Создать резюме»

Для электронного каталога резюме были разработаны элементы дизайна, которые применялись для обозначения основных разделов. Для разработки использовался графический редактор CorelDRAWX5.

ЛИТЕРАТУРА

1. Миронов, И. А. Интеграция информационных систем БГТУ в единый электронный университет / И. А. Миронов, С. А. Борисевич // Информационные технологии: материалы 83-й научно-технической конференции профессорско-преподавательского состава, научных сотрудников и аспирантов (с международным участием), Минск, 4-15 февраля 2019 г. / отв. за изд. И. В. Войтов; УО БГТУ. – Минск : БГТУ, 2019. – С. 16.

УДК 004.5

Студ. А.А. Марченко
Науч. рук. ст. преп. Н.И. Потапенко
(кафедра информатики и веб-дизайна, БГТУ)

НАВИГАЦИЯ В MATERIALDESIGN 2.0

В мае 2018 года Google представила MaterialDesign 2.0. В связи с переходом на новую дизайн-систему, была выпущена обновленная библиотека дизайна.

Теперь Material Design поддерживает более продвинутую тематизацию, чем просто замена цвета. Можно менять шрифтовую сетку,

скругления элементов интерфейса (точнее, даже их форму – например, можно сделать диагональные скосы), иконки. [1]

Одна из ключевых особенностей MaterialDesign 2.0 – удобная навигация. Навигация – это процесс перемещения между экранами приложения для выполнения некоторых задач (рисунок 1).

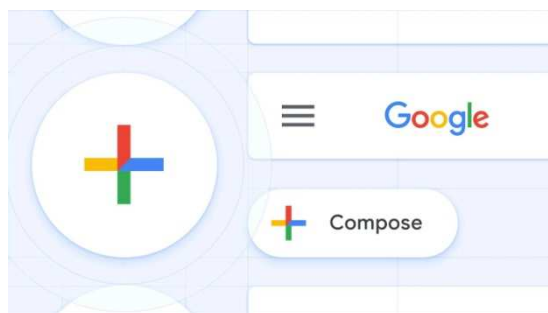


Рисунок 1 – Элементы навигации

Основываясь на информационной архитектуре приложения, пользователь может использовать для продвижения боковую, прямую и обратную навигации.

Рассмотрим и проанализируем особенности каждого из типов навигации.

Боковая навигация (рисунок 2). Под боковой навигацией понимается перемещение между экранами на одном и том же уровне иерархии. Основной компонент навигации приложения должен обеспечивать доступ ко всем адресатам на верхнем уровне его иерархии. Приложения с двумя или более пунктами назначения верхнего уровня могут обеспечивать боковую навигацию через основную панель навигации, нижнюю панель навигации или вкладки.

Прямая навигация – это перемещение между экранами на последовательных уровнях иерархии. Прямая навигация встраивает навигационное поведение в контейнеры (такие как карточки, списки или изображения), кнопки, ссылки или с помощью поиска.

Прямая навигация предусматривает три типа перемещения между экранами для выполнения задачи:

- вниз в иерархии приложения для доступа к более глубокому контенту, от родительского экрана (более высокий уровень иерархии) до дочернего экрана (более низкий уровень);
- последовательно через поток или упорядоченную последовательность экранов, такую, например, как процесс оформления заказа;
- прямо с одного экрана на любой другой в приложении, например, с домашнего экрана на экран в глубине иерархии приложения.

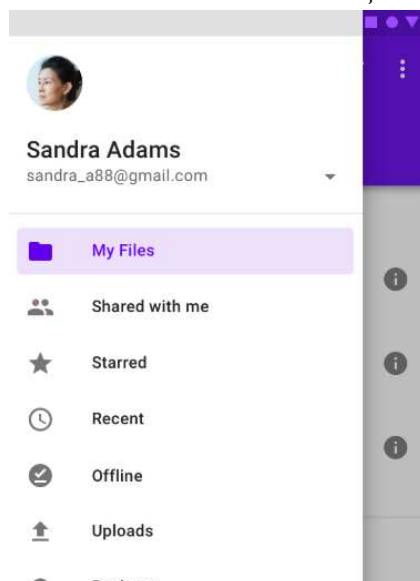


Рисунок 2 – Пример боковой навигации

В то время как боковая навигация использует выделенные навигационные компоненты, прямая навигация часто встроена в содержимое экрана с помощью различных компонентов (рисунок 3). Она может быть реализована с использованием:

- контейнеров содержимого, таких как карточки, списки или списки изображений;
- кнопок, которые переходят на другой экран;
- поиска в приложении на одном или нескольких экранах;
- ссылок внутри контента.

Обратная навигация относится к движению назад между экранами. Она может перемещать пользователей в хронологическом порядке по их недавней истории посещения страниц/экранов или вверх по иерархии приложения.

Обратная хронологическая навигация относится к навигации в обратном порядке по истории недавно просмотренных страниц / экранов. Она может перемещать пользователей между экранами в приложении или между несколькими приложениями. Например, кнопка «Назад» в веб-браузере является формой обратной хронологической навигации. Этот тип навигации обычно предоставляется операционной системой или платформой. Отдельные платформы определяют, как она ведет себя и как пользователи могут получить доступ к этим функциям (рисунок 4). Навигация вверх позволяет пользователям перемещаться на один уровень вверх в иерархии одного приложения до тех пор, пока не будет достигнут домашний экран или экран верхнего уровня приложения. Например, стрелка вверх на верхней панели приложения представляет собой форму обратной навигации вверх.

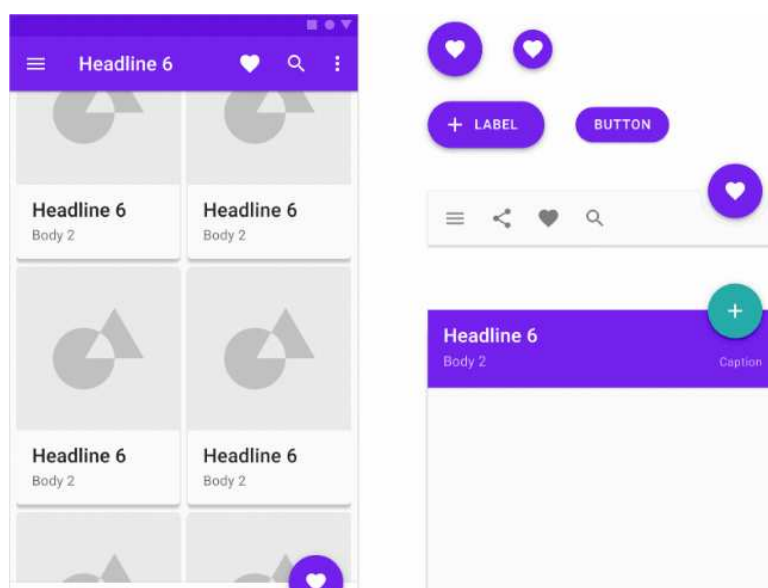


Рисунок 3 – Элементы прямой навигации

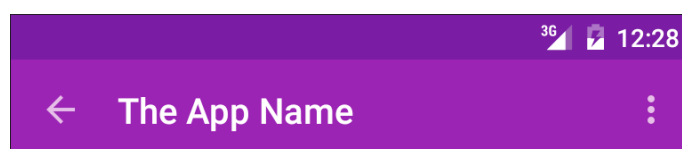


Рисунок 4 – Кнопка «Назад» в мобильном приложении

Дизайн и функциональность приложений должны учитывать оба вида обратной навигации на целевых платформах разрабатываемых приложений. Чтобы оптимизировать работу пользователя при навигации в обратном направлении следует:

- вернуть пользователей в их прежнее положение и состояние экрана, например, в положение вертикальной прокрутки, чтобы ускорить возврат информации и возобновление задачи;
- обеспечить четкую передачу сообщений, если предыдущее состояние экрана больше не доступно, например, когда информация из формы была очищена для конфиденциальности;
- четко указать отношения дочерних экранов с экранами над ними в иерархии. Например, если пользователь переходит непосредственно к дочернему экрану в вашем приложении, он должен иметь возможность идентифицировать родительский экран, к которому он может перейти вверх.

ЛИТЕРАТУРА

1. «Хабр» [Электронный ресурс] / Ресурс для IT-специалистов – Режим доступа: <https://habr.com/ru/company/mailru/blog/358246/> – Дата доступа: 23.03.2019.

Студ. К. В. Тозик
Науч. рук. ассист. С. А. Осоко
(кафедра информатики и веб-дизайна, БГТУ)

ЗАЩИТА ИНФОРМАЦИИ В ОТКРЫТЫХ ОБЛАЧНЫХ ХРАНИЛИЩАХ

Облачное хранилище данных – модель онлайн-хранилища, в котором данные хранятся на многочисленных распределённых в сети серверах, предоставляемых в пользование клиентам в основном третьей стороной. Облачными хранилищами являются такие интернет-сервисы как: Dropbox, OneDrive, GoogleDrive, iCloud, Яндекс Диск, облако Mail.Ru [1].

У данных интернет сервисов есть много плюсов, например, возможность доступа к данным с любого компьютера, имеющего выход в Интернет, возможность организации совместной работы с данными, высокая вероятность сохранения данных даже в случае аппаратных сбоев и другие. Однако, главный вопрос, возникающий при использовании данных облачных хранилищ это насколько безопасно хранить данные в таких интернет-сервисах. В 2011 году одна аналитическая фирма провела исследования в данной области, которое показало, что 68% опрошенных руководителей европейских IT-компаний в целях безопасности отказываются использовать облачные технологии из-за того, что данные могут стать достоянием общественности [2–3]. Например, провайдер имеет возможность просматривать данные клиента (если они не защищены паролем), которые так же могут попасть в руки хакеров, сумевших взломать системы защиты провайдера. Так же известны несколько инцидентов, связанных с утечкой данных с облачных хранилищ. Так как же можно тогда защитить свою информацию в открытых облачных хранилищах?

Цель работы: рассмотреть, сравнить и проанализировать способы защиты информации при хранении в открытых облачных хранилищах. Наверно, один из самых простых способов, это создание запароленного архива с помощью программ WinRAR и 7-Zip. Каким образом это сделать? Порядок действий в этих программах практически не отличается. При использовании WinRAR для начала необходимо выбрать папку, которая будет помещаться на облачном хранилище, нажать правой кнопкой мыши, в появившемся списке выбирается «Добавить в архив». В открывшемся окне выбирается «Установить пароль». Устанавливается пароль и далее, при открытии данного архива и разархивировании папки будет запрашиваться пароль.

При использовании 7-Zip так же выбирается папка и нажимается по ней правой кнопкой мыши. В появившемся списке выбирается «Добавить в архив». На следующем шаге необходимо найти раздел «Шифрование», который показан на рисунке 1. В поле под цифрой 1 вводится пароль, в поле под цифрой 2 подтверждается этот пароль и нажимается кнопка «ОК». В результате чего появляется заархивированная папка, при открытии или разархивировании которой так же будет запрашиваться пароль. Данный способ, наверно, один из самых простых и легкодоступных. Однако не самый удобный.

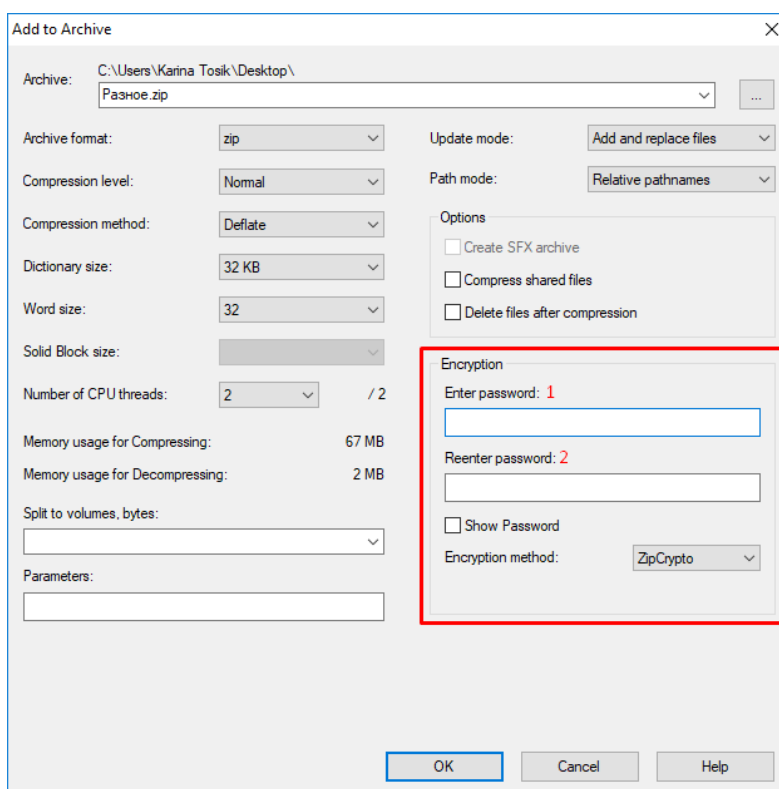


Рисунок 1 – Окно шифрования в программе 7-Zip

Имеется программное обеспечение SpiderOak, которое разработано компанией SpiderOakInc., которое предназначено для резервного копирования пользовательских данных и дальнейшего хранения на «облаке». Информация предоставляется на сервер только в зашифрованном виде, а пароль не передаётся никогда, кроме тех случаев, когда пользователь работает с веб-интерфейсом. Также имеется возможность синхронизации файлов и папок на нескольких устройствах одновременно и автоматического устранения дублирования файлов. Данная программа использует многоуровневый подход к шифрованию, используя комбинацию 2048 бит RSA и 256-битное AES. История изменения файлов ведётся по принципу дельта-кодирования, что-

бы сэкономить место, занимаемое файлами. В истории изменения записывается только отличие версии файла от другой.

Отличительными возможностями данной программы является следующие: экономия места в хранилище и времени выгрузки файлов за счёт дедупликации и внесения изменений в уже имеющиеся в хранилище файлы (вместо перезаписи файлов целиком). Далее настраиваемая мульти платформенная синхронизация. DropBox для синхронизации создаёт специальную папку, в которую надо помещать все синхронизируемые файлы. SpiderOak может работать с любым каталогом; сохранять все хронологические версии файлов и удалённых файлов; получать файлы с любого подключенного к Интернету устройства с полным шифрованием данных по принципу «нулевого знания»; поддерживать неограниченное количество устройств. В качестве плюсов у данной программы можно выделить то, что для взаимодействия с сервисом данные передаются в открытом виде (не шифруются на своей стороне). Поэтому все данные не будут доступны для всеобщего обозрения в случае неполадок, что уже случалось в некоторых альтернативных вариациях облачных технологий. На стороне клиента происходит полное шифрование всех данных. Отсутствие длительных регистраций, регистрация есть, но она сокращена до минимума и не подразумевает длительное заполнение различных форм. Так же у этой программы много других плюсов, которые можно узнать в сети Интернет.

Конечно, говоря о защите информации на облачных хранилищах, не стоит забывать об аутентификации. Защита паролем для обеспечения более высокой надежности. Наиболее простой и достаточно надежный метод аутентификации – это технология одноразовых паролей (OneTimepassword, OTP). Такие пароли могут генерироваться либо специальными программами, либо дополнительными устройствами, либо сервисами, с пересылкой пользователю по SMS.

Ещё один метод защиты информации в облачных хранилищах это изоляция пользователей. Использование индивидуальной виртуальной машины и виртуальной сети. Виртуальные сети должны быть развёрнуты с применением таких технологий, как VPNи VPLS. Часто провайдеры изолируют данные пользователей друг от друга за счёт изменения кода в единой программной среде. Этот подход имеет риски, связанные с опасностью найти дыру в нестандартном коде, позволяющем получить доступ к данным. В случае возможной ошибки в коде пользователь может получить доступ к информации другого пользователя. Последнее время такие инциденты часто имели место.

Таким образом, можно выделить 4 наиболее популярные метода защиты информации в «Облачных технологиях»:

1. Шифрование.
2. Защита данных при передаче.
3. Аутентификация.
4. Изоляция пользователей.

В заключение следует сказать, что безопасность не всегда обеспечивается только защитой. Она может быть достигнута также соответствующими правилами поведения и взаимодействия объектов, надёжность всех видов обеспечения функционирования объектов информационной безопасности.

ЛИТЕРАТУРА

1. Сравнение облачных хранилищ – самые популярные облака: <https://www.boxcryptor.com/ru/blog/post/list-best-clouds-private-use/>.
2. Защита информации в «Облачных технологиях» как предмет национальной безопасности: <https://moluch.ru/archive/86/16357/>.
3. Безопасность облачных хранилищ: <https://moluch.ru/conf/tech/archive/286/13236/>.

УДК 004.451.642

Студ. В. В. Назаренко
Науч. рук. ст. преп. Т. В. Кишкурно
(кафедра информатики и веб-дизайна, БГТУ)

ТЕМНЫЕ ПАТТЕРНЫ: ИСПОЛЬЗОВАНИЕ UX ДИЗАЙНА ДЛЯ ОБМАНА ПОЛЬЗОВАТЕЛЕЙ

Опыт пользователей прошел большой путь за последнее десятилетие, особенно в мире веб-дизайна. Сейчас, как никогда ранее, люди больше сосредоточены на том, чтобы убедиться, что пользователю не только легко, но и нравится продукт или услуга, которые они используют. Но пока большинство людей работают над созданием более удобной и приятной для пользователя сети, некоторые работают, чтобы обмануть пользователя и обмануть его с помощью методов UX, известных как темные паттерны [1].

Темные Паттерны – это пользовательские интерфейсы или методы взаимодействия с пользователем, разработанные специально для того, чтобы обмануть людей. Эти методы могут быть относительно безвредными и оставить только чувство раздражения у пользователя. Другие, однако, могут стоить пользователю гораздо больше в финансовом или даже профессиональном плане. Хотя их не следует путать с антипаттернами, которые являются обычной практикой, приводящей к плохому UX (и не преднамеренно вводящей в заблуждение).

По сути, основная цель Темного Паттерна состоит в том, чтобы заставить пользователя купить или сделать что-то, чего он не сделал бы иначе. Nielsen Norman Group имеет очень популярный пост по эвристике дизайна. Следуя этой эвристике, в программном обеспечении принято, чтобы кнопки «ОК» и «Отмена» были согласованными. В противном случае это может вызвать замешательство или раздражение – что является плохим UX. Замена кнопки «ОК» время от времени на кнопку «Купить еще 1000 приложений за \$ 0,99» будет считаться темным паттерном [2].

Цель работы: проанализировать темные паттерны и рассмотреть методы разработки удобного для пользователя дизайна интерфейса. Существуют следующие виды темных паттернов: приманка и переключение, подтверждение с чувством стыда, дезориентация и подписка «по принуждению», нагнетание страха, принудительное продление, спам друзей, скрытая стоимость, смена фокуса, предотвращение сравнения цен, и другие.

Приманка и переключение используются, когда на странице, занимающей высокие позиции в поиске, заменяется контент. Этот темный UX паттерн сначала показывает предупреждение, затем подсказывает пользователю какие нужно совершить действия и после того, как эти действия будут предприняты, предоставляет совершенно другие результаты, чем ожидалось.

Microsoft использовал в прошлом этот темный паттерн – появлялось всплывающее окно (рисунок 1).



Рисунок 1– Использование Microsoft темных паттернов

Как обычно, если вы не хотите запускать обновление, вы нажимаете «X» в правом верхнем углу, что фактически запускало процесс обновления операционной системы до Windows 10. А кнопка «ОК» приумножала обман, так как она стояла рядом с «Upgrade Now». При этом обе они приводили к одному и тому же результату (согласие на обновление).

Позор и чувство вины. Говоря об этике, как вы думаете, является ли хорошей идеей заставлять посетителей сайта чувствовать себя плохо из-за того, что они не подписались на рассылку или не загрузили вашу последнюю электронную книгу? Пример особенно плохого вкуса приведен на рисунке 2. Из-за симпатичного медведя со слезой на глазах и еще более вопиющей ссылки с текстом «Я плохой человек» (также намеренно не выделенного жирным шрифтом или красным цветом дизайнерами сайта) [3].

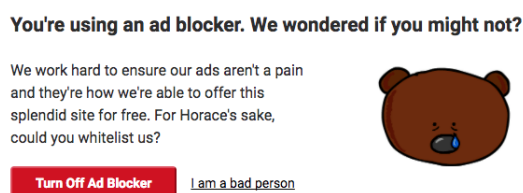


Рисунок 2 – Пример паттерна «Позор и чувство вины»

Дезориентация и подписка «по принуждению» – работает так же, как фокусник отвлекает внимание от того, что он или она действительно делает за кулисами, чтобы сосредоточить внимание на иллюзии. Введение в заблуждение достигается за счет использования ярких цветов, скрытого дизайна и запутанной формулировки. Подписка «по принуждению» насколько возможно усложняет отмену подписки, деинсталляцию и выход, одновременно упрощая продолжение платежа (или платеж даже остается незамеченным, пока клиенты не проверят свои счета).

Нагнетание страха (запугивание). Наиболее актуальные и узнаваемые примеры этого можно найти у разработчиков антивирусных программ. Заставлять людей загружать программное обеспечение из страха – не совсем элегантно и несправедливо. Это можно сравнить с видами страхования, которые не обязательны по закону: мы покупаем их «на всякий случай». Некоторые компании играют по этому сценарию и описывают на сайте «что может случиться», вселяя страх в посетителей:

Скрытая стоимость – другая (теперь незаконная в ЕС) тактика, которая остается широко распространенной в США и за ее пределами – неожиданные издержки, которые появляются в корзине потребителя в самом конце долгой транзакции. Что еще хуже, приходится внимательно изучить расчеты, чтобы разобраться в новых неожиданных цифрах.

Разрабатывая дизайн своих интерфейсов, помните фразу от мастера в UX: «Использовать людей – низко. Хороший бренд нравится людям. Отличный бренд любят и уважают. Вы никогда не достигнете этого, если будете использовать темные паттерны».

ЛИТЕРАТУРА

1. Resisting the dark side: a primer on dark pattern ux [Электронный ресурс]. – 2019. – Режим доступа: <https://www.webdesigner-depot.com/2013/11/resisting-the-dark-side-a-primer-on-dark-pattern-ux/>. – Дата доступа: 08.04.2019.
2. DarkPattern [Электронный ресурс]. – 2010. – Режим доступа: <https://darkpatterns.org> – Дата доступа: 05.04.2019.
3. Темные паттерны в дизайне или просто мудаккий дизайн [Электронный ресурс]. – 2018. – Режим доступа: <https://ux.pub/temnye-patterny-v-dizajne-ili-prosto-mudackij-dizajn/>. – Дата доступа: 12.04.2019.

УДК 004.043

Студ. В.Ю. Булохова, В.В. Сорока
Науч. рук. ст. преп. С.А. Борисевич
(кафедра информатики и веб-дизайна, БГТУ)

ПРИМЕНЕНИЕ СЕМАНТИЧЕСКИХ ТЕГОВ HTML5 <SECTION> И <ARTICLE>

Теги, которые используются в HTML, в большинстве своём предназначены для форматирования контента. Они дают инструкции браузеру как отображать контент на странице, но не дают определение типу содержащегося контента или какую роль играет контент на странице. Семантические теги в HTML5 устраняют этот недостаток, определяя четкую роль контента на странице. Эта дополнительная информация помогает роботам/индексаторам, таким как Google и Яндекс лучше понять, какой контент важен, какой является второстепенным, какой используется для навигации и так далее. Добавляя семантические HTML теги на страницы, мы даем дополнительную информацию, которая помогает поисковым сервисам понимать роли и относительную важность разных частей наших страниц. Цель работы: рассмотреть пути реализации семантической разметки для улучшения позиций сайта в выдаче поисковых систем.

Примером не семантических HTML элементов являются теги <div> и . Они являются универсальным контейнером для потокового контента, и не дают информации о роли своего содержимого на странице. Примером семантических HTML элементов являются теги: <header>, <footer>, <nav>, <article>, <main>, <section>, <aside>. Они ясно определяют роль своего содержимого.

Для пользователя обычно легко определить различные части веб-страницы с первого взгляда. Заголовки, меню и основной контент □ –

все мгновенно, визуально очевидно. Для поисковых роботов или программ чтения с экрана на странице, размеченной блочными элементами `<div>`, сложно определить какой из блоков является основным, а какой второстепенным и им в этом нужна помощь. Если мы сможем успешно передать поисковым системам, какая часть страницы является хедером, какая подвалом, какая навигацией, а где основной контент, то это сделает страницу гораздо понятнее и легче для правильной индексации Google и Яндекс. Понимание поисковыми системами контента страницы позволит улучшить ее позиции на странице выдачи SERP.

Размеченная семантическими тегами `<article>` и `<section>` страница отображается браузерами так же, как если бы она была размечена тегами `<div>`. Данный факт позволяет заменить существующие блочные теги `<div>` на семантические теги без изменения общего макета. То есть в большинстве случаев применение семантического HTML5 может быть легко применено нахождением подходящей пары `<div>` и `</div>` и её заменой.

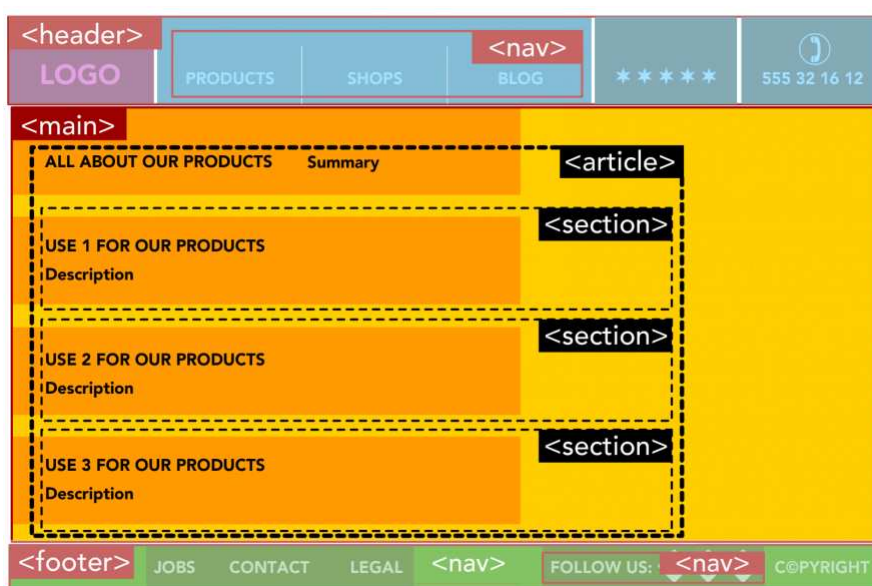


Рисунок 1 – Пример разметки страницы

На рисунке 1 приведен пример разметки страницы с помощью семантических тегов. Главная часть страницы выделена с помощью тега `<main>`, что позволяет роботам поисковых систем и программам считывания с экрана однозначно ее отделить от блоков навигации, рекламных блоков и т.д. Внутри тега `<main>` находится тег `<article>` который определяет независимую, самодостаточную часть страницы или документа. Это означает, что передаваемый содержимым этого тега смысл будет восприниматься так же, если его разместить в другом месте страницы. Если содержимое тега `<article>` имеет несколько подтем, которые формируют основную тему, то их следует размечать

тегами <section>. Главное правило: <section> формирует часть чего-то ещё, а <article> это что-то, что само по себе. На каждом этапе построения структуры страницы, ее можно проверять при помощи онлайн инструмента Outliner (<https://gsnedders.html5.org/outliner/>).

ЛИТЕРАТУРА

1. Борисевич, С. А. Семантическая разметка для оптимизации сайтов под поисковые системы / С. А. Борисевич, И. А. Миронов // Информационные технологии: тезисы докладов 82-й научно-технической конференции профессорско-преподавательского состава, научных сотрудников и аспирантов (с международным участием), Минск, 1-14 февраля 2018 г. / Белорусский государственный технологический университет. – Минск : БГТУ, 2018. – С. 69–70.

2. Дженнифер Нидерст Роббинс «HTML5, CSS3 и JavaScript. Исчерпывающее руководство». 4-ое изд., 2014.

УДК 339.138

Студ. Я.С. Крушинский

Науч. рук. ст. преп. Е.С. Мирончик

(кафедра информатики и веб-дизайна, БГТУ)

СОЗДАНИЕ ИНТЕРНЕТ-МАГАЗИНА В СОЦИАЛЬНОЙ СЕТИ ВКОНТАКТЕ

Интернет-магазин ВКонтакте открывает ряд преимуществ как для покупателей, так и для продавца. Первым не нужно заходить на сайты в поисках товара, есть возможность быстро и удобно общаться с продавцом и уточнять детали, а также оставлять отзывы (это плюс для других покупателей – перед покупкой можно почитать, что пишут о товаре).

Для продавца создать магазин в ВКонтакте с нуля легче и быстрее, чем открывать сторонний ресурс, также не нужно покупать домен и решать проблемы с хостингом. Для оценки эффективности работы можно использовать такие простые инструменты, как репосты и лайки, количество подписчиков. Если их мало, необходимо предпринимать меры для изменения текущей ситуации. К тому же, привлечь посетителей на магазин в ВК из социальных сетей гораздо проще, чем на отдельный сайт.

Сегодня ВКонтакте является популярной социальной сетью, которая объединила больше отечественных подписчиков, чем Фейсбук и Одноклассники. Следовательно, интернет-магазин ВКонтакте позволит достичь желаемой цели.

Неограниченное количество возможностей предоставляет использование VK.apі.API–это посредник между разработчиком приложений и какой-либо средой, с которой это приложение должно взаимодействовать. API упрощает создание кода, поскольку предоставляет набор готовых классов, функций или структур для работы с имеющимися данными.API ВКонтакте – это интерфейс, который позволяет получать информацию из базы данныхvk.com с помощью http-запросов к специальному серверу. Вам не нужно знать в подробностях, как устроена база, из каких таблиц и полей каких типов она состоит – достаточно того, что API-запрос об этом «знает». Синтаксис запросов и тип возвращаемых ими данных строго определены на стороне самого сервиса. Также можно сказать, что VK.apі–является открытым apі,и это возможно применить в интернет магазине.

Рассмотрим возможности ВКонтакте на примере реализации интернет-магазина «Штиль», который специализируется на реализации футболок, худи и свитшотов с дизайнерскими пинтами.

В 2015 году ВКонтакте запустила новый сервис для розничной торговли. «Товары ВКонтакте» – это витрина с товарами. Чтобы сделать заказ, нужно связаться с продавцом (сообщение со ссылкой на товар можно отправить со страницы этого товара).С помощью Эквиды можно добавит в свой каталог товаров ВКонтакте функционал полноценного интернет-магазина – автоматический заказ и оплату.

Важным этапом создания интернет-магазина является разработка уникального дизайна. Именно дизайн позволяет пользователю быстро понять назначение магазина, сформировать первичное представление о товаре, а также стимулирует вступить в группу.

Для интернет-магазина «Штиль» был разработан логотип, представленный на рисунке 1. Его дизайн должен создать солнечное летнее настроение и атмосферу отдыха на море, пляжа, серфинга. Далее эта тема будет поддерживаться и в дизайне других основных элементов страницы: обложки группы, миниатюры группы, баннеров группы, меню, обложек товаров, альбомов, а также шаблонов для постов.



Рисунок 1 – Логотип

Рассмотрим пример реализации работы с VK.арина боте, который был написан на языке программирования Python.

Задача бота понизить загруженность менеджера по продажам, ускорить время реагирования на вопросы пользователя, что позволит удержать его интерес к товару и укорит принятие решения о покупке. Разработанный бот способен отвечать на основные наиболее часто задаваемые вопросы. Примеры работы бота показаны на рисунке 2.

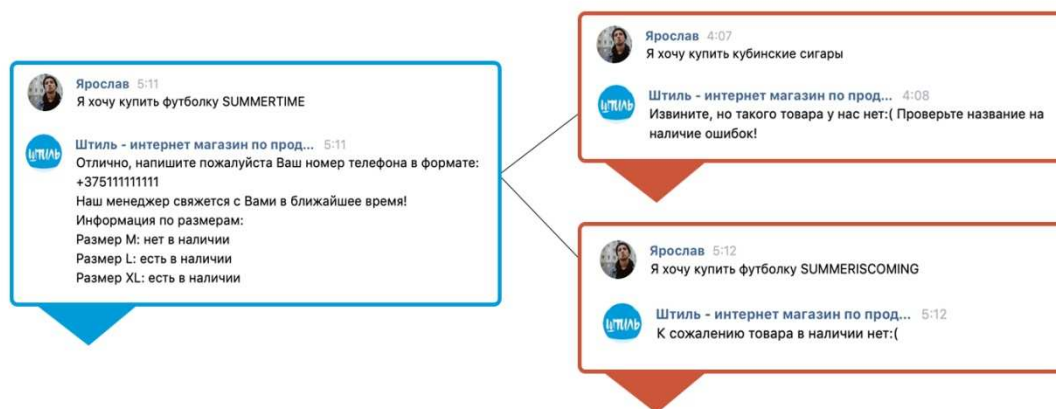


Рисунок 2 – Пример работы бота

Также был разработан парсер для отслеживания вирального контента у конкурентов, который можно использовать в своем продукте для привлечения нового трафика. В итоге получен функционирующий интернет-магазин с уникальным дизайном, каталогом товаров, возможностью консультирования с помощью бота, а также парсер для анализа контента конкурентов.

УДК 004.9, 004.438

Студ. Ю. Н. Ратайко
Науч. рук. доц. О. А. Новосельская
(кафедра информатики и веб-дизайна, БГТУ)

ИГРОВОЙ ПОРТАЛ «THE SECRET OF THE PAST»

Игровая индустрия меняется каждый год. Стремительно развивается, осваивает новые просторы и пытается перепрыгнуть через саму себя в своём стремлении создать что-то новое и уникальное, потрясти попробовавших всё игроков [1].

Цель проекта в том, чтобы создать игру, которая будет удивлять своей реалистичной графикой, чтобы игроку хотелось просто побегать и понаблюдать за фантастическим миром. Чтобы сюжет заинтересовал пользователя и побудил его скачать игру, а после прохождения оставил эмоции и вопросы, которые ещё какое-то время не будут давать покоя.

Игра «The secret of the past» обладает рядом достоинств:

- игра обладает захватывающим и уникальным сюжетом;
- управление организовано от третьего лица;
- игра фэнтези, а не боевик, что очень редко встречается на рынке;
- красивое музыкальное и звуковое сопровождение;
- анимация и эффекты.

Для оформления сцены были созданы модели архитектуры, подобраны модели растений, окружающей среды, была создана поверхность воды, небо, освещение, различные системы частиц: огонь, туман, свечения (рисунок 1).



Рисунок 1 – Пример использования системы частиц

Из всего вышперечисленного было составлена композиция, которая составляет активную зону, где игрок может ходить, и фоновую зону (рисунок 2). Сцена разрабатывалась с учётом того, что на ней живут и люди, и драконы, поэтому мы видим, как и массивные стены и камни, так и обычные дома.



Рисунок 2 – Скриншот сцены 1

Перед разработкой проекта была спроектирована логика игры «The secret of the past» (рисунок 3).

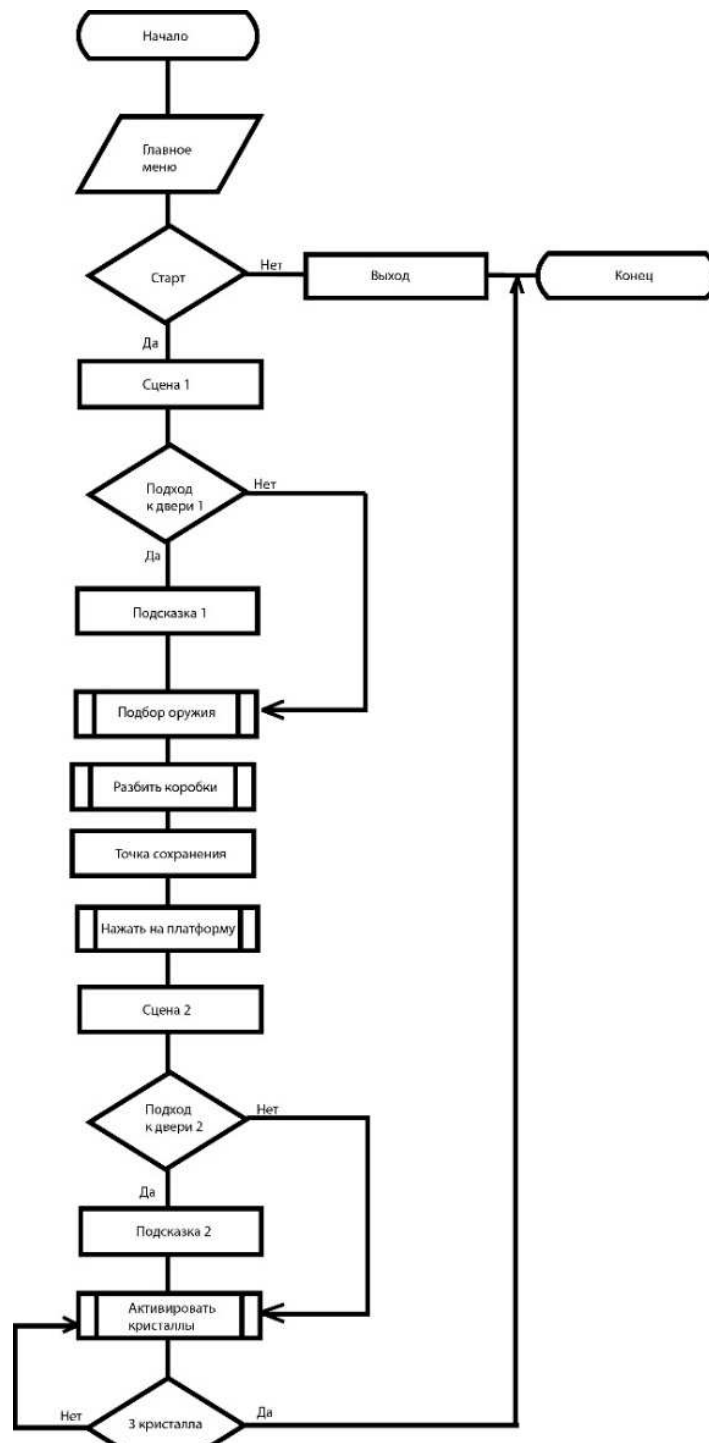


Рисунок 3 – Логическая схема игры

Игра разрабатывалась на движке Unity с использованием языка программирования с#. Анимирование персонажей происходило по-кадрово в среде 3dsMAX.

Эффект ветра создавался в Unity с помощью наложения эффектов на каждую двигающуюся от «ветра» модель. Пример представлен на рисунке 4– набор текстур, дающий нам минимальные колыхания.

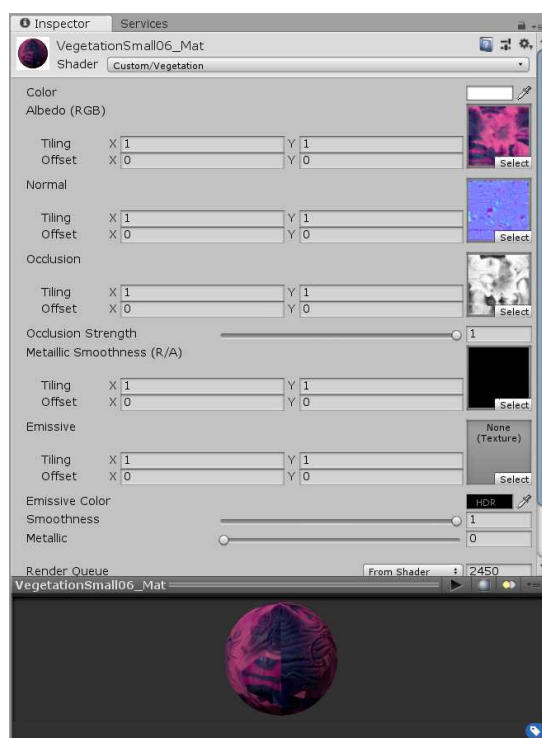


Рисунок 4– Набор текстур для создания эффекта «ветер»

Функции открытия дверей, подбора элементов обрабатываются при событиях `OnTriggerEnter`, `OnCollisionStay`, `On Trigger Exit` и т.д. [2, 3]. Для работы меню были разработаны панели и скрипты, которые висят на панелях, при нажатии на кнопку, мы устанавливаем переменную `Visible` в `true` или `false` для нашей панели. Также на каждой кнопке висит аудиофайл, который запускается при наведении. Громкость звука во всей игре можно регулировать в меню, для этого был создан скрипт на языке `C#`. К игре разработан свой сайт.

ЛИТЕРАТУРА

1. 13 основных принципов геймплейного дизайна. [Электронный ресурс]. – 2017–2018 – Режим доступа: https://www.gamasutra.com/view/feature/132341/the_13_basic_principles_of_.php. – Дата доступа: 21.04.2019.

2. Guiding Players through Structural Composition Patterns in 3D Adventure Games. [Электронный ресурс]. – 2008–2019 – Режим доступа: http://www.fdg2014.org/papers/fdg2014_paper_35.pdf. – Дата доступа: 18.04.2019.

3. Каталог игр. [Электронный ресурс]. – 2008-2018 – Режим доступа: <https://steamray.com/>. – Дата доступа 19.04.2019.

FLUTTEL – ФРЕЙМВОРК ДЛЯ КРОСС-ПЛАТФОРМЕННОЙ РАЗРАБОТКИ

Flutter SDK – молодая многообещающая платформа. 4 декабря 2018 года, прошла презентация версии 1.0 – базируясь как набор инструментов от Google для создания нативных приложений iOS и Android из одной кодовой базы. На данный момент Flutter поддерживает мобильные операционные системы: Android Jelly Bean, v16, 4.1.x или новее и iOS 8 или новее. Платформа интересна своей простотой, схожестью с разработкой веб-приложений, скоростью работы наравне с нативными приложениями и высокой производительностью, достигнутой за счет нескольких техник [1–3]. В отличие от многих мобильных платформ, Flutter не использует JavaScript. Он построен на C, C++, Dart и Skia (движок 2D-рендеринга). Язык программирования для Flutter – Dart, который компилируется в бинарный код, за счет чего достигается скорость выполнения операций сравнимая с Objective-C, Swift, Java, или Kotlin. Flutter не использует нативные компоненты, следовательно, нет необходимости писать прослойки для коммуникации с ними. Вместо этого, подобно игровым движкам, кнопки, текст, медиа-элементы, фон – все это отрисовывается внутри графического движка самого Flutter. При этом вес приложения невелик: «Hello, World» приложение занимает: iOS $\approx 2.5\text{Mb}$ и Android $\approx 4\text{Mb}$.

Для построения UI во Flutter используется декларативный подход, схожий с веб-фреймворком ReactJS, на основе виджетов (аналог веб-компонентов). Flutter поставляется с набором мощных базовых виджетов, из которых очень часто используются следующие: Text, Stack, Row. Так же можно создать свои виджеты унаследовав от StatelessWidget или StatefulWidget, в зависимости от того, управляет ли ваш виджет каким-либо состоянием. Для еще большего прироста в скорости работы интерфейса виджеты перерисовываются по необходимости – только когда в них что-то изменилось, подобно тому как это делает Virtual DOM веб-фронтенда.

Также в фреймворк встроен Hot-reload. Горячая перезагрузка работает путем внедрения обновленных файлов исходного кода в работающую виртуальную машину Dart.

Flutter оптимизирован для 2D мобильных приложений. Можно создавать полнофункциональные приложения, включая камеру, геолокацию, сеть, хранилище, сторонние SDK. Flutter работает с редакторами AndroidStudio, IntelliJIDEA и VSCode. На рисунке 1 представлен пример создания Виджета данной платформы.

```

import 'package:flutter/material.dart';

void main() {
  runApp(new SampleApp());
}

class SampleApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: 'Sample App',
      theme: new ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: new SampleAppPage(),
    );
  }
}

class SampleAppPage extends StatefulWidget {
  SampleAppPage({Key key}) : super(key: key);

  @override
  _SampleAppPageState createState() => new _SampleAppPageState();
}

class _SampleAppPageState extends State<SampleAppPage> {
  // Default placeholder text
  String textToShow = "I Like Flutter";

  void _updateText() {
    setState(() {
      // update the text
      textToShow = "Flutter is Awesome!";
    });
  }
}

```

Рисунок 1 – Листинг SimpleApp

И так, что же находится внутри Flutter SDK?

- оптимизированный движок 2D-рендеринга с поддержкой текста;
- богатый набор виджетов для Android и iOS;
- API для модульных и интеграционных тестов;
- интерфейс и плагин API для подключения к системе сторонних SDK;
- запуск тестов в Windows, Linux и Mac;
- инструменты командной строки для создания, сборки, тестирования и компиляции приложений.

ЛИТЕРАТУРА

1. Электронное издание «Flutter Documentation»
<https://flutter.dev/docs/cookbook>.

2. Электронное издание «Flutter Git Open Source»
<https://github.com/flutter>.

3. Электронное издание «Про Flutter, кратко» <https://habr.com/ru/post/430918/>.

О ПРОБЛЕМЕ ТЕМНЫХ ПАТТЕРНОВ В ВЕБ-ДИЗАЙНЕ

Интернет стал неотъемлемой частью жизни современного человека. Преувеличить его роль сложно, ведь он заменил человеку большой спектр сред: работу, общение, досуг и другое. Специалисты различных направлений трудятся над тем, чтоб всемирная сеть была интересна и востребована. К примеру, дизайнеры UX рассматривают проблемы, касающиеся вопросов о том, «почему», «для чего» и «как» использовать продукт, будь то веб-сайт или мобильное приложение.

Открыв тот или иной сайт, пользователь хочет лишь одного – достичь свою цель. Основная задача UX-дизайнеров – помочь пользователю в этом, концентрируясь на таких аспектах пользовательского опыта, как удовольствие, удобство, эффективность и др. [1]. Тем не менее, довольно часто происходит ситуация, в которой пользователь «застревает» в Интернете, позабыв о том, зачем он зашел, и совершая действия, которые не планировал. В таких случаях напрашивается вывод – UX-дизайнер плохо работал или был ленив. Неудобный интерфейс – это не всегда результат неопытности его создателя, иногда это тёмный паттерн, который создан для того, чтобы быть «нечестным» по отношению к пользователю. Тёмные паттерны не являются ошибками. Они досконально продуманы с чётким пониманием психологии человека, и они не учитывают интересов пользователя. Существуют различные виды тёмных паттернов, каждый из которых имеет свои особенности, призван достигнуть определенных результатов для компании и по-своему влияет на пользователя. Цель данной работы – проанализировать интерфейсы некоторых сайтов на присутствие темных паттернов.

Всемирно известная система интернет-бронирования отелей Booking.com представляет собой кладезь «темных паттернов», что не удивительно, ведь ее главная задача – получение прибыли любой ценой. Предположим, что пользователь хочет забронировать отель у моря на несколько дней. Он вводит необходимую информацию для поиска и получает результат – список подходящих под его критерии мест. Каждый элемент результата имеет одинаковое оформление. Интерес представляет строка, указывающая стоимость. Ярко зеленым цветом на экран выведена некоторая сумма, а ниже незаметным серым цветом и мелким шрифтом дополнительная стоимость, которую придется заплатить. Это можно увидеть на рисунке 1, демонстрирующем часть скриншота страницы. Таким образом, цена в 452, на кото-

рую среагировал пользователь, превращается в 496, что уже не так приятно. Данный пример можно отнести к виду «скрытая стоимость».

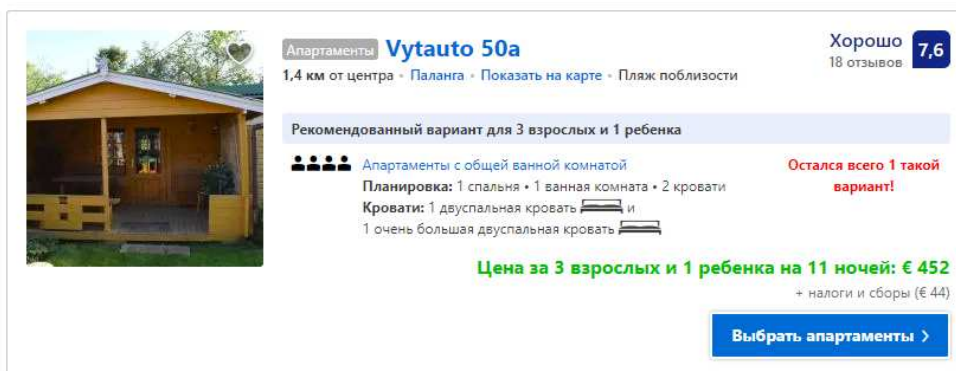


Рисунок 1 –Скриншот страницы сайта booking.com

Графа «налоги и сборы» имеется у каждого варианта и составляет в 90% случаев не более 50 евро. Предположим, что пользователь уже принял данный «недочет» и, просматривая результаты, автоматически прибавляет к стоимости 50 евро для получения конечной стоимости. Это, конечно, неудобно и усложняет пользователю работу на сайте. Однако, разработчиками интерфейса сайта выявлено, что выгодней завлечь пользователей дешевой ценой, чем предоставить удобный интерфейс для работы.

На приведенном выше примере темные паттерны данного сайта только начинаются. Следующие две ситуации иллюстрируют «приманку». Продолжая пролистывать страницу результатов, пользователь увидит блок, представленный на рисунке 2.

Все та же основная цена зеленым цветом и незаметные цифры ниже. Они снова не более 50? К сожалению, нет: здесь добавочная стоимость уже составляет 385. Таким образом, итоговая цена данного места $386+385=771$. Разница значительна, неприятна и способна вызвать раздражение, ведь на каком-то моменте пользователь поймет, что его обманули.

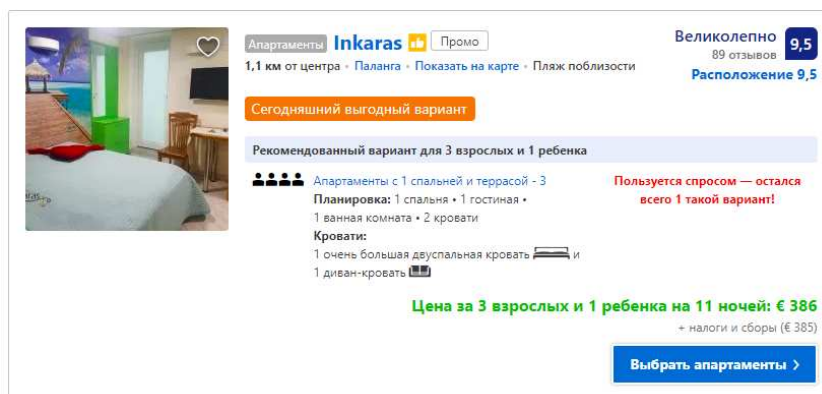


Рисунок 2–Скриншот страницы сайта booking.com

Пользователь совершил клик и перешел на страницу того или иного отеля (рисунок 3). Здесь его приветствует приятное уведомление: «Вам улыбнулась удача – обычно этот вариант распродан». Разработчики интерфейса очень ловко и аккуратно управляют пользователем: они «задобрили» его, дав почувствовать победу и удачу, и тут же призвали к быстрому принятию решения – скорейшему бронированию.

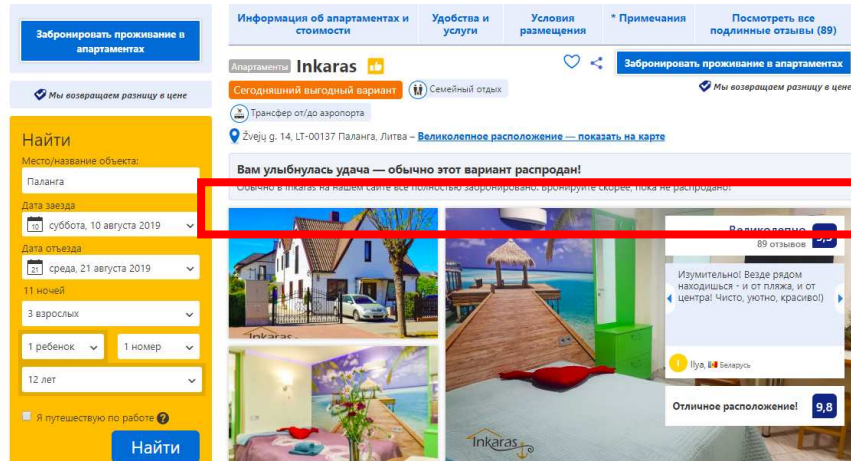


Рисунок 3—Скриншот страницы сайта booking.com

Однако нужно быть предельно осторожным! Страницы с другими вариантами также имеют подобную строку. В итоге получается, что пользователя обманули, но сделали это очень искусно.

Часто встречающимся видом темного паттерна является «замаскированная реклама». Главная страница популярного в рунете новостного портала Рамблер демонстрирует данный вид максимально наглядно. На рисунке 4 представлен скриншот страницы, блоки рекламы на нем внедрены в новостные блоки и оформлены схожим образом. Невнимательному пользователю будет тяжело понять, где реклама, а где ожидаемый контент.

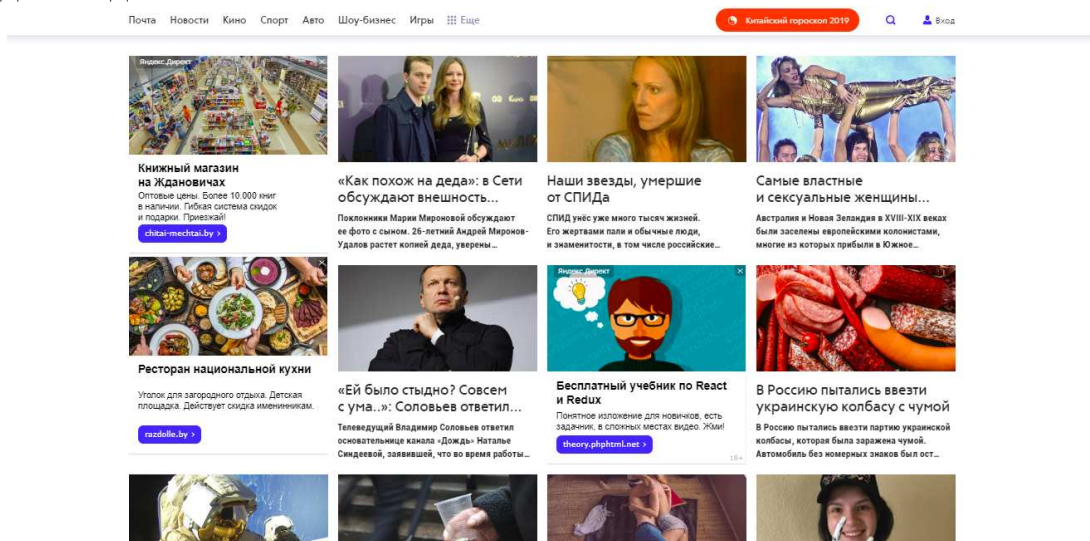


Рисунок 4—Скриншот страницы сайта rambler.ru

Другие виды темных паттернов включают: подтверждение с чувством стыда, принудительное продление, скрытая стоимость, вопросы с подвохом, сюрпризы в корзине и другие.

Тёмные паттерны, не требуя больших усилий и помогая компаниям увеличить краткосрочные продажи, подписки, КРІ и другие показатели, получить дополнительные личные данные, становятся все популярнее. А популярны они по причине того, что пользователи ведутся на них, легко позволяя обманывать себя. Чтобы защитить себя от нечестных разработчиков интерфейсов и не попасть в сети темного паттерна, необходимо быть предельно внимательным и помнить, что каждый действует в угоду своих интересов.

ЛИТЕРАТУРА

1. Акулич, М. Дизайн пользовательского интерфейса (UX) и дизайнерское мышление / М. Акулич – Издательские решения, 2018. – 80с.

УДК004.946

Студ. М.Ю. Радченко

Научн. рук. доц. О.А. Новосельская
(кафедра информатики и веб-дизайна, БГТУ)

ИСПОЛЬЗОВАНИЕ ACTIONS ПРОГРАММЫ ADOBE PHOTOSHOP ДЛЯ СОЗДАНИЯ ЭФФЕКТА АКВАРЕЛЬНОГО РИСУНКА

Пожалуй, в современном мире не встретить ни одного человека, который бы не слышал о такой программе как Adobe Photoshop. А все потому, что хотим мы этого или нет, мы живем в мире digital, где огромный процент информации мы получаем именно в цифровом формате, что, конечно, подразумевает невероятное количество изображений, фото, вивуалов, видео, рекламы и многого другого.

Adobe Photoshop обладает практически безграничными возможностями. Его с успехом используют фотохудожники для обработки снимков, программа позволяет производить ретушь, цветовую и тоновую коррекцию, размытие и повышение резкости. Возможность выделения и работы с частями изображения незаменима для оформления монтажей. Обширный набор фильтров (искажение, цветовые сдвиги, другие специальные эффекты) активно используются при создании как коммерческого дизайна, так и художественных произведений [1].

Программа используется многими дизайнерами и другими специалистами в профессиональных целях, они в отличии от любителей знают

обо всем функционале программы. Примером можно назвать actions. Actions (Операции) – это, по сути, макрос, записанная последовательность каких-то команд меню или операций, выполняемых с помощью инструментов Photoshop, или функций редактирования изображения.

Для чего будут полезны операции? Предположим, у вас есть некоторое количество фотографий, будь то воспоминания из путешествия, фоторепортаж мероприятия или результат фотосессии, которые необходимо обработать определенным образом и сохранить. Для таких случаев и существуют Actions (Операции). Действия выполняются один раз, их последовательность записывается и применяется для каждой фотографии отдельно или для всех вместе. Удобно и быстро, происходит некая автоматизация наших действий. Подойдут Операции и для художественной обработки изображений.

Сегодня Adobe Photoshop перестал быть навороченным инструментом, доступным исключительно немногочисленной когорте профессионалов, он стал отличным помощником для любого современного человека. И, если обрезать фотографию и увеличить контрастность может почти каждый, то, к примеру, сделать эффект рисунка карандашом, эскиза акварелью или картины масляными красками, такой пользователь сделать не сможет. Здесь ему также могут помочь Операции. Дело в том, что операции обладают уникальным свойством – их можно сохранить и свободно распространять. Операцию можно создать и сохранить как файл с расширением *.atn, после чего любой пользователь, у которого будет данный файл сможет применить тот или иной эффект для своего изображения.

Целью работы является создание Операции, обеспечивающей эффект акварельной картины и последующее сохранение ее в формате *.atn для распространения.

Для записи операции вызывается окно Операции. В момент создания новой операции указывается ее имя и горячие клавиши, с помощью которых можно будет быстро вызвать операцию.

Открывается изображения, используются следующие Фильтры программы Adobe Photoshop: Аппликация, Масляная живопись, Губка.

В процессе работы при использовании фильтров происходит потеря цветовых составляющие изображения. Для их восстановления создается копия фонового слоя (Ctrl+J), он перемещается на самый верх списка слоев. Устанавливается режим наложения на «Цветность». Исходное изображение и промежуточный результат работы представлены на рисунках 1 и 2 соответственно.



Рисунок 1 – Исходное изображение



Рисунок 2 – Промежуточный результат

Для достижения эффекта реального рисунка прodelываются следующие действия:

- размытие изображения: Фильтр – Размытие – «Умное» размытие;
- усиление резкости: Фильтр – Усиление резкости – «Умная» резкости;
- добавление Шума: Фильтр – Шум – Добавить шум;
- применение фильтра Волна: Фильтр – Искажение – Волна.

Следующим этапом является создание эффекта обрисовки контура. Создается копия фонового слоя (Ctrl+J), он перемещается на самый верх списка слоев. Применяется фильтр Свечение краев: Фильтр – Галерея фильтров – Стилизация – Свечение краев. Производится инвертирование цвета на слое (Ctrl+I), его обесцвечивание (Shift+Ctrl+U). Комбинацией клавиш Ctrl+L открывается окно Уровни и поднимается контрастность слоя. Применяется фильтр Аппликация. Устанавливается режим наложения Умножение. Этапы создания данного эффекта представлены на рисунках 3 и 4.



Рисунок 3 – Этап создания контура



Рисунок 4 – Этап придания контура

Акварель – водная техника рисования, а значит необходимо создать «водные подтеки» на изображении. Они создаются следующим образом. Создается новый слой, заливается белым цветом. К слою применяется эффект Шум (Фильтр – Шум – Добавить шум). Параметры: эффект – 400%, распределение – по Гауссу, монохромный. Применяется фильтр Губка, осуществляется размытие слоя. Фильтр – Размытие – Размытие по Гауссу, радиус – 9 пикс. Повыша-

ется контрастность слоя. Создается дубликат данного слоя. Происходит масштабирование нового слоя (увеличение в 2 раза). Для каждого из двух слоев устанавливается режим наложения Перекрытие. На данном этапе появляется текстура акварельной бумаги.

Добавляются корректирующие слои Яркость / Контрастность и Цветовой тон / Насыщенность. Достигается желаемый результат путем перемещения необходимых ползунков. Результат совершенных действий представлен на рисунке 5.

Нажимается кнопка Stop, чтобы закончить запись. Операция перемещается в новый набор и сохраняется в формате *.atn. Теперь этот файл можно распространять и загружать в программу Adobe Photoshop на других компьютерах. Окно Операции показано на рисунке 6.



Рисунок 5 – Результат

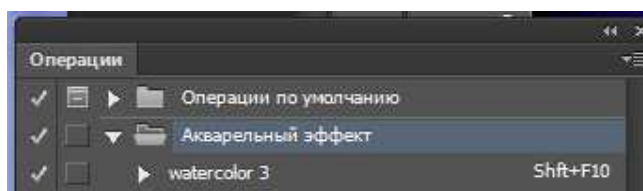


Рисунок 6 – Окно Операции

Созданная операция была применена для выполнения курсовой работы при создании рекламного баннера выставки акварельных работ. Полученный результат представлен на рисунке 7.



Рисунок 7 – Работа с использованием созданной операции

Таким образом, Операции являются полезным функционалом, который может быть полезен как людям, работающим профессионально в программе Adobe Photoshop для более быстрой и удобной работы, так и любителям, которые просто хотят создать интересные эффекты своих изображений.

ЛИТЕРАТУРА

1. Комолова, Н. В. Adobe Photoshop CS4 для всех/ Н. В. Комолова, Е. С. Яковлева – СПб: БХВ-Петербург, 2009. – 992 с.
2. Эффект акварели в фотопше [Электронный ресурс] / Lumpics – 2016 – Режим доступа: <http://lumpics.ru/watercolor-effect-in-photoshop/> – Дата доступа: 02.04.2019.

УДК 003.26 +347.78

Магистрант Р.И. Белькевич
Научн. рук. зав. кафедрой Д.М. Романенко
(кафедра информатики и веб-дизайна, БГТУ)

МЕТОД ОСАЖДЕНИЯ АВТОРСКОЙ ИНФОРМАЦИИ В ЗВУКОВЫЕ ФАЙЛЫ

В настоящее время одним из главных источников получения и распространения информации является глобальная сеть Интернет. В связи с этим практически все, кто занимается производством медиа продуктов перешли на цифровые форматы. Именно главенство цифрового формата подняло проблемы защиты информации и авторских прав. И оказалось, что эта проблема достаточно серьезна для вмешательства государств разных стран и регулировки на уровне законодательства.

Чтобы скопировать файл, практически не требуется ни высокой квалификации, ни времени, ни затрат. Вместе с глубоким проникновением компьютеров в образ жизни современного человека это формирует и соответствующие привычки поведения в цифровой среде, где копирование, передача и распространение файлов и информации является столь же естественным, как совместное прослушивание музыки, просмотр кино и обмен книгами в отношении объектов авторского права в физическом мире. А это, в свою очередь, порождает неавторизованное копирование и распространение, то есть создание и распространение копий произведений без соответствующих национальному и международному законодательству разрешений.

Формат WAV/WAVE или WAFF (WindowsAudioFileFormat) – стандарт, который используется для хранения аудио потока на ПК. Характеризуется высоким качеством и большим размером аудиофай-

лов [3]. Несмотря на то, что WAV файл может содержать сжатое аудио, наиболее обычным его применением является хранение как раз несжатого аудио в формате линейной импульсно-кодовой модуляции (LPCM). Стандартным форматом Audio-CD, например, является аудио в формате LPCM, с 2 каналами, частотой дискретизации 44 100 Гц. WAV аудио файл также может быть изменен практически в любом аудио редакторе. WAV файлы достаточно большие, что делает этот формат неудобным для обмена по сети Интернет, и это сильно подрывает его популярность. Однако, этот формат – как правило, чаще всего используется для сохранения первоначального вида для файлов высокого качества в таких случаях, где размер свободного дискового пространства не является ограничением.

В конечном итоге формат WAV – это сугубо компьютерный звуковой формат, обладающий высоким качеством, но и большим объемом. Может храниться на любом приемлемом для ПК виде.

Стандарт MIDI (от Musical Instruments Digital Interface – цифровой интерфейс музыкальных инструментов) представляет собой «язык», на котором «общаются» MIDI-совместимые устройства. Иными словами, этот стандарт описывает определённые сообщения (двоичные команды), которыми могут обмениваться различные MIDI-устройства [4].

Ключевой особенностью интерфейса MIDI является то, что с его помощью устройства обмениваются именно логическими командами, и никакого отношения к передаче звука как такового MIDI не имеет. Очевидно, любая последовательность MIDI-сообщений может быть сохранена в виде обычного файла. В отличие от формата Wave, формат MIDI предназначен для хранения данных совершенно иного типа и не предназначен для хранения звука, являясь по сути лишь способом представления нотных партитур в цифровом (двоичном) виде.

Особенностью этого формата является необходимость в вычислительном устройстве с весьма высокой производительностью, поскольку синтез качественных звуков происходит по довольно сложным формулам. Файлы формата MIDI по сути являются набором команд для "защитых" в звуковую плату ПК синтезированных звуков. При воспроизведении данного формата можно изменять тональность, темп, подменять инструменты. MIDI имеет очень маленький объём и может воспроизводиться только при помощи специальных компьютерных программ. Формат MIDI не приемлем для записи и хранения голоса.

MP3 (англ. – MPEG-1/2/2.5 Layer 3) – третий формат кодирования звуковой дорожки MPEG – лицензируемый формат файла для хранения аудио-информации. На данный момент MP3 является самым известным и популярным из распространённых форматов цифрового кодирования

звуковой информации с потерями. Формат может проигрываться практически в любой популярной операционной системе, на практически любом устройстве [9]. В формате MP3 используется алгоритм сжатия с потерями, разработанный для существенного уменьшения размера данных, необходимых для воспроизведения записи и обеспечения качества воспроизведения очень близкого к оригинальному. В этом формате звуки кодируются частотным образом (без дискретных партий). Степень сжатия можно варьировать, в том числе в пределах одного файла. Структура MP3-файла представлена на рисунке 1.

Интервал возможных значений битрейта составляет 8–320 кбит/с. MP3 файлы могут создаваться с высоким или низким битрейтом, который влияет на качество файла-результата.

ID3 – формат метаданных, наиболее часто используемый в звуковых файлах в формате MP3. В ID3 содержатся данные о названии трека, альбома, имени исполнителя и т.д. [8]. Существует две несовместимых версии ID3: ID3v1 и ID3v2. ID3v1 – имеет фиксированный размер в 128 байт, которые дописываются в конец mp3-файла. Формат стандарта ID3v1 представлен на рисунке 2.

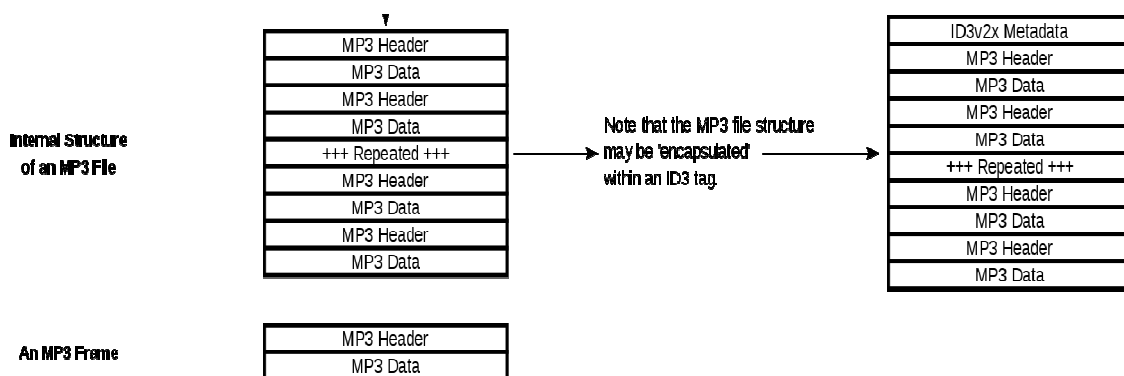


Рисунок 1 – Структура MP3-файла

Поле	Длина, байт	Описание
Header	3	Маркер. Всегда содержит 'TAG'
Title	30	Название трека
Artist	30	Исполнитель
Album	30	Название альбома
Year	4	Год
Comment	28 (ID3v1.1) или 30 (ID3v1.0)	Комментарий
Zero-byte	1 (ID3v1.1)	Если есть номер трека, тут установлен нулевой байт
Track	1 (ID3v1.1)	Номер трека
Genre	1	Индекс жанра, или 255 (0xFF)

Рисунок 2 – Формат стандарта ID3v1

Очевидно, что 128 байт недостаточно для хранения таких данных. И поэтому появилась и успешно используется вторая версия данных – ID3v2. В отличие от первой версии, теги v2 имеют перемен-

ную длину и размещаются в начале файла, что позволяет поддерживать потоковое воспроизведение. Данные ID3v2 состоят из заголовка и последующих фреймов ID3v2, что показано на рисунке 3.



Рисунок 3 – Формат стандарта ID3v2

Тэги могут быть удалены любым из редакторов, вся информация легкодоступна и отображается с помощью простейших программ.

Весь mp3-файл состоит из фреймов, которые можно извлекать только последовательно. Фрейм содержит в себе заголовок и аудиоданные. Самым интересным местом является именно заголовок фрейма. Его структура представлена на рисунке 4.

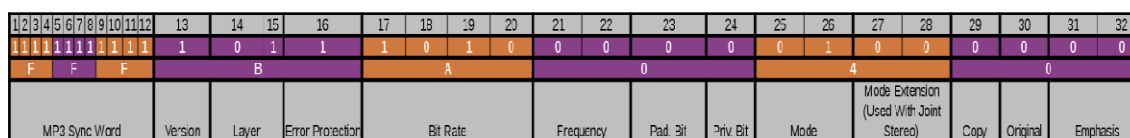


Рисунок 4 – Структура заголовка MP3 – фрейма

Потенциальными местами осаждения информации здесь являются неиспользуемые биты 31 и 32, а так же биты 17-20, используемые для кодирования битрейта файла.

Как итог, можно предложить биты 17-20 для осаждения информации путём смены кодов битрейтов для последовательности фреймов, задаваемых уникальным ключом автора. Биты 31 и 32 используются в качестве контрольной суммы и средства восстановления потерянных данных. Так же предлагается вносить изменения в кодировки битрейта, не несущие в себе осаждаемую информацию. Это необходимо для усложнения вычисления места осаждения авторской информации. В листинге ниже представлен код осаждения 2 бит авторской информации в биты кодировки битрейта 19 и 20 битового массива, полученного из тестового mp3-файла.

```
var b3 = view.getUint8(offset + 2);
b3 = octetToBinRep(b3);
let bitrateBits = "10";
let lastBits = b3.substr(4, 4);
let depositedData = bitrateBits.concat(data, lastBits);
depositedData = parseInt(depositedData, 2);
view.setUint8(offset+2, depositedData);
```

На рисунках 5 и 6 отображено содержимое заголовка фрейма MP3-файла до осаждения информации и после соответственно.

```
header:
  { _section: { type: 'frameHeader', byteLength: 4, offset: 26396 },
    mpegAudioVersionBits: '11',
    mpegAudioVersion: 'MPEG Version 1 (ISO/IEC 11172-3)',
    layerDescriptionBits: '01',
    layerDescription: 'Layer III',
    isProtected: 1,
    protectionBit: '1',
    bitrateBits: '11101'
```

Рисунок 5 – Содержимое заголовка фрейма до осаждения

```
header:
  { _section: { type: 'frameHeader', byteLength: 4, offset: 26396 },
    mpegAudioVersionBits: '11',
    mpegAudioVersion: 'MPEG Version 1 (ISO/IEC 11172-3)',
    layerDescriptionBits: '01',
    layerDescription: 'Layer III',
    isProtected: 1,
    protectionBit: '1',
    bitrateBits: '1011',
    bitrate: 192,
```

Рисунок 6 – Содержимое заголовка фрейма после осаждения

В ходе реализации представленного алгоритма было выявлено, что использовать все 4 бита кодировки битрейта невозможно из-за существующих запрещенных комбинаций «1111» и «0000». Для решения этой проблемы было решено использовать только биты 19 и 20, при этом биты 17 и 18 устанавливаются в значениях «1» и «0» соответственно. Это обусловлено тем, что в таком случае битрейт будет колебаться в диапазоне от 112 до 192 килобит/сек, что малозаметно для слуха человека.

ЛИТЕРАТУРА

1. Хофман, Л. Современные методы защиты информации / Л. Хофман; под ред. В. А. Герасименко. – М.: Сов. радио, 1980. – 264 с.
2. Урбанович, П.П. Информационная безопасность и надежность систем: учеб.-метод. пособие для студентов специальности 1–40 01 02 «Информационные системы и технологии» / П. П. Урбанович, Д. М. Романенко, Е. В. Романцевич. – Минск: БГТУ, 2007. – 100 с.
3. Формат WAV [Электронный ресурс] / audiocoding.ru – Режим доступа: <https://audiocoding.ru/formats/wav/> – Дата доступа: 12.04.2019.
4. MIDI ID3 [Электронный ресурс] / wikipedia.org – Режим доступа: <https://ru.wikipedia.org/wiki/MIDI>– Дата доступа: 12.04.2019.
5. ID3 [Электронный ресурс] / wikipedia.org – Режим доступа: <https://ru.wikipedia.org/wiki/ID3> – Дата доступа: 12.04.2019.

6. How is mpr3built [Электронный ресурс] / id3.org – Режим доступа: <http://id3.org/mp3Frame> – Дата доступа: 12.04.2019.

7. Внутри MP3. А как оно всё устроено? [Электронный ресурс] / habr.com – Режим доступа: <https://habr.com/ru/post/103635/> – Дата доступа: 14.04.2019.

8. Формат представления звуковых данных MP3 [Электронный ресурс] / kunegin.com – Режим доступа: <http://kunegin.com/ref/mp3/> / – Дата доступа: 15.04.2019.

УДК 528.854

Магистрант А. В. Олеферович
Науч. рук. зав. кафедрой Д. М. Романенко
(кафедра информатики и веб-дизайна, БГТУ)

ВЛИЯНИЕ ДЕФЕКТОВ НА ПРОЦЕСС РАСПОЗНАВАНИЯ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ МЕТОДОМ КОНТУРНОГО АНАЛИЗА

Ранее, в качестве метода распознавания образов был выбран метод контурного анализа. Однако в ходе изучения было определены его недостатки, а именно влияние дефектов изображения на процесс распознавания. Был сделан вывод о том, что методы контурного анализа требуют проведения предварительной обработки изображения.

Далее необходимо понять, что именно подразумевается под определением «предварительная обработка изображения». Под предварительной обработкой понимается формирование и последующее улучшение изображения, его бинаризация и кодирование (в частности, получение контурного представления).

Изображения, вводимые в компьютер, часто являются малоконтрастными. Слабый контраст, как правило, обусловлен широким диапазоном воспроизводимых яркостей, нередко сочетающийся с нелинейностью характеристики передачи уровней. Характер зависимости изменения яркости палитры пикселей от минимального значения до максимального также влияет на качество изображения. Оптимальной является линейная функция изменения интенсивности пикселей. При вогнутой характеристике изображение будет более темным, при выпуклой – более светлым. И в том, и в другом случае признаки объектов могут быть искажены и недостаточно хорошо идентифицируемы. Коррекция (линеаризация) яркости палитры существенно улучшает качество изображения. Малая контрастность может быть обусловлена и тем, что вариации функции яркости пикселей на изображении намного меньше допустимого диапазона шкалы яркостей. В этом случае

контрастность изображения повышается путем "растягивания" реального динамического диапазона яркостей на всю шкалу при помощи линейного поэлементного преобразования.

Существует два возможных способа коррекции яркости. Согласно первому способу изображение линейно отображается в диапазоне яркостей исходного. Второй способ предусматривает ограничение яркости пикселей в обработанном изображении максимальным и минимальным пороговыми уровнями, и имеет более широкое применение. Присутствие в изображении самых светлых и самых темных тонов создает впечатление хорошей контрастности, однако излишняя контрастность приводит к тому, что максимальные градации влияют на средние тона, а большинство деталей изображения окрашены именно в средних тонах и излишняя контрастность может приводить к потере этих деталей или затруднить их выделение.

Выбор порога яркости является важнейшей процедурой первого этапа предварительной обработки изображения, поскольку его качество сильно зависит от освещенности рабочей сцены. Если освещенность занижена, то увеличивается количество помех на изображении вплоть до потери объекта; при очень сильной освещенности происходит засвечивание объекта. В большинстве случаев порог яркости регулируют при вводе изображения через фреймграббер[1].

Гистограммы яркости. Инструментом для оценки уровней интенсивности пикселей является гистограмма – графическое отображение количественной характеристики вероятностного распределения интенсивности (яркости) пикселей в выделенном участке изображения. Максимальному значению интенсивности пикселей присваивается уровень градации интенсивности 255 (белый цвет), самому темному – значение 0 (черный цвет). Интенсивности в диапазоне от 0 до 255 имеют линейную шкалу изменения, либо устанавливаемую в соответствии с принятой функцией изменения. Например, усиливающей слабые сигналы (градации серого) и ослабляющей сильные сигналы (в области белого цвета), чем повышается пространственное и контрастное разрешение изображения или определенной зоны интереса.

Известен метод улучшения изображений, основанный на вычислении логарифма спектральных коэффициентов преобразования Фурье исходного изображения (вычисление кепстра). При обратном преобразовании кепстра в изображение происходит выравнивание гистограммы изображения за счет логарифмического преобразования спектра изображения.

Многие изображения характеризуются гистограммами с высокой концентрацией линий в определенных зонах распределения интенсивно-

сти [2]. Часто гистограмма распределения яркостей изображения имеет перекосяк в сторону малых уровней (яркость большинства элементов ниже средней). Одним из методов улучшения качества таких изображений является видоизменение их гистограммы. Выравнивание гистограммы может быть осуществлено на основе возведения в степень модуля спектральных коэффициентов Фурье-преобразования изображения, при этом знак и фаза коэффициентов сохраняется. Если обозначить показатель степени α , то при $\alpha < 1$ операция извлечения корня степени α уменьшает большие спектральные коэффициенты и увеличивает малые. Такое перераспределение энергии в частотной плоскости изображения приводит к более эффективному использованию динамического диапазона интенсивностей пикселей изображения в пространственной области.

Выбор хорошей маски регулирования гистограммы интенсивности пикселей повышает контраст, тем самым улучшая контрастную разрешающую способность деталей. Существует достаточно большой арсенал математических моделей и алгоритмов, программная реализация которых позволяет значительно повысить контрастное разрешение изображений. Эти алгоритмы основаны на процессах линейной и нелинейной фильтрации изображений, преобразующей гистограмму интенсивности, что позволяет сделать вывод о том, что контуры объектов будут четко определены, следовательно, метод контурного анализа сработает.

Шумопонижение – процесс устранения шумов из полезного сигнала с целью повышения его субъективного качества или для уменьшения уровня ошибок в каналах передачи и системах хранения цифровых данных [3]. Методы шумоподавления концептуально очень похожи независимо от обрабатываемого сигнала, однако предварительное знание характеристик передаваемого сигнала может значительно повлиять на реализацию этих методов в зависимости от типа сигнала.

Шумоподавление изображений чаще всего служит для улучшения визуального восприятия, однако возможно применение в медицине в целях увеличения четкости изображения на рентгеновских снимках, в качестве предобработки для последующего распознавания и в других случаях.

Любая из процедур обработки и анализа изображений содержит в своей структуре этап предварительной обработки, включающий сглаживание, фильтрацию шумов, повышение четкости и контрастности. Кроме того, предварительная обработка изображений включает в себя коррекцию нелинейности датчика, яркости, контраста, устранение геометрических искажений, подчеркивание интересных объектов относительно фона. Часто, на данном этапе осуществляется коррекция возмущений в изображении, обусловленных расфокусировкой оптики, размы-

тостью изображения в результате движения объекта, погрешностями в датчике, либо при передаче сигналов изображения.

Эффективность этапа предварительной (первичной) обработки изображений оказывает решающее влияние на результат распознавание образов. Более того, в ходе обработки изображения, возможно неоднократное обращение к данному этапу для достижения необходимого результата.

ЛИТЕРАТУРА

1. Определение порога яркости [Электронный ресурс]. – 2015-2018. – Режим доступа: <http://www.cyberforum.ru/csharp-net/thread355817.html>. – Дата доступа: 07.04.2019.

Задание гистограммы [Электронный ресурс]. – 2015-2018. – Режим доступа: <https://photoshop-master.ru/lessons/practice/gistogramma.html>. – Дата доступа: 08.04.2019.

2. Шумопонижение [Электронный ресурс]. – 2015-2018. – Режим доступа: <https://ru.wikipedia.org/wiki/Шумопонижение>. – Дата доступа: 08.04.2019.

УДК 004.9

Студ. Е.А. Буйко

Науч. рук. ассист. А.В. Олеферович
(кафедра информатики и веб-дизайна, БГТУ)

РЕНДЕРИНГ ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ

Рендеринг – термин в компьютерной графике, обозначающий процесс получения изображения по модели с помощью компьютерной программы. Часто в компьютерной графике под 3D-рендерингом понимают создание плоской картинке цифрового растрового изображения по разработанной 3D-сцене[1].

В зависимости от цели, различают пре-рендеринг, как достаточно медленный процесс визуализации, применяющийся в основном при создании видео, например в Vegas Pro, и рендеринг в режиме реального времени, например, в компьютерных играх. Последний часто использует 3D-ускорители.

Компьютерная программа, производящая рендеринг, называется *рендером* или *рендерером*.

На текущий момент разработано множество алгоритмов визуализации. Существующее программное обеспечение может использовать несколько алгоритмов для получения конечного изображения. Трассирование каждого луча света в сцене непрактично и занимает неприемлемо долгое время. Даже трассирование малого количества

лучей, достаточного, чтобы получить изображение, занимает чрезмерно много времени, если не применяется аппроксимация (сведение объектов к другим более простым).

Вследствие этого, было разработано четыре группы методов, более эффективных, чем моделирование всех лучей света, освещающих сцену [2]:

– растеризация совместно с методом сканирования строк (Scanline). Визуализация производится проецированием объектов сцены на экран без рассмотрения эффекта перспективы относительно наблюдателя;

– raycasting (*рейкастинг*). Сцена рассматривается, как наблюдаемая из определённой точки. Из точки наблюдения на объекты сцены направляются лучи, с помощью которых определяется цвет пикселя на двумерном экране. При этом лучи прекращают своё распространение, когда достигают любого объекта сцены либо её фона. Возможно использование каких-либо очень простых способов добавления оптических эффектов. Эффект перспективы получается естественным образом в случае, когда бросаемые лучи запускаются под углом, зависящим от положения пикселя на экране и максимального угла обзора камеры;

– трассировка лучей похожа на рейкастинг. Но при этом луч не прекращает своё распространение, а разделяется на три луча-компонента, каждый из которых вносит свой вклад в цвет пикселя на двумерном экране: отражённый, теневой и преломлённый. Количество таких компонентов определяет глубину трассировки и влияет на качество и фотореалистичность изображения. Благодаря своим концептуальным особенностям, метод позволяет получить очень фотореалистичные изображения, однако из-за большой ресурсоёмкости процесс визуализации занимает значительное время;

– трассировка пути использует похожий принцип трассировки распространения лучей, однако этот метод является наиболее приближённым к физическим законам распространения света. Также является самым ресурсоёмким.

Передовое программное обеспечение обычно совмещает в себе несколько техник, чтобы получить достаточно качественное и фотореалистичное изображение за приемлемые затраты вычислительных ресурсов. Рассмотрим из них: Autodesk 3ds Max.

Список модулей рендеринга: Scanline, Art Render, Arnold Renderer, mental ray, V-Ray, RenderMan, FinalRender, Brazil R/S, Fryrender, Indigo Renderer, Maxwell Render, LuxRender, Kerkythea, Verge3D, Arion Render, BIGrender, Sunflow, Krakatoa, Corona Renderer, Hydra Renderer [3].

– *Scanline*. Визуализатор по умолчанию в 3ds Max. Исходным методом рендеринга в *3DS Max* является сканирующий построчный алгоритм.

– *V-Ray*. Высококачественный фотореалистичный визуализатор, спроектированный в качестве плагина для 3ds Max. Популярнейший в русскоязычном пространстве внешний визуализатор компании Chaos Group. *V-Ray* – это рейтрейсный рендерер, в котором присутствует несколько алгоритмов расчёта глобального освещения. Имеется возможность выбора различных алгоритмов для расчёта отражения и переотражения глобального освещения.

– *FinalRender*. Внешний визуализатор компании Sebas. Является наиболее полным фотон-основанным визуализатором (трассировка пути), уступая по своим возможностям только *mental ray*.

– *Fryrender*. Фотореалистичный, основанный на законах физики, спектральный визуализатор. Создан компанией RandomControl. Предоставляет возможность получать изображения высочайшего качества и достигать естественного реализма.

– *IndigoRenderer*. Физически корректный рендер. Основная особенность его в том, что все расчеты света, энергии, каустики и т. д. происходят взаимозависимо, что и отличает его от других рендеров.

– *MaxwellRender*. Является первой системой визуализации, в которой принята «физическая парадигма». В основу всей системы положены математические уравнения, описывающие поведение света. Вводя в обращение реальные физические законы.

– *Verge3D*. 3D рендерер реального времени и сопутствующий инструментарий, предназначенный для создания и отображения интерактивной трёхмерной графики в браузерах.

– *BIGrender*. Рендер, способный визуализировать большие изображения. В процессе визуализации разбивает изображение на части и каждую часть сохраняет в отдельном файле.

Рассмотрим объект в 3D моделировании [4]:

– *Определение сторон*. Поверхность определяется в списке вершин, текстурных координат и нормалей. Полигоны, такие как квадрат, могут быть определены с помощью более 3 вершин/текстурных координат/нормалей.

– *Вершины*. Строка, начинающаяся с f , представляет собой индекс Поверхности. Каждая поверхность (полигон) может состоять из трех или более вершин. Индексация начинается с первого элемента, а не с нулевого, как принято в некоторых языках программирования, также индексация может быть отрицательной. Отрицательный индекс указывает позицию относительно последнего элемента (индекс -1 указывает на последний элемент).

– *Вершины / Текстуры координаты*. Наряду с вершинами могут сохраняться соответствующие индексы текстурных координат.

– *Вершины / Текстуры координаты / Нормали*. Также допустимо сохранение соответствующих индексов нормалей.

– *Вершины / Нормали*. При отсутствии данных о текстурных координатах допустима запись с пропуском индексов текстур.

Далее было рассмотрено, какие алгоритмы понадобятся для создания движка. В качестве графического API был использован SFML. Так как файл для хранения созданных объектов и их изменения в профессиональных инструментах 3D моделирования предпочитают использовать формат “.obj”, то было решено выполнять рендеринг текущих объектов из файла расширения “.obj”. Он позволяет упростить работу с финальным выводом.

Основные темы, которые потребовалось затронуть для создания данной программы:

1. Линейные преобразования (перемещение, повороты, масштабирование объекта и камеры).

2. Метод Брезенхэма (для размытия полигонов и получение реалистичного вида а так же для исправления неполадок в пересчёте координат).

3. Полигоны и стандартные методы манипуляций над ними (отображение видимых, углубление, z-buffer).

4. Освещение.

5. Структура объектного файла (хранение и сохранение).

6. Изучения SFML. В качестве теоретической основы большинства представленных тем для создания движка была использована литература по КГ и графике.

ЛИТЕРАТУРА

1. Дятко, А. А. Визуализация изображений. – Методическое пособие БГТУ по дисциплине «Компьютерная геометрия и графика».С. 6–8.

2. Википедия – свободная энциклопедия [Электронный ресурс] / Wikimedia Foundation, Inc. – 2019. – Режим доступа: <https://ru.wikipedia.org/wiki/Obj>– Дата доступа: 06.04.2019.

3. Википедия – свободная энциклопедия [Электронный ресурс] / Wikimedia Foundation, Inc. – 2019. – Режим доступа: https://ru.wikipedia.org/wiki/Autodesk_3ds_Max– Дата доступа: 08.04.2019.

4. Хабрахабр [Электронный ресурс]. – 2019. – Режим доступа: <https://habr.com/ru/post/334580/> – Дата доступа: 10.04.2019.

УДК 004.9

Студ. Ю. А. Карленок
Науч. рук. ассист. А. В. Олеферович
(кафедра информатики и веб-дизайна, БГТУ)

ДИСКРЕТНОЕ КОСИНУСНОЕ ПРЕОБРАЗОВАНИЕ

Одним из наиболее частых преобразований двумерных изображений является дискретное косинусное преобразование (Discrete Cosine Transformation, DCT). Оно лежит в основе почти всех стандартов сжатия, которые применяются в видеонаблюдении, за исключением Wavelet и JPEG-2000. Таким образом, все стандарты JPEG, MPEG и семейство H.26x используют DCT-преобразование в той или иной форме [1].

Как работает дискретное косинусное преобразование? Во всех изображениях неизменно присутствует некоторая избыточность. Если в пределах одного поля находится объект достаточного размера, все пиксели, представляющие этот объект, имеют примерно одни и те же значения яркости и цвета.

Это и есть избыточность, позволяющая уменьшить количество информации о каждом отдельном пикселе, определённым образом описав контуры объекта и указав средние значения яркости и цвета в пределах этого контура. Крупные объекты соответствуют низким пространственным частотам, а мелкие – высоким. На верхнем уровне эти частоты одновременно не присутствуют. В цифровом видеосигнале может быть передан весь спектр пространственных частот, однако если провести частотный анализ изображения, то возможно оставить в сигнале лишь те частоты, что действительно в нём присутствуют. Следовательно, важным шагом в процессе сжатия изображений является анализ пространственных частот.

Основной этап процедуры сжатия цифровых изображений заключается в преобразовании небольших блоков изображения при помощи двумерного ДКП. Обработка ведется блоками 8x8 пикселей.

В результате выполнения ДКП формируется 64 коэффициента. В результате исходный фрагмент представлен в области пространственных частот. Этот шаг еще не приводит к сжатию изображения. Однако при его выполнении полагается, что в подавляющем большинстве изображений близкие по своим координатам пиксели имеют и близкие значения. Поэтому, при переходе от фрагмента к его частотному представлению большая часть энергии сигнала сосредотачивается в области низких частот, т.е. компоненты с меньшим значением индекса имеют большие значения.

При выполнении этой операции 64 исходных пикселей преобразуются в матрицу из 64 коэффициентов, которые характеризуют "энер-

гию" исходных пикселей. Важнейшей особенностью этой матрицы коэффициентов является то, что первый коэффициент передает подавляющую часть "энергии", а количество "энергии", передаваемой остальными коэффициентами, очень быстро убывает. То есть, большая часть информации исходной матрицы 8x8 пикселей представляется первым элементом матрицы, преобразованной по способу ДКП.

На этом этапе происходит некоторая потеря информации, связанная с принципиальной невозможностью точного обратного преобразования (на этапе восстановления изображения). Однако эта потеря информации весьма незначительна по сравнению с потерями на следующем этапе. Затем преобразованная матрица из 64 пикселей проходит операцию квантования, которая применяется для сокращения разрядности коэффициентов. В результате квантования происходит обнуление многих коэффициентов. Именно здесь происходит самая значительная потеря информации – отбрасываются малые изменения коэффициентов. Поэтому в процессе восстановления изображения после операции обратного ДКП получаются уже другие параметры пикселей. Квантование также обеспечивает возможность последующего эффективного сжатия данных при помощи любого способа сжатия без потерь. После квантования компоненты спектра всех обработанных фрагментов «вытягиваются» в последовательность чисел с помощью алгоритма диагонального сканирования. В основе такого сканирования лежит прием, позволяющий достичь большего уплотнения и основывающийся также на характерном виде спектра изображений реальных сцен. Статистически доказано, что для реальных многоуровневых изображений двумерный квантованный спектр представляет собой матрицу треугольного вида. Большинство значений снизу и справа – нули. Построение элементов матрицы в цепочку производится таким образом, чтобы в последовательность включались только элементы от первого до последнего ненулевого. После него в последовательность включается специальный стоп – код. Это позволяет исключить из последовательности встречающиеся нули.

Далее обычно применяется метод однопроходного кодирования Хаффмана. Сначала анализируется вся последовательность символов. Часто повторяющимся сериям бит присваиваются короткие обозначения (маркеры). Различие размеров маркеров и представляемых ими битовых серий определяет достигаемую степень сжатия.

ЛИТЕРАТУРА

1. Practical Fast 1-D DCT Algorithms with 11 Multiplications // C. Loeffler, A. Ligtenberg and G. Moschytz // Proc. Int'l. Conf. on Acoustics, Speech, and Signal Processing 1989 (ICASSP '89), стр. 988–991.

Студ. М.Г. Савельева
Науч. рук. ст. преп. Т.П. Брусенцова
(кафедра информатики и веб-дизайна, БГТУ)

ИГРОВОЙ ПОРТАЛ «МИНЬОН В ЛАБИРИНТЕ»

Игровой портал состоит из игры и поддерживающего ее сайта.

Игра является трёхмерным компьютерным симулятором. Компьютерный симулятор – компьютерный имитатор, задача которого состоит в имитации управления каким-либо процессом. Чаще всего сейчас слово «симулятор» используется применительно к компьютерным программам, обычно играм. Симуляторы – вид компьютерных игр, имитирующих управление каким-либо процессом, аппаратом или средством передвижения. Для симулятора очень важна реалистичность и полнота моделируемого объекта [1].

Симулятор разрабатывался в среде Unity3D с использованием языка C# на основе технологических требований.

Unity – межплатформенная среда разработки компьютерных игр. Unity позволяет создавать приложения, работающие под более чем 20 различными операционными системами, включающими персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие. Основными преимуществами Unity являются наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов. К недостаткам относятся появление сложностей при работе с многокомпонентными схемами и затруднения при подключении внешних библиотек [2].

Целью игры является нахождение банана. Банан заперт в сундуке, который находится на отдельной локации. Сундук можно открыть только при наличии трёх ключей, которые нужно найти в лабиринте. Все ключи расположены в разных местах лабиринта. Найденные ключи будут находиться в инвентаре.

Инвентарь можно просмотреть при нажатии клавиши «Q» на клавиатуре. С помощью клавиши «P» есть возможность просмотреть или установить настройки. Клавиша «E» открывает справку.

Переход на локацию с сундуком осуществляется с помощью телепорта, который находится на выходе из лабиринта.

После нахождения банана, нужно принести его локацию с лабиринтом и вернуться в начало лабиринта. Для того чтобы выиграть нужно встать на мраморный постамент вместе с бананом.

На прохождение игры отведено 5 минут. Управление персонажем осуществляется с помощью клавиш «W», «S», «A», «D». При прохождении игры персонаж встречается с ботами. Бот – программа-

робот, управляемая компьютером, имитирующая партнёров в сетевой игре, в сетевых поединках, командных сражениях и т. д. [3]. Присутствует несколько типов ботов. При встрече с персонажем первый тип ботов смотрит на него и говорит: «О, привет», а второй тип показывает пальцем и смеётся.

К особенностям игры можно отнести:

- наличие двух типов случайно генерируемых ботов;
- возможность выбора персонажа из 4 представленных;
- возможность создания индивидуальных настроек с их сохранением;
- наличие автоматически обновляемого инвентаря;
- случайная генерация ключей;
- система оценки прохождения уровня со звёздами;
- наличие миникарты с указанием местонахождения персонажа;
- наличие нескольких уровней и возможность разработки дальнейших;
- уникальность каждой локации.

Так как игре соответствует веб-сайт, то для него была разработана диаграмма использования. Диаграмма использования – диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Прецедент – возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе. Основное назначение диаграммы – описание функциональности и поведения.

Для отражения модели прецедентов на диаграмме используются такие элементы, как: рамки системы; актёр; прецедент. Между прецедентами может появиться либо отношение расширения (extend), либо включения (include), которые отображаются на диаграммах [4].

Для игрового портала была разработана диаграмма использования, представленная на рисунке 1.

Так же для веб-сайта и игры была разработана информационная структура, которая представлена на рисунке 2.

Информационная структура сайта – в общем виде это методология организации контентного наполнения веб-сайта, включающая все

материалы и связи, позволяющие наполнению, не перебивая взаимодействия друг с другом.

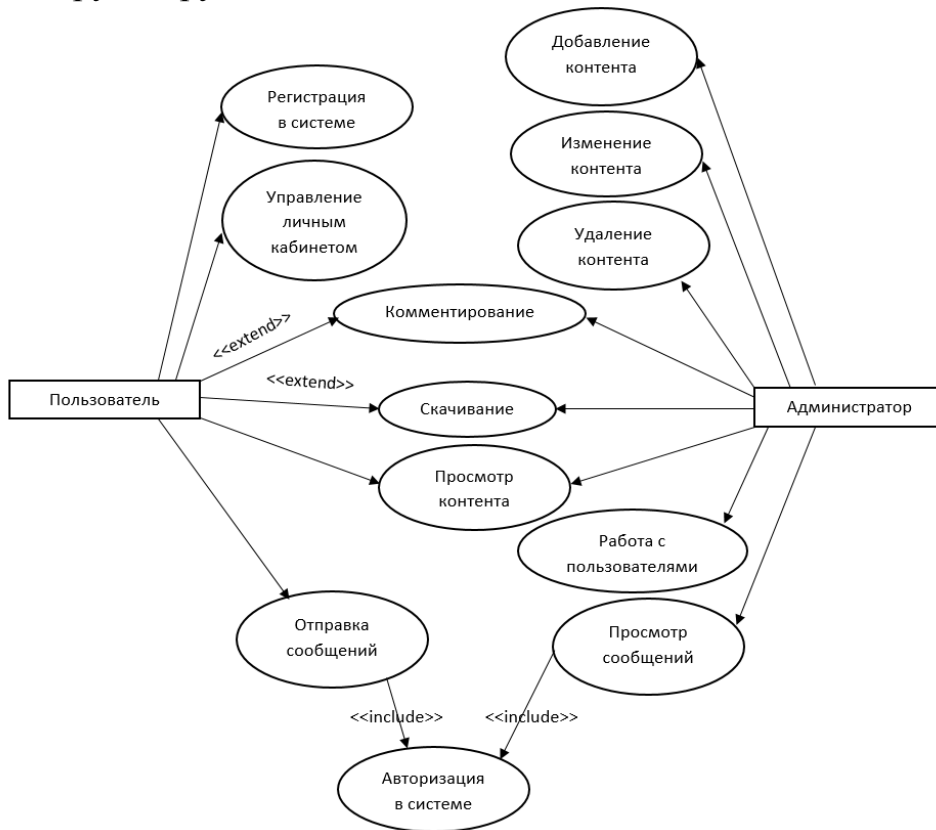


Рисунок 1 – Диаграмма использования

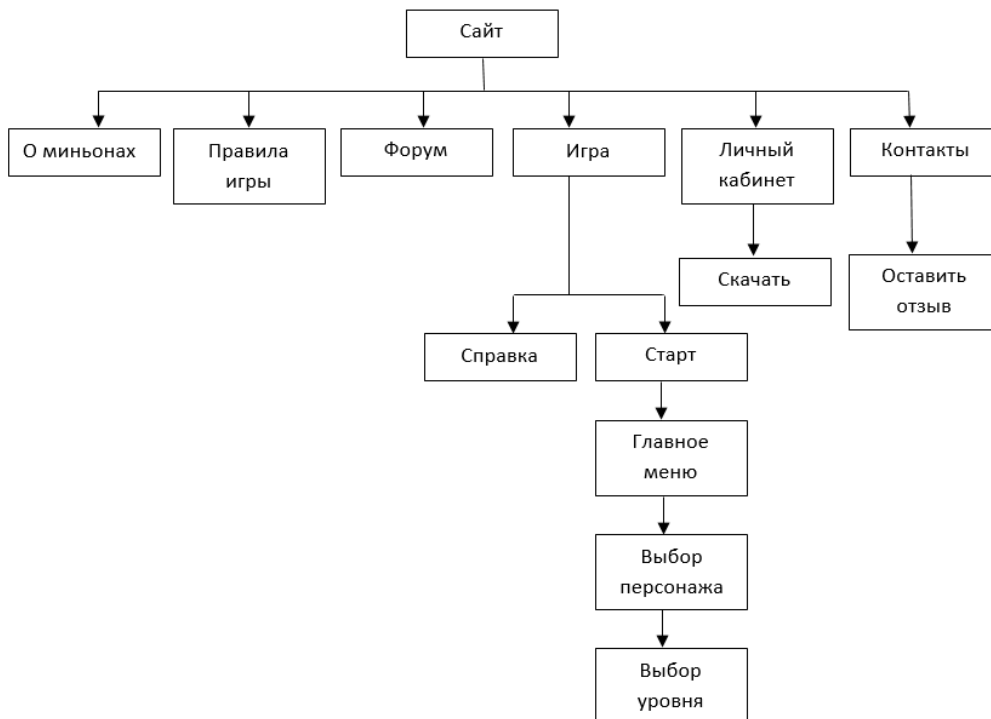


Рисунок 2 – Информационная структура приложения

Как видно из схемы, с главной страницы сайта можно перейти на страницы «О миньонах», «Правила игры», «Форум», «Игра», «Личный кабинет» (при условии, что пользователь прошел авторизацию), страница «Контакты». От продуманности и грамотности информационной структуры сайта зависит удобство использования интернет-ресурса посетителем.

На главную страницу можно перейти с любой страницы сайта. Также с любой страницы сайта можно перейти на любую страницу сайта (на страницу «Личный кабинет» только при условии, что пользователь прошел авторизацию).

ЛИТЕРАТУРА

1. Википедия – свободная энциклопедия [Электронный ресурс] / Wikimedia Foundation, Inc. – 2012. – Режим доступа: <https://ru.wikipedia.org/wiki/Симулятор>. – Дата доступа: 06.04.2019.

2. Википедия – свободная энциклопедия [Электронный ресурс] / Wikimedia Foundation, Inc. – 2012. – Режим доступа: [https://ru.wikipedia.org/wiki/Unity_\(игровой_движок\)](https://ru.wikipedia.org/wiki/Unity_(игровой_движок)). – Дата доступа: 05.04.2019.

3. Википедия – свободная энциклопедия [Электронный ресурс] / Wikimedia Foundation, Inc. – 2012. – Режим доступа: [https://ru.wikipedia.org/wiki/Бот_\(программа\)](https://ru.wikipedia.org/wiki/Бот_(программа)). – Дата доступа: 04.04.2019.

4. Википедия – свободная энциклопедия [Электронный ресурс] / Wikimedia Foundation, Inc. – 2012. – Режим доступа: https://ru.wikipedia.org/wiki/Диаграмма_прецедентов. – Дата доступа: 05.04.2019.

УДК 004.43:004.735

Студ. В.С. Кантарович

Науч. рук. ст. преп. Т.П. Брусенцова
(кафедра информатики и веб-дизайна, БГТУ)

ИНТЕРНЕТ-ИЗДАНИЕ О ЗИМНИХ ВИДАХ СПОРТА «MISHA»

Интернет-издание, как и печатный журнал, является серьёзным новостным и информационным порталом, предоставляя читателю не просто факты и мнения. Информация в журналах носит, в основном, познавательный и в какой-то мере учебный характер. Спортивный журнал предназначен любителям спорта и фанатам. Однако читатели спортивного журнала не ограничиваются категорией любителей спорта. Ими могут быть люди, которые заинтересованы в каком-то одном определённом соревновании или увлечены одним спортсменом. Спортивный журнал включает информацию, которая разнообразна по видам спорта и времени их проведению, отражает ход и результаты различных этапов

соревнований. Так как у журнала будет электронная копия с транслированием главных событий из мира спорта, то он остаётся одним из основных оперативных источников новой информации.

Цель проекта: разработка интернет-издания, позволяющего иметь онлайн доступ к печатному периодическому изданию с функционалом: просмотр журнала, скачивание журнала, подписка на журнал, просмотр архива, наблюдения за результатами соревнований и отслеживание последних спортивных новостей, с целью популяризации зимних спортивных видов спорта среди населения РБ.

Задачи проекта:

- популяризация зимних видов спорта среди населения РБ;
- обеспечение пользователей информацией о предстоящих соревнованиях и различных других спортивных мероприятиях;
- привлечение новых пользователей для денежной прибыли;
- отслеживание популярности журнала и монетизация, за счёт продажи выпусков и подписок, реализованных через сайт.

Задачи программного продукта:

- отслеживание и информирование о текущих соревнованиях и их результатах;
- разработка функции скачивания и просмотра журнала;
- разработка админ-панели для добавления новой информации на сайт и для контроля подписок пользователей;
- создание информативного и привлекательного периодического издания;
- организации базы данных журналов.

На сайте должно происходить постоянное обновление новостей, результатов соревнований и победителей, как и на любом новостном сайте. Также необходима реализация возможности скачивания электронной копии журнала зарегистрированным пользователям, оформившим подписку, а также возможность просмотра последнего выпуска с функцией листания страниц. Разработка админ панели необходима для осуществления добавления заголовка, описания новости, изображения через удобный редактор, после которого автоматически вся новая информация будет заноситься в базу данных.

База данных содержит 7 таблиц, представлена на рисунке 1.

Таблица «Новости» содержит в себе идентификатор, заголовок, описание, дату, категорию, то есть вид спорта, к которому относится, изображение и количество просмотров, которые отображают каждую новость, представленную на сайте.

Таблица «Комментарии» включает идентификационный номер комментария, его статус (подтверждён он администратором или нет),

текст, номер пользователя, который его оставил, и номер новости, к которой он прикреплен.

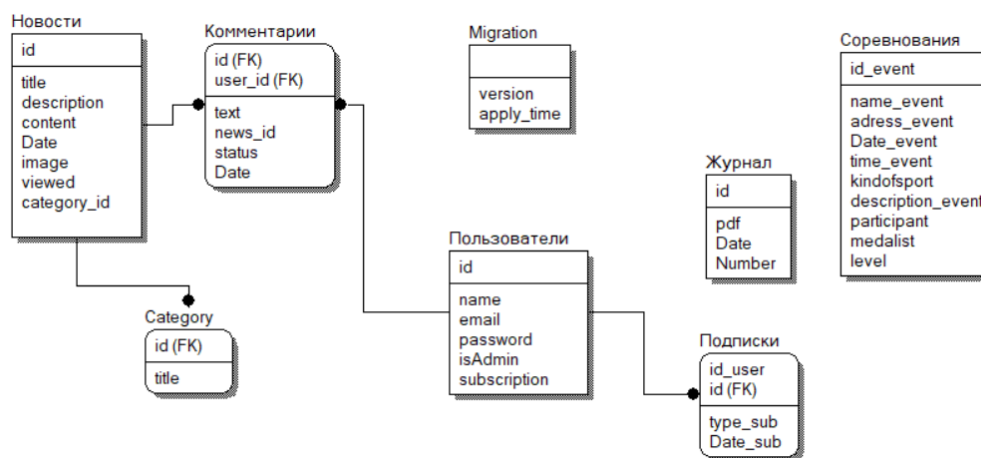


Рисунок 1 – Таблицы базы данных

Таблица «Пользователи» необходима для записи новых зарегистрированных пользователей сайта. Включает номер пользователя, его имя, почтовый ящик, зашифрованный пароль, список его подписок и его статус (администратор или нет).

Таблица «Category» содержит добавленные категории, которые относятся к новостям, то есть вид спорта.

Таблица «Журнал» состоит из идентификационного номера журнала, его версии в формате pdf, даты выпуска и номера издания.

Таблица «Соревнований» необходима для хранения списка всех соревнований в разных категориях, таких как «Международные», «Национальные», «Юношеские». Включает номер, название, описание, дату, время соревнования, адрес прохождения, вид спорта, а также результаты по прошедшему соревнованию и категорию.

Таблица «Подписки» включает номер пользователя, оформившего подписку, её тип и дату.

Функциональная схема сайта, представленная в формате карты сайта, отображена на рисунке 2.

С главной страницы, она же страница «Новости», осуществляется переход на 4 блока. В разделе «Соревнования» доступны все соревнования по зимним видам спорта в таких категориях как «Международные», «Национальные» и «Юношеские». Также для каждого подраздела отображаются отдельно результаты, как прошедших соревнований, так и таблиц результатов по текущему сезону. В разделе «Регистрация/Авторизация» доступна регистрация и вход на сайт, для дальнейшего оформления подписки. Раздел «Подписаться» необходим для оформления подписки на журнал, которая включает в себя

два тарифа. В разделе «Архив» доступны все выпуски журналов, а также существует возможность неавторизованному пользователю в электронном виде просмотреть текущий выпуск журнала.

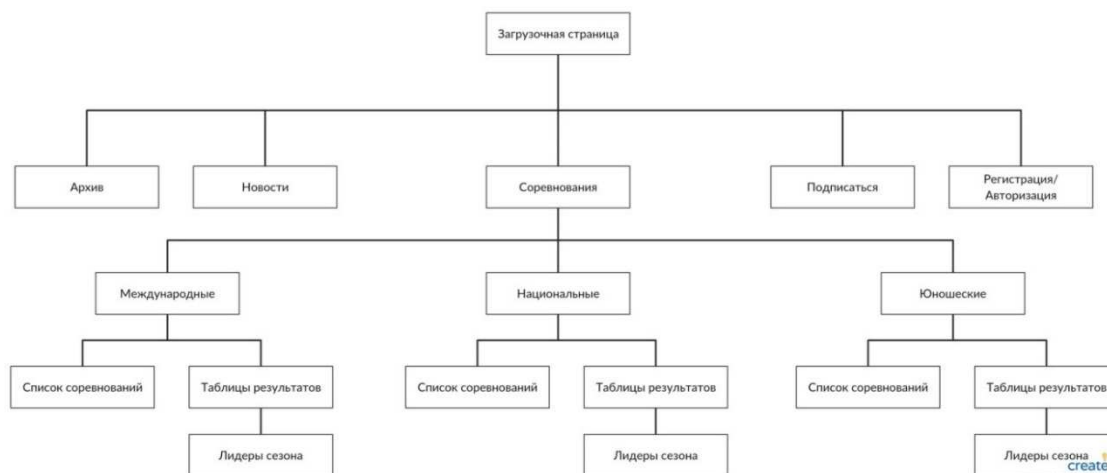


Рисунок 2 – Функциональная схема

Сам журнал необходим для дальнейшего создания и продвижения интернет-издания. Хороший журнал состоит из 40% наполнения, подразумевает тему, направленную на свою аудиторию, из 20% оформления, то есть формат, плотность бумаги и так далее, для привлечения покупателя, и из 40% дизайна, который направлен на пользователя. В нём должны быть соблюдены правила UX, для удобства, правильного расположения информации.

УДК 339.138

Студ. Е.И. Жаденова

Научн. рук. ст. преп. Н.И. Потапенко
(кафедра информатики и веб-дизайна)

ТЕХНОЛОГИЯ WEBGL В ДИЗАЙНЕ САЙТОВ

Появление в HTML5 тега canvas привело к появлению технологии WebGL для рендера 3D изображений. Создание приложений на WebGL, используя нативный JavaScript несколько затруднительно, поэтому различными компаниями были разработаны фреймворки, работающие исключительно с 3D визуализацией. Одним из таких фреймворков является three.js. Это позволило создавать в браузере 3D игры, сцены и эффекты. Пока большинство из них, но некоторые могут быть уже сегодня использоваться на живых сайтах.

Есть мнение, что WebGL используется как API для 3D. На самом деле WebGL – это средство растеризации. Он отображает точки, линии и треугольники на основе написанного кода.

WebGL позволяет веб-контенту использовать API, основанный на OpenGL ES 2.0 для визуализации трехмерной графики без использования элемента canvas в браузерах. WebGL программы состоят из кода управления, написанного на JavaScript, и кода специальных эффектов (шейдерного кода), который выполняется на графическом процессоре. WebGL элементы могут быть смешаны с другими HTML элементами и собраны с другими частями веб-страницы или фоном веб-страницы [1].

Шейдерный код представлен в виде пары функций. Эти две функции – вершинный и фрагментный шейдер, обе они написаны на очень строго типизированном языке, подобном C/C++, который называется GLSL. (GL Shader Language). Вместе эта пара функций называется *программа*. Задача вершинного шейдера – вычислять положения вершин. Основываясь на положениях вершин, которые возвращает функция, WebGL может растеризовать различные примитивы, включая точки, линии или треугольники. В процессе растеризации этих примитивов WebGL прибегает к использованию второй функции – фрагментному шейдеру. Задача фрагментного шейдера – вычислять цвет для каждого пикселя примитива, который в данный момент отрисовывается [2].

Практически всё API WebGL заключается в настройке состояния для работы этих двух функций. Любые данные, которые будут использованы в этих двух функциях, должны быть переданы на графический процессор. Есть четыре способа, которыми шейдер может получить данные:

- атрибуты и буферы. Буферы – это массивы бинарных данных, загруженных в графический процессор. Обычно буферы содержат положения вершин, нормалей, координаты текстур, цветов вершин и т.д., хотя вы вольны положить в них что угодно. Атрибуты определяют, каким образом данные из буферов передаются в вершинный шейдер. Доступ к буферам не произвольный. Вместо этого вершинный шейдер выполняется заданное количество раз и каждый раз, когда он выполняется, выбирается следующее значение каждого из указанных буферов и назначается атрибуту;

- uniform-переменные. Uniform-переменные – это глобальные переменные, которые устанавливаются перед выполнением программы шейдера;

- текстуры. Текстуры – это массивы данных, к которым есть произвольный доступ в программе шейдера. Чаще всего в текстуру помещается картинка, но текстура – это просто набор данных можно запросто поместить в неё что-то отличное от набора цветов;

– varying-переменные. Varying-переменные позволяют передавать данные из вершинного шейдера фрагментному шейдеру. Во фрагментном шейдере получают интерполированные значения вершинного шейдера – зависит от того, отображаются ли точки, линии или треугольники [3–5].

Three.js – это библиотека, которая опирается на возможности WebGL и делает над ним надстройку, что позволяет работать с WebGL более удобным образом и писать меньше строчек кода.

Использование библиотеки three.js, а не чистого WebGL, даёт быстрый результат и удобство. Например, сделать первую сцену, вывести 3D модель и сразу иметь возможность получить результат.

Three.js выполняет всю низкоуровневую работу с WebGL, предоставляя набор классов с понятным интерфейсом для удобной работы с 3D графикой, и кроме этого за счёт скрытия сложных механизмов отрисовки 3D объектов значительно упрощает разработку [6].

Рассмотрим, что представляет 3D изображение в терминах библиотеки three.js.

Основой 3D пространства является рендерер. Рендерер – это сущность, отображающая на canvas выбранную сцену с выбранной камерой. Сцена – это 3D пространство, в котором располагаются нужные объекты: элементы (меш) и источники света, которые освещают элементы с какой-либо стороны. Рендерер, чтобы сделать отображение чего-либо на canvas, всегда принимает в себя сцену и камеру. Именно эти две ключевые сущности формируют отображение пространства на полотно canvas.

Источник света – элемент, который создаёт освещение. Он может быть рассеянным, точечным, направленным.

Следующая важная сущность – это меш. Меш – это элемент сцены, который состоит из геометрии и материала. Поскольку любой 3D объект является сложным.

Геометрия – это набор вершин, которые при генерации соединяются между собой графическими примитивами. Если взять простейшую плоскость как геометрию, то в рамках этой плоскости точки при отображении соединяются между собой прямыми линиями, и виден результат.

Материал – понятие намного более сложное, чем цвет. Материал – это способ отображения и внешний вид элемента. Какие-то элементы отражают свет и/или отбрасывают тени, а какие-то нет. Материал ведёт себя по-разному в рамках представления на сцене. Например, тени и отражение.

Текстура – это изображение, которое может использоваться в рамках материала, чтобы задать внешний вид объекта.

На рисунке 1 представлена взаимосвязь сущностей рендерера.

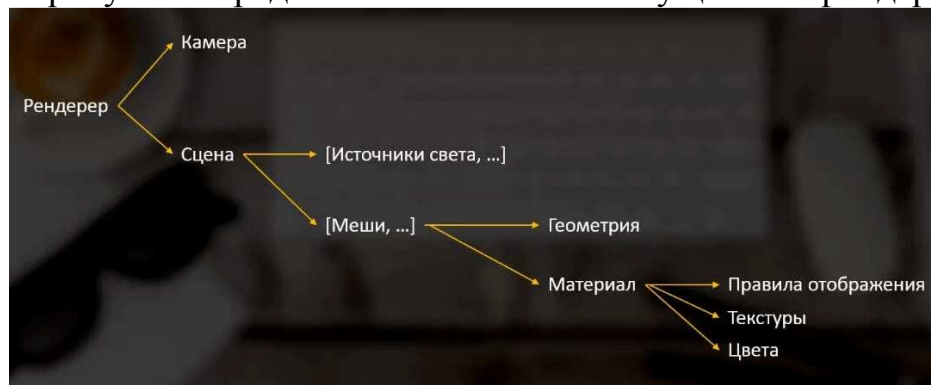


Рисунок 1 – Взаимосвязь сущностей рендерера

Подводя итог вышерассмотренному, можно сделать вывод, что WebGL – технология, позволяющая создавать запоминающиеся интерактивные веб-приложения и объёмные визуализации данных. Однако, несмотря на то, что большинство браузеров поддерживает эту технологию, не так много ещё разработчиков её используют. Технология WebGL со временем займёт достойное место в арсенале веб-разработчиков.

ЛИТЕРАТУРА

1. Подготовка к использованию WebGL / Frontender Magazine [Электронный ресурс]. – 2013. – Режим доступа: https://frontender.info/Getting_started_with_WebGL/. – Дата доступа: 19.03.2019.
2. WebGL tutorial / Mozilla [Электронный ресурс]. – 2019. – Режим доступа: https://developer.mozilla.org/ru/docs/Web/API/WebGL_API/Tutorial. – Дата доступа: 01.04.2019.
3. WebGL Fundamentals / WebGLFundamentals [Электронный ресурс]. – 2017. – Режим доступа: <https://webglfundamentals.org/webgl/lessons/ru/webgl-fundamentals.html>. – Дата доступа: 10.03.2019.
4. Основы WebGL / DevBurn [Электронный ресурс]. – 2019. – Режим доступа: <https://devburn.ru/2017/04/22/>. – Дата доступа: 27.03.2019.
5. Learn WebGL / LearnWebGL [Электронный ресурс]. – 2018. – Режим доступа: <http://learnwebgl.brown37.net/>. – Дата доступа: 25.03.2019.
6. Three.js for beginners / Medium [Электронный ресурс]. – 2019. – Режим доступа: <https://medium.com/@PavelLaptev/three-js-for-beginners-32ce451aabda>. – Дата доступа: 10.03.2019.

Студ. Н. В. Попеня
Науч. рук. ст. преп. Т.В. Кишкурно
(кафедра информатики и веб-дизайна, БГТУ)

ПОВЫШЕНИЕ КОНВЕРСИИ САЙТА С ПОМОЩЬЮ УВЕЛИЧЕНИЯ КЛИКАБЕЛЬНОСТИ

Кнопки играют важную роль в процессе перехода на страницы покупок товаров в интернет-магазинах, а также множестве других задач. Однако следует помнить, что неправильно оформленные кнопки имеют низкую кликабельность и, как следствие, низкую конверсию. Прежде чем пользователь дойдет до финального этапа клика по кнопке, он должен захотеть это сделать. Кнопка должна привлечь пользователя внимание, но при этом не оттолкнуть. Изменяя только ее цвет и форму можно значительно повысить рейтинг кликабельности кнопки [1]. Следуя некоторым простым правилам можно значительно повысить этот показатель:

1. Наличие кнопки. Для того чтобы по кнопке кликали, пользователям необходимо понимать, что перед ними кнопка. Для этого их необходимо делать понятными и узнаваемыми, кнопки должны быть похожи на кнопки.

Форма может быть любой (круглая, квадратная, прямоугольная), но наибольшее распространение и популярная именно прямоугольная, так как на ней можно уместить надпись с её предназначением. Большинство кнопок в физическом мире имеют закругленные углы, и такие кнопки на сайте проще узнать. К тому же, скругленные углы привычнее и потому проще воспринимаются человеком.

Кнопку лучше делать объемной, для этого используется тень, градиент и рамка. Тренд последних лет – плоский дизайн. Но один из явных его недостатков заключается в том, что из-за отсутствия теней и объема кнопки сложнее отличить от некликабельных элементов дизайна. Кнопку желательно делать отзывчивой, то есть при наведении курсора менять её цвет и объем. Таким образом, ещё при наведении курсора становится понятно, что элемент кликабелен.

2. Приоритет кнопки. Не стоит перегружать страницу множеством кнопок, это рассеивает внимание, создает лишние вопросы и, как следствие, снижает кликабельность. Следует использовать кнопку, которая использует явный приоритет.

3. Цвет кнопок. Главная кнопка на странице должна быть контрастного цвета, чтобы выделяться от фона и окружающих элементов. Для менее приоритетных кнопок лучше не использовать цвет,

сделав их прозрачными и поместив в рамку – это так называемые кнопки-призраки.

4. Размер кнопок. Размер кнопок важен по двум основным причинам: с помощью крупного размера можно привлечь внимание, поэтому главную кнопку лучше делать крупнее; в мобильных устройствах отсутствует курсор мыши, а палец человека больше курсора, поэтому и кнопки необходимо делать больше, чтобы по ним удобно было попадать подушечками пальцев. В Массачусетском технологическом институте провели исследование, в рамках которого выявили, что средний размер подушечек пальцев составляет от 10 до 14 мм, а кончики пальцев – от 8 до 10 мм [2]. Поэтому рекомендуемый минимальный размер кнопки 10×10 мм.

5. Расположение кнопок. Кнопки должны быть заметны, для этого их лучше всего размещать там, где их ожидают увидеть, например, рядом с товаром на странице товара или рядом с итоговой ценой в корзине. Главную кнопку на странице необходимо размещать в первом экране. Не заставляйте пользователей искать важные кнопки по сайту, это снизит конверсию.

6. Текст на кнопке. Текст должен оставаться понятным, надписи на кнопке требуют подсказывать, что ожидает человека после клика, и побуждать к действию. В противном случае, кликабельность кнопки будет ниже. Другой вариант – это изменяющиеся надписи на кнопках при наведении. По ним пользователь сразу понимает, какое действие он совершит, кликнув на кнопку. Это снижает неопределенность и повышает кликабельность. Помимо надписей, на кнопку можно вынести значок, так становится гораздо понятнее.

Желательно указывать подписи к кнопкам и значкам. Без подписи кликабельность снижается, а вместе с ней и конверсия.

Чтобы кликабельность кнопки была повышена, а, следовательно, и конверсия, требуется выполнять все пожелания в комплексе. Изменение одного из параметров может только незначительно повлиять на результат работы.

Следует учитывать множество факторов, которые подтолкнут посетителя вашего сайта к действию:

1. Построение пути пользователя к кнопке;
2. Продумывание дизайна кнопки;
3. Рассчитывание размера области кнопки;
4. Формулировка надписи на кнопке и пояснение текста рядом;
5. Поиск оптимального положение;
6. Тестирование результата.

ЛИТЕРАТУРА

1. Хабрахабр [Электронный ресурс]. – 2019. – Режим доступа: <https://habr.com/ru/hub/conversion/> – Дата доступа: 27.03.2019
2. Dandekar K, Raju BI, Srinivasan MA 3-D finite-element models of human and monkey fingertips to investigate the mechanics of tactile sense/ Touch Lab, Department of Mechanical Engineering, Research Laboratory of Electronics, Massachusetts Institute of Technology – Cambridge, 2003. – 10 с.

УДК 339.138

Студ. Н.В. Попеня

Науч. рук. ст. преп. Н. И. Потапенко
(кафедра информатики и веб-дизайна, БГТУ)

О ТЕХНОЛОГИИ СОЗДАНИЯ АВТОРСКОГО ШРИФТА

Шрифты окружают нас в повседневной жизни. Любой товар в магазине, любой веб-сайт, который вы просматриваете в Интернете, и любой рекламный баннер на улице содержат хотя бы одно слово или фразу, набранную тяжёлым шрифтом или ручной прописью.

Шрифты бывают разными, в зависимости от цели дизайна. Могут быть геометрически точными или кривыми с прыгающими буквами. Рукописные экранные шрифты становятся все более модными, освежая дизайн [2].

Однако бывают ситуации, когда подобрать тот самый шрифт, который подойдет к работе, достаточно сложно. Возникает одно из решений – создание собственного шрифта. Разработка требует множества решений и исследований. Требуется определиться, как именно будет использоваться шрифт, будет ли он уникальным. Требуется четкое понимание того, каким должен быть законченный результат работы.

Важным этапом в проектировании шрифта является выбор художественного облика шрифта. Будет ли он декоративным, изящным или строгим. Далее нужно определиться с характером графического построения. Поскольку в зависимости от выбора будет построена дальнейшая работа. Считают, что дизайн рубленного шрифта в разработке сложнее за счёт его специфики.

На ранних этапах работы могут возникнуть некоторые подводные камни. Например, компьютеризация своего рукописного шрифта из-за своей индивидуальности может не иметь большого успеха. А взятый за основу существующий шрифт не позволит получить лучший результат и не разовьет навыки.

Существует множество материалов на тему того, как рисовать шрифты с помощью компьютерных программ, однако рекомендуется сначала нарисовать его от руки. Попытки сделать это на компьютере существенно усложнят вашу работу. При разработке шрифта рекомендуется создать формы первых букв на бумаге, а только потом начинать компьютерную работу. Следующие буквы будут сконструированы на основе уже существующих форм, в соответствии с ключевыми особенностями [1].

Создание сначала базовых или «контрольных» символов помогает задать уникальный стиль своего собственного шрифта. Позже они могут быть использованы как направляющие. Обычно в латинице – это «n» и «o», а заглавные – «N» и «O».

Для переноса шрифта на компьютер можно использовать трассировочные программы, однако предпочтительнее сделать это вручную, чтобы иметь полный контроль над точками.

Выбор программы, в которой будет проводиться дальнейшая работа, также является сложным шагом из-за многообразия программного обеспечения. Некоторые разработчики рекомендуют использовать Adobe Illustrator. Эта программа отлично подходит для рисования отдельных форм и экспериментов, однако для создания шрифтов она не подходит. Лучше начать работу в программе, которая позволит сразу управлять характеристиками шрифта.

Fontstruct это веб-приложение, являющееся замечательным средством для создания шрифтов, особенно для новичков в этом деле. Для создания шрифтов нужно зарегистрироваться. Графический интерфейс приложения настолько прост и удобен в использовании, что можно начать пользоваться им моментально. После создания шрифта, его можно скачать в формате TrueType[2].

Если же рассматривать десктопные приложения, можно рассмотреть FontForge: бесплатную программу, позволяющую создавать собственные шрифты. Она содержит большое количество настроек и инструментов. На главном экране находится растровое изображение знаков и символов, которые можно формовать и переделывать, в соответствии с желаемым дизайном. FontForge совместим с множеством различных форматов шрифтов, включающих в себя TrueType и OpenType.

TypeTool является мощным редактором шрифтов, который способен работать с наборами, содержащими до 65 000 символов [3]. Поддерживаются шрифты семейства TrueType, Type 1 и OpenType шрифты, в основе которых лежит TrueType. С помощью этого редактора можно создавать новые шрифты, редактировать существующие и открывать любые виды шрифтов, установленных в системе.

После работы с созданием форм обязательно нужно посмотреть, как шрифт выглядит в словах. Требуется проанализировать его внешний вид в строке и абзаце. Это можно делать на протяжении всей разработки шрифта, не дожидаясь создания всего алфавита. Нужно понять, как шрифт будет реагировать при изменении его размера, а также других действиях – изменения начертания, переносе и добавлении эффектов.

Следующим этапом может быть печать, если шрифт предназначен для бумажных носителей. Нужно критично оценить его вид на бумаге. Гораздо проще увидеть недочеты, если просмотреть на шрифт под различными ракурсами.

Собственный шрифт стоит создать, потому что это отличная возможность глубже понять полиграфию. Научившись совмещать шрифты, вы наберёте опыта в дизайне. Этот навык пригодится для дальнейшей работы, потому что в своих проектах лучше использовать собственные шрифты, не заботясь о нужной лицензии в зависимости от цели вашего конечного продукта.

ЛИТЕРАТУРА

1. Буковецкая О. А. Дизайн текста: шрифт, эффекты, цвет – 2-е изд., испр. – М.: ДМК Пресс. – 278 с.
2. Комолова Н. В. Компьютерная верстка и дизайн – СПб.: БХВ-Петербург, 2003. – 512 с.
3. Хабрахабр [Электронный ресурс]. – 2019. – Режим доступа: <https://habr.com/ru/hub/typography/> – Дата доступа: 30.03.2019

УДК 004.9, 004.438

Студ. А. В. Хамутовская
Науч. рук. доц. О. А. Новосельская
(кафедра информатики и веб-дизайна, БГТУ)

ON-LINE ИГРА «CUCARACHA»

На сегодняшний день игровая индустрия стремительно развивается [1]. С тех пор как появился сотовый телефон, перед разработчиками возник вопрос о внедрении не только средств связи, но и развлечения. Одной из первых игр, появившихся на телефонах, была игра «змейка» – аналог PC-версий. Она стала прообразом современных браузерных игр жанра Эволюция. Суть игр в этом жанре заключается в управлении одноклеточными организмами, которым необходимо съесть организмы других игроков и, помимо этого, питательную еду на поле, обеспечивающую рост до большего количества клеток (размеров) [2].

Главным героем игры данного проекта был выбран таракан. Это решение обосновано тем, что игра с таким персонажем раньше еще не было. Также подтекстом в этой игре будет обучение от обратного – чем меньше мусоришь, тем меньше тараканов.

Название было взято в результате перевода слова таракан на испанский язык и созвучно одноименной песне «Cucaracha». Разработка является первым белорусским аналогом игр серии IO с непохожим персонажем и тематикой.

Cucaracha – многопользовательская браузерная игра, в которой игрок контролирует одну или несколько клеток. В игре есть нисходящая перспектива на карте, размер которой и видимая область игрока основана на массе и подсчете их клеток. Цель игры – вырастить игрока как можно большего размера, поглощая пищу или других игроков меньшего размера. Сама игра не имеет конца. Игроки могут присоединиться к постоянной игре в любой момент времени. Игрок начинают игру как маленький тараканчик в окружении других игроков различных размеров. Игра имеет простое управление. Положение курсора на экране определяет движение игрока. При наборе массы игрок начинает двигаться медленнее, что затрудняет в дальнейшем его рост, поэтому он также должен иметь возможность «разделиться», при условии достаточной массы игрока. Вэтом случае скорость движения восстанавливается и число тараканов будет зависеть от конечной массы тела. Деление осуществляется на части одинаковой массы, каждая часть составляет ровно половину массы оригинального игрока.

Анализ игр подобного типа показал, что привлечению целевой аудитории будут способствовать качественный функционал и дизайн. В первую очередь был сделан именно функционал как каркас сайта. Определены обязательные функции: управление мышкой, лидерчарт, увеличение размера при поглощении еды. Это то, без чего игра браузерного типа не смогла бы функционировать. Функциональная схема представлена на рисунке 1. Игра разработана в среде NodeJS с использованием Socket.IO для создания сервера WebSocket.

Для того чтобы структурировать функционал и дизайн будущей игры, был разработан прототип в онлайн-сервисе Cacoо (рисунки 2, 3).

На рисунке 3 красные квадратики символизируют еду, разбросанную по полю.

Для оформления игры было принято решение использовать яркие, насыщенные цвета, поскольку именно цветовое оформление может существенно повлиять на восприятие человеком сайта целиком, не только его главной страницы. Хотя персонажем игры был выбран таракан, его визуальное восприятие не должно вызывать негатива, по-

этому отрисовка тараканов должна быть тщательной, они должны быть милыми и приятными глазу.



Рисунок 1– Функциональная схема



Рисунок 2– Прототип главной страницы

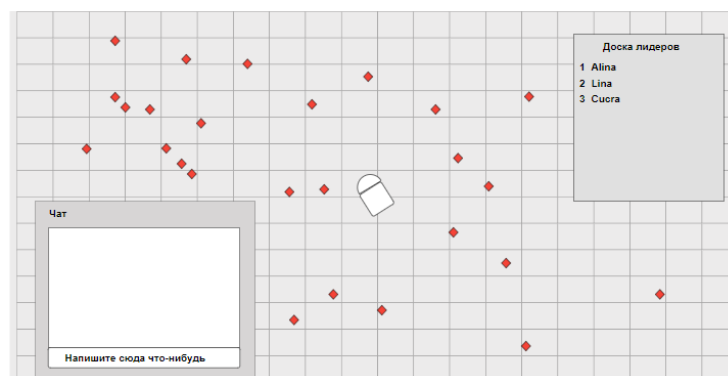


Рисунок 3– Игровое поле

Векторная графика используется на любых ресурсах, независимо от направленности и масштаба. Именно поэтому персонажи игры были разработаны в программе векторной графики. На сайте продуман вариант выбора внешнего вида игрового персонажа, который осуществляется на главной странице. При этом продуман простой и удобный интерфейс, разделенный на блоки. Правую часть главной страницы занимает блок входа в игру (рисунок 4). В нем находится поле для ввода имени, слайдер для выбора таракана для игры, кнопки играть,

настройки и наблюдение. В данный момент внешний вид находится в разработке, поэтому представленный на рисунке 3 персонаж отображается численно. Всего разработано четыре персонажа. Основным цветом главной страницы выбран оттенок фиолетового, а весь текст оформлен светлыми серыми цветами. Поле с игрой (рисунок 5) пока имеет клетчатую подложку, что облегчает ориентирование на местности, однако в дальнейшем планируется использование клеток на слабом текстурном фоне рельефа. Верхний правый угол занимает блок со списком лидеров, он имеет темную полупрозрачную заливку. Нижний левый угол занимает блок с чатом для игроков. В мобильной версии игры чат отсутствует.

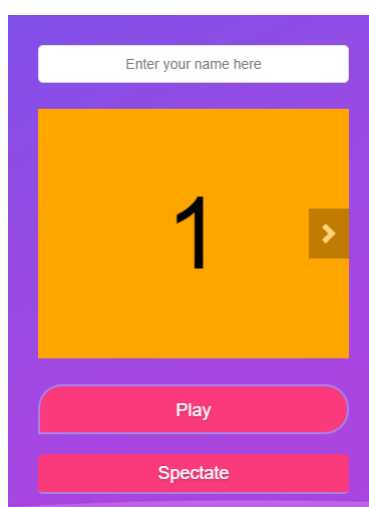


Рисунок 4– Вход в игру



Рисунок 5– Мобильная версия

Игра не предусматривает регистрации. Чтобы начать играть, нужно просто ввести имя на главной странице (или не вводить) и нажать кнопку Play. Монетизация ресурса предусматривает следующие позиции: размещение ссылок на другие ресурсы, реклама после каждого проигрыша, размещение баннеров. По возможности, это останется максимально пассивным на первоначальном этапе.

ЛИТЕРАТУРА

1. Индустрия компьютерных игр [Электронный ресурс] / Википедия – свободная энциклопедия. – 2001–2019. – Режим доступа: <https://ru.wikipedia.org/wiki/Компьютерные%Игры>. – Дата доступа: 24.03.2019.

2. Игровой портал Slither.io [Электронный ресурс] / Slither.io. – Режим доступа: slither.io. – Дата доступа: 18.03.2019.

УДК 004.738.1:004.738.5

Студ. Н. Н. Ромашко
Науч. рук. Т. В. Кишкурно
(кафедра информатики и веб-дизайна, БГТУ)

РАЗРАБОТКА И СОЗДАНИЕ САЙТА КАФЕ «ВАФЛИ ЕСТЬ»

Создание Web-сайтов является одной из важнейших технологий разработки ресурсов Интернет. Хороший сайт, вбирая в себя всю полезную информацию, является лучшей визитной карточкой и коммерческой фирмы, работая на них в любое время суток.

Целью моей работы было создание сайта кафе «ВАФЛИ ЕСТЬ» и размещение его в интернете. Сайт должен быть визиткой для данного кафе. При разработке сайта использовалась информация о кафе. Передо мной стояла задача разместить на сайте следующие элементы: меню данного кафе, фотоматериалы о данном кафе, контактную информацию. Сайт должен иметь понятную навигацию, быть простым, удобным и привлекательным.

Структурная схема сайта представлена на рисунке 1. Сайт разработан таким образом, что находясь на любой странице, вы можете попасть в любой необходимый вам раздел. Это упрощает навигацию пользователя, делая её очень простой и удобной. Главная страница содержит информацию обо всех разделах сайта, а так же имеется возможность по ссылке перейти на нужный раздел для получения более полной информации (рисунок 1).

Для индивидуального представления данного кафе был разработан собственный шрифт. Буквы разрабатывались с помощью Adobe Illustrator (рисунок 2).



Рисунок 1 – Структурная схема сайта

А аБ б

Рисунок 2 – Разработанные буквы

А после создания, они загружались по отдельности в программу FontCreatorProfessional (рисунок 3).

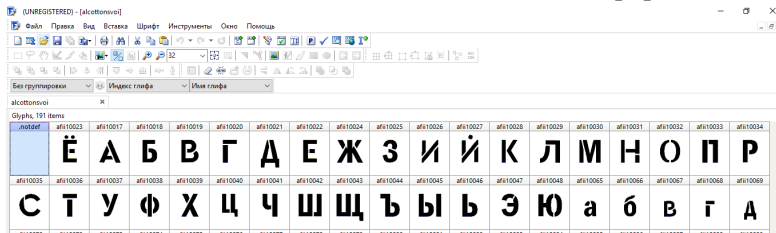


Рисунок 3 – Создание собственного шрифта

Палитра цветов сайта состоит из нежно- фиолетового, черного и белого цвета, что показывает современность данного кафе. Каждая страница сайта выполнена в одном стиле (рисунок 4).

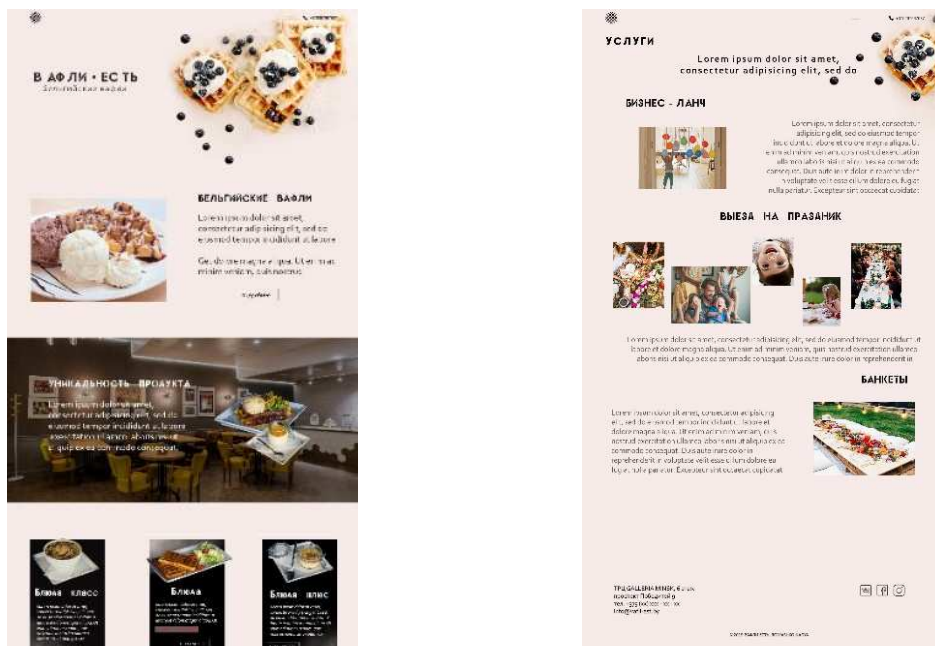


Рисунок 4 – Страницы сайта

Для лучшего представления внешнего вида кафе, расположения столиков была создана 3 d модель данного кафе. Модель разработана с помощью 3dsmax, а при размещении на самом сайте использовался сервис A360 (рисунок5). Для работы с кодом использовались программы VisualCode.Для редактирования изображений была использована программа Adobe Photoshop CC 2019.



Рисунок5– 3d модель кафе

В результате выполненной работы создан сайт кафе «ВАФЛИ ЕСТЬ», который соответствует требованиям заказчика, простой удобный, привлекательный, с понятной схемой навигации и с требуемым контентом.

ЛИТЕРАТУРА

1. Попов Е. – Уроки по созданию сайтов <http://ruseller.com/>[Электронный ресурс].
2. Морковин, А. – Видеоуроки «Блочная div-вёрстка сайта» <http://www.sdelaysite.com/>[электронный ресурс].
3. Морковин, А. – Видеоуроки «Основы Java Script» <http://www.sdelaysite.com/>[электронный ресурс].
4. Учебник по HTML <http://www.postroika.ru/html/>[Электронный ресурс].

УДК 004.774:654.197

Студ. Д. В. Буртыль, А. А. Кондрашова
Науч. рук. ст. преп. Т.П. Брусенцова
(кафедра информатики и веб-дизайна, БГТУ)

ОСОБЕННОСТИ И СТРУКТУРА ИНТЕРФЕЙСА В ИНТЕРНЕТ-ТЕЛЕВИДЕНИИ

Интернет-телевидение или онлайн-телевидение – это система, основанная на двусторонней цифровой передаче телевизионного сигнала через интернет-соединения посредством широкополосного подключения. В середине 2000-ых в сети Интернет стали появляться онлайн-версии телевизионных передач. Так, например, iTunes в 2005 году предложил пользователям платный сервис просмотра телепередач онлайн, в том же году на YouTube стали появляться версии популярных программ бесплатно. Далее начли свою работу AmazonVideo, Netflix, Hulu и другие. Свою популярность данные платформы набрали к 2015 году в США и после этого не теряют её.

Для того чтобы смотреть онлайн ТВ, не нужны ни спутниковые антенны, ни тюнеры, ни какое-либо другое специальное оборудование. Онлайн телевидение можно смотреть в любое время и в любом месте, лишь бы там был доступ к сети Интернет. Причем большинство таких порталов абсолютно бесплатны и требуют регистрации.

Среди преимуществ онлайн-телевидения можно выделить:

- поиск прошлых телевизионных передач для просмотра;
- функцию паузы для телеканала в режиме реального времени;
- большинство сервисов абсолютно бесплатные;

- пользователь сам выбирает, что ему смотреть;
- нет необходимости в подключении каких-то пакетов, платных подписках и т. д.;
- минимум рекламы;
- фанаты каких-либо телепередач могут общаться в комментариях на сайте интернет ТВ.

К недостаткам интернет ТВ можно отнести:

- необходимость устойчивого интернет-подключения, а также безлимитного тарифа, т.к. онлайн-телевидение требует большого трафика;
- ограничения просмотра по региону.

На данный момент существует множество сервисов для просмотра ТВ онлайн. Проанализировав некоторые из них, можно выделить следующие три как наиболее примечательные:

- tvplayer;
- BBC iPlayer;
- ABC tv;

На рисунках 1–3 представлены интерфейсы видеоплеера вышеперечисленных сервисов. Рассмотрим интерфейсы приведенных выше сайтов. На всех сайтах присутствует форма регистрации/авторизации пользователей, поиск, ТВ гид. Также есть возможность выбора категории, для которого на одном из сайтов достаточно зайти на главную страницу, а на остальных двух – второстепенную. На двух из них есть подборка рекомендуемых к просмотру каналов, также категории программ, например, спортивные каналы, новостные и т.д., и сортировка по алфавиту. На двух сайтах можно смотреть каналы в режиме реального времени.

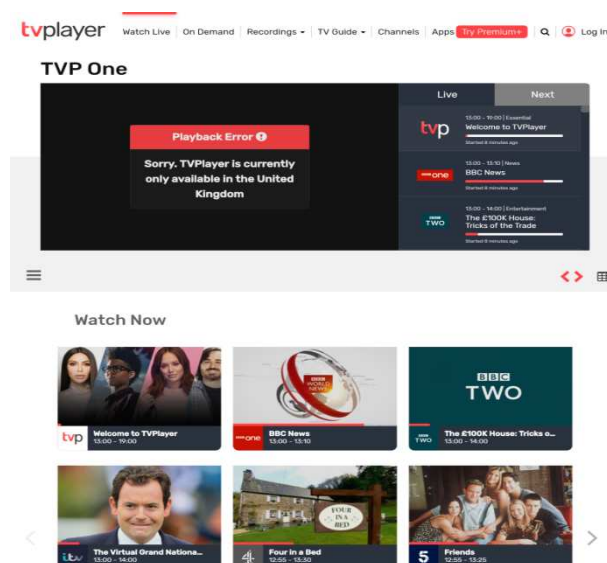


Рисунок 1 – Интерфейс сайта tvplayer.com



Рисунок 2 – Интерфейс сайта BBCone

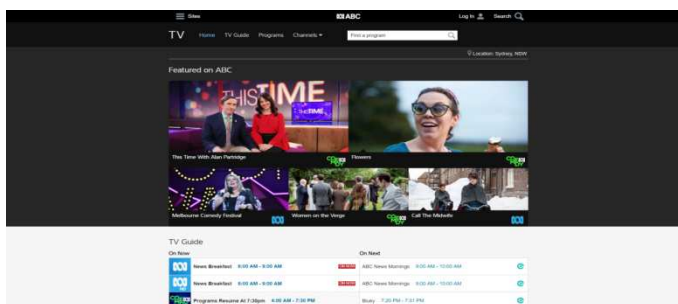


Рисунок 3– Интерфейс сайта ABC tv

На всех просмотренных сайтах плеер кастомизированный. Данные сайты онлайн ТВ выполнены в строгом классическом стиле, меню которых находится обязательно вверху страницы и включает в себя логотип, строку поиска, кнопку регистрации, сами пункты меню и категории, если таковые есть. По цветовой гамме они выполнены довольно-таки скромно: два из них - в темных тонах, один - в светлых, однако на всех трех присутствуют яркие акценты. Анимация на сайтах отсутствует. Общая структура интерфейса представлена на рисунке 4.

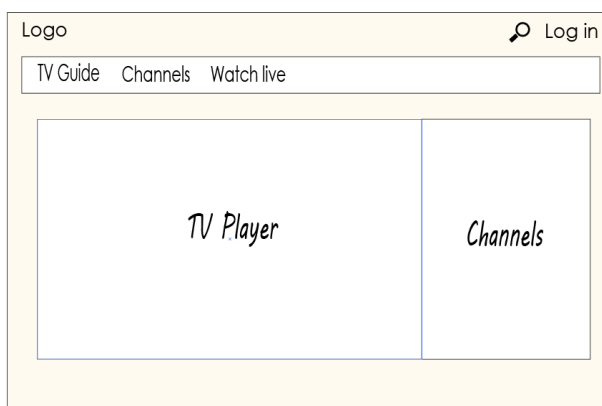


Рисунок4 – Общая структура интерфейса

Вывод: все рассмотренные нами сайты выполнены в классическом стиле, меню горизонтальное и расположено вверху страницы. Везде присутствует строка поиска и регистрация. Логотип чаще всего находится в верхнем левом углу. В пункты меню обязательно входят разделы «каналы» и «ТВ гид». Анимация на сайтах отсутствует.

Студ. А. А. Кондрашова
Науч. рук. ассист. С.А. Осоко
(кафедра информатики и веб-дизайна, БГТУ)

РАЗРАБОТКА СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ ПО ПРОГРАММЕ MSEXCEL

Система дистанционного обучения – набор программных продуктов и решений, который объединяет и автоматизирует все или большую часть процессов, связанных с обучением.

СДО должна автоматизировать актуальные задачи, поэтому она предоставляет пользователю следующие возможности:

- управление всеми видами обучения (электронное, очное и заочное);
- проверка знаний и навыков;
- анализ обучения и оценка результатов;
- предоставление контента и программ;
- архив учебных материалов.

Как правило, дистанционное обучение разбивает на 3 блока, которые обладают специальным функционалом, – управление учебным процессом, взаимодействии между пользователями и разработке учебных материалов.

Причины, по которым СДО имеет место быть:

- вся информация собрана в одном месте;
- свободный доступ к необходимому контенту;
- анализ процесса обучения – возможность отслеживания прогресса обучаемых;
- высокая эффективность;
- обновление материалов.

1. Сложности при изучении программы MSeXcel.

MicrosoftExcel – программа для работы с таблицами. Данный табличный процессор позволяет вводить данные, обрабатывать их и выводить, в том числе и в виде диаграмм и графиков. В Excel имеется большое число встроенных функций такие, как математические, статистические, финансовые, текстовые, информационные и т.д.

Основная проблема работы с данным ПО заключается в том, что пользователи не имеют достаточных знаний для работы с этими функциями: не знают, как они работают, не знают все возможности, предоставляемые программой, не знают, как правильно их использовать.

2. Функционал проекта.

В проекте могут быть задействованы 3 вида пользователей: гость, студент, администратор. Сравнение возможностей для разных

видов пользователей представлено на рисунке 1.

Функционал / Роль	Гость	Студент	Администратор
Чтение теории	✓	✓	✓
Скачивание файлов примеров и упражнений	✗	✓	✓
Прохождение тестов	✗	✓	✗
Чтение комментариев	✓	✓	✓
Добавление комментариев	✗	✓	✗
Удаление комментариев	✗	✗	✓
Просмотр ссылок	✓	✓	✓
Поиск	✓	✓	✓
Статистика по своим тестам	✗	✓	✗
Статистика по всем пользователям	✗	✗	✓

Рисунок 1 – Сравнительная таблица возможностей разных пользователей

Гость – незарегистрированный пользователь на сайте, соответственно студент – зарегистрированный, но имеющий ограниченный доступ, администратор, который занимается наполнением сайта, может узнавать информацию об успеваемости студентов.

Схема базы данных проекта представлена на рисунке 2.

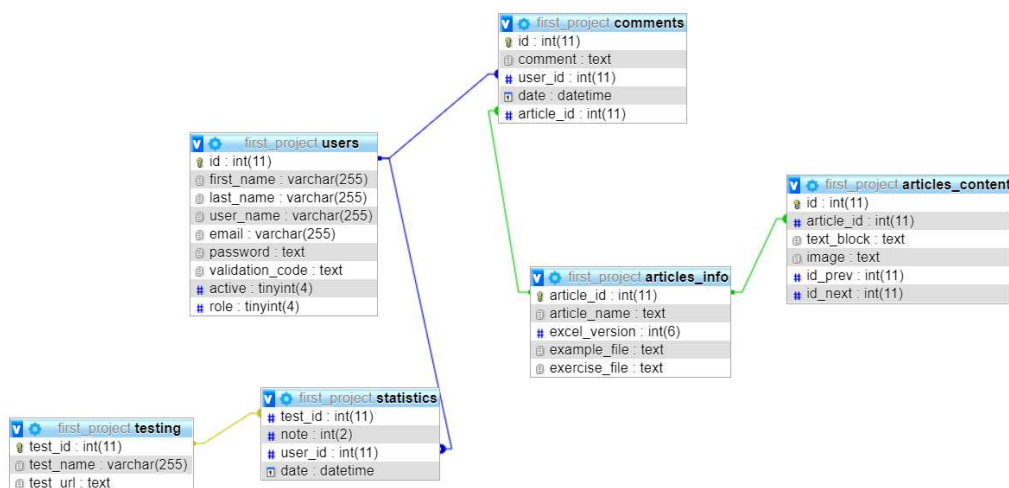


Рисунок 2 – Схема базы данных проекта

Разработанная база данных отвечает всем требованиям и обеспечивает функционирование СДО.

Студ. А. С. Леонова, Л. С. Юхновец
Науч. рук. ст. преп. Т.П. Брусенцова
(кафедра информатики и веб-дизайна, БГТУ)

ОСНОВНЫЕ ЭЛЕМЕНТЫ ДИЗАЙНА УЧАСТВУЮЩИЕ В СОЗДАНИИ КОМПОЗИЦИИ

Создание красивого дизайна – это больше, чем вдохновение или отличная идея, это понимание основ предмета. Существует несколько основных элементов, которые каждый дизайнер должен знать перед началом любого проекта. Прежде чем решать сложные дизайнерские задачи, необходимо понять основные компоненты дизайна. Даже любители в данной области, ведущие личные блоги, и любой, кто намерен заработать на этом начинании, должны их знать: линия, цвет, фигура, пространство, текстура, типографика, размер, иерархия, баланс, гармония. Цель данной работы: детально рассмотреть и проанализировать на эффективность пользовательского интерфейса элементы дизайна при создании веб-композиции.

Линия. Основной элемент дизайна – это линия. На рисунке линия – это штрих пера или карандаша, но в графическом дизайне это любые две соединенные точки. Линии полезны для разделения пространства и привлечения внимания к определенному месту. Например, журнал использует линии для разделения содержимого, заголовков и боковых панелей. С помощью линий задаются контуры или очертания. Линии могут быть кривые, горизонтальные, вертикальные, волнистые, параллельные, пунктирные

Посредством линий также можно передавать различные идеи. Прямые линии говорят о порядке и опрятности, волнистые создают ощущение движения, а зигзаги – напряжения или возбуждения.

Цвет. Цвет является одним из наиболее очевидных элементов дизайна, как для пользователя, так и для дизайнера. Он может быть изолирован в качестве фона или применяться к другим элементам, таким как линии, фигуры, текстуры или типографика. Каждый цвет говорит что-то свое, и их комбинации могут изменить это впечатление.

С помощью цвета можно привлечь внимание к определенной части сайта. Существуют основные цвета (красный, желтый, синий), вторичные цвета (оранжевый, зеленый, фиолетовый), получаемые путем смешения основных цветов, промежуточные (основной + вторичный) и дополнительные цвета (противоположные цвета на цветовом круге).

Фигуры. Формы, геометрические или органические, добавляют интерес. Формы определяются границами, такими как линии или цвет, и они часто используются для выделения части страницы. Все в ко-

нечном счете является формой, поэтому вы всегда должны думать о том, как различные элементы вашего дизайна создают фигуры и как эти фигуры взаимодействуют.

Когда линия пересекает себя или другую линию, то образуется фигура. Существуют геометрические фигуры (круг, квадрат, прямоугольник, треугольник) и органические фигуры (очертания листа, ракушки, цветка и так далее).

Пространство. Очень часто хочется разместить как можно больше информации на ограниченном пространстве, чтобы донести до конечного пользователя максимальное количество деталей. В результате получается каша из текста и картинок, лишенная смысловой нагрузки для постороннего человека. Чтобы привлечь внимание к информации, ее нужно выделить из общей массы. Можно придумать разные варианты – яркие цвета, крупный шрифт, особые графические элементы. Но чтобы все это сработало, вокруг каждого блока информации нужно оставлять свободное пространство. Чем больше воздуха вокруг элемента, тем выше его значимость для пользователя. Также увеличивая или уменьшая пространство между элементами, можно влиять на логическую связь между ними.

Текстура. Нелогично думать о текстуре, когда к нему никогда не прикоснуться. Тем не менее, веб-сайты и графический дизайн полагаются на внешний вид и фактуру текстур на экране. Текстуры могут создавать более трехмерный вид на этой двухмерной поверхности. Это также помогает построить захватывающий мир.

Текстура передает визуальный или тактильный вид поверхности. Текстура может быть гладкой, шероховатой, мягкой. Различают реалистичные текстуры, такие как дерево, бумага, тканевые текстуры, текстуры кожи и так далее.

Типографика. Типографика является одной из самых важных составляющих дизайна. Текст может многое сказать (в буквальном смысле), а само оформление текста в дизайне способно сказать еще больше. Любой шрифт сам по себе является законченным объектом дизайна. Хороший шрифт – результат кропотливой работы опытных дизайнеров.

В профессиональной среде бытует мнение, что разработка шрифтов – это верх дизайнерского ремесла. Человек, знающий, что такое антиква, гротеск, кернинг, лигатура, капитель, интерлиньяж и многое другое из мира шрифтов и типографики обречен быть внимательным к деталям.

Размер. Масштаб играет в дизайне огромную роль. В самом базовом смысле, масштаб – это конкретный размер отдельных элементов. Масштаб помогает нам правильно толковать дизайны и изображения. Не обязательно всегда руководствоваться реализмом при вы-

боре масштаба. Вы можете создавать невероятно большие или очень маленькие изображения, чтобы поразить зрителей или подчеркнуть, что в дизайне самое важное, а что – второстепенное.

Каждый дизайнерский элемент, участвующий в композиции, обладает определенным размером, или массой по отношению к другим элементам и ко всему дизайну в целом. Регулируя размер, можно обеспечить внимание, передать настроение, создать необходимый контраст, выделить информацию.

Иерархия. Иерархия в дизайне – это тоже самое, что иерархия в обществе: идея одна и та же. Сверху иерархического списка располагаются самые важные вещи. В то время как один элемент должен быть доминирующим, вы можете создать иерархию доминирования, чтобы направлять людей через ваш дизайн. Создавая визуальную иерархию в своем дизайне, вы позволяете сканировать вашу страницу и сообщать об относительной важности различных частей целого. Основные принципы проектирования помогают нам создать иерархию и помогают нам перевести взгляд с наиболее важных на наименее важные элементы.

Баланс. Баланс – это важная вещь во многих отношениях, и проекты – не исключение. Рассмотрим два общих типа баланса.

Симметричный баланс создает симметрию в проекте. Располагая важные элементы справа налево, сверху вниз, вы создаете правильное чувство баланса у пользователя.

Асимметричный баланс достигается пропорциональным рассеиванием и масштабированием элементов. Хорошим способом создания асимметричного баланса является представление, что каждый элемент имеет реальный вес. Маленькие объекты весят меньше, большие наоборот. Также можно представлять, что на вес влияет четкость текстуры объекта (более четкие весят больше). Таким образом, располагая элементы, чтобы они не перевешивали друг друга.

Гармония. Хорошая и эффективная композиция достигается, когда каждый элемент дополняет какой-либо другой, создавая общую картину. Распространенной ошибкой в композиции является выбор не подходящих друг другу изображений. Поэтому используя несколько изображений в композиции, стремитесь к тому, чтобы они сочетались.

Гармония – это «главная цель графического дизайна», по словам Алекса Уайта, автора «Элементов графического дизайна». Гармония – это то, что вы получаете, когда все части работают вместе.

На рисунке 1 представлен дизайн-макет сайта, использующего такие принципы как цвет, текстура, фигура, баланс, гармония (разработчик – студ. Анна Козловская).



Рисунок 1 – Дизайн-макет сайта GreenHome

Существуют сотни правил и намного больше принципов, когда речь идет о создании качественного дизайна в сети и за ее пределами, но это основные элементы, которые обеспечат привлекательность, профессионализм и совершенство всего, что вы создаете.

ЛИТЕРАТУРА

1. Элементы и принципы дизайна[Электронный ресурс] – Режим доступа: <https://sveres.ru/articles/kompozitsiya/osnovy-kompozitsii-v-dizayne.html>. – Дата доступа: 12.04.2019 г.

2. The 5 rules of design composition and layout[Электронныйресурс] –

Режим доступа: <https://99designs.com/blog/tips/design-composition-and-layout/>
<https://www.osp.ru/pcworld/2008/06/5371423/>. – Дата доступа: 14.04.2019 г.

3. Элементы и принципы дизайна. [Электронный ресурс] – Режим доступа <https://medium.com/основы-визуального-дизайна/элементы-и-принципы-дизайна>. – Дата доступа: 17.04.2019 г.

УДК 004.5:004.738.1

Студ. Е. В. Гришаева
Науч. рук. ст. преп. Т. В. Кишкурно
(кафедра информатики и веб-дизайна, БГТУ)

ИСПОЛЬЗОВАНИЕ ПРИНЦИПОВ ЮЗАБИЛИТИ В СОВРЕМЕННЫХ ВЕБ-САЙТАХ

В современном мире понимание основ юзабилити является необходимостью. Большинство посетителей сайта не совершают действий, которых от них ждут владельцы ресурсов. Люди не заказывают товар, не регистрируются, не подписываются на обновления и т.д. В

чем причина такого поведения пользователей? Причина проста – ресурс оказывается неудобным для пользователя [1].

Юзабилити (от английского usability – «удобство использования») – это качественная оценка простоты и комфорта работы с сайтом. Пользователь должен без труда находить нужную информацию, не теряясь в функционале и многочисленных страницах, и при этом получать эстетическое удовольствие от работы с ресурсом. Цель данной работы – проанализировать основные принципы юзабилити, которые позволят улучшить конверсию и посещаемость сайта.

Почему важно соблюдать принципы юзабилити сайтов?

1. Конверсия. Правильно продуманная навигация подтолкнет пользователя выполнить целевое действие: позвонить в компанию, скачать прайс-лист, заказать товар, воспользоваться услугой и т.д. Важно грамотно выстроить повествование и структуру, ведь если посетитель задумается хотя бы на несколько секунд или не сразу поймет, что делать дальше – можно его потерять. К примеру, если, попав на главную страницу, он с первых же строк не узнает о сфере работы компании, не сориентируется, где находится и что делать дальше – он без сожалений закроет вкладку. Если он не найдет развернутых ответов на интересующие вопросы в виде статьи, видео, графики – произойдет то же самое.

2. Посещаемость. Пользователь вернется на сайт только в том случае, если ему понравится с ним работать. А потому ресурс должен быть удобным и понятным со всех позиций: текст, навигация, интерфейс, последовательность действий и т.д. Благодаря обилию выбора современному потребителю не приходится долго разбираться в тонкостях работы с сайтом: сложно здесь – он перейдет к конкурентам, у которых понятнее и проще [2].

Основные принципы юзабилити, которые должны использоваться в современных сайтах:

– Лучший цвет для ссылок это синий. Разработка юзабилити сайта базируется на главном правиле: «чем проще – тем лучше». Чем более привычную структуру и интерфейс видит пользователь – тем легче ему сориентироваться и тем меньше времени он потратит, чтобы найти нужный раздел, т.к. будет чувствовать себя более уверенно. Цвет ссылок браузера по умолчанию это синий, так что люди ожидают увидеть именно этот цвет [3].

– Дизайн – залог доверия к сайту. Грамотно продуманный «внешний вид» сайта – залог доверия со стороны потребителя, попавшего к вам впервые. Простота и доступность использования, цветовые сочетания, стиль, подходящие вашей целевой аудитории кар-

тинки, качество контента, отсутствие ошибок – все это влияет на расположение к ресурсу [2].

– Логичный структурированный контент. Один из главных факторов юзабилити – удобное восприятие текстов. Структура, оформление, информативность, логическая разбивка, оптимизация, отсутствие ошибок – все эти параметры важно учитывать при написании статьи. Если это длинный текст-руководство – сделайте оглавление в виде ссылок, чтобы пользователь мог быстро найти нужный раздел [2].

– Свободное пространство улучшает понимание. Сайт можно сравнить с жилой комнатой: если каждый квадратный метр чем-то занят – человек будет чувствовать себя неудобно. Аналогично и здесь: если страница будет переполнена полезными блоками, разделами, ремарками и прочей информацией – посетитель потеряется и не поймет, куда устремить внимание в первую очередь. Современный дизайн склонен к минимализму: на странице – только самое важное [3].

– Важная информация – в левом углу. Посетители просматривают web страницу по F-образному шаблону: внимание концентрируется в верхнем левом углу и ограничивается первыми несколькими абзацами текста в убывающем порядке. Потому при распределении важной информации использование такой схему увеличит вероятность того, что посетитель быстро найдет нужные данные, увидит самое важное, бегло пробежавшись по разделу взглядом [2].

– Правило 7-ми. Известно, что способности человеческого мозга неограниченные. Человек способен держать в кратковременной памяти не более 5-9 сущностей. Поэтому не следует размещать в навигации больше 7 пунктов [1].

– Пользователи концентрируются на лицах. Люди инстинктивно замечают других людей сразу же, как они приходят в поле зрения. На веб-страницах мы обращаем основное внимание на лица и глаза, что дает маркетологам хороший инструмент для привлечения внимания. Но наше внимание к лицам и глазам людей это только начало, оказывается, на самом деле мы даже начинаем смотреть в том направлении, куда смотрит человек на картинке [4].

– Общение с пользователем. Успешная регистрация, принятый заказ, несуществующая страница – человек должен быть всегда в курсе происходящего. Причем во всех подробностях. Если он зарегистрирован в системе – расскажите, что делать дальше, если заказ принят: укажите номер, время ожидания звонка, контактные данные; если попал на страницу 404 – как найти нужную информацию или куда вернуться [2].

– Отмена действия. У человека должна быть возможность передумать. К примеру, социальная сеть ВКонтакте позволяет восстано-

вить страницу после ее удаления. Некоторые интернет-магазины – изменить или удалить заказ после его оформления, откорректировать способ доставки и оплаты. Все это – составные части юзабилити веб-сайта [2].

– Ненавязчивое предложение помощи. Не нужно настаивать на общении. Большие выскакивающие окна с предложением помочь в выборе, цветная полоска с номером телефон, занимающая треть экрана – подобные элементы не только смотрятся навязчиво, отталкивают посетителя, но и закрывают часть функционала [2].

– Разговор на понятном языке. Не делайте упор на специализированные термины – говорите проще, чтобы пользователю не пришлось тратить время на толкование непонятного слова. Если без узких понятий, пояснений не обойтись – разрабатывайте всплывающие окна. Это же касается заполнения форм: если вы диктуете определенные условия в виде длины имени, пароля, формата телефона, допустимой географии – укажите это в виде подсказок [2].

– Все возможности – на виду. Удачное решение – когда пользователь видит все разделы, в которые может перейти, т.к. в любой момент может быстро сориентироваться, где находится и как вернуться обратно. Если на сайте планируется много контента – резонно предусмотреть горизонтальное верхнее, вертикальное меню (и то, и другое – возможно, с подразделами), а также «подвал». Если речь идет о минималистичной «визитке» или landing page – достаточно только верхнего горизонтального меню [2].

– Если категорий очень много – добавьте пункт «Карта сайта», который позволит пользователю сориентироваться, какие разделы содержит ресурс и как ему туда попасть. Не стоит пренебрегать «хлебными крошками» – навигационной цепочкой в верхней части блока контента, которая подсказывает, по какой схеме человек попал на страницу.

– Форма регистрации и заказа: чем лаконичнее – тем лучше. Чем больше полей для заполнения – тем меньше пользователю хочется тратить на это время. Причем компоновка полей, надписей, подбор шрифтов могут быть идеальными, а вот длина – отталкивающей. Современные основы юзабилити заставляют веб-разработчиков сокращать длину этих форм до минимума, нещадно выкидывая вопросы об образовании, уровне дохода, профессии и т.д. [2].

– Заметное длинное поле поиска. Соответствующая кнопка – лучший помощник посетителя. Особенно если речь идет о крупном ресурсе. Строку поиска пользователь должен обнаружить без труда. Чтобы при первом знакомстве он не тратил на обнаружение соответствующей строки даже несколько секунд [5].

– Еще один важный момент – длина поля поиска. Согласно исследованию Якоба Нильсена, в среднем туда вмещается 18 символов, а оптимальная длина – 27 символов. В 90% случаев сюда поместится даже длинный запрос, чтобы пользователь его полностью увидел и исправил при надобности. Для установки нужной ширины используйте относительную единицу «em», равную размеру используемого шрифта.

После того как все принципы юзабилити учтены, переходят к тестированию, которое проводится постоянно, т.к. любая ошибка может оттолкнуть потенциального клиента.

Важно применять правила юзабилити не в самый последний момент: чем раньше разработчик начнет планировать структуру и внешний вид будущего ресурса – тем больше аспектов продумает, более качественный результат получит и сможет без спешки внести все коррективы. Предварительно следует проанализировать сайты конкурентов, выявив их слабые и сильные стороны, а также оценить предыдущий ресурс при его наличии, обнаружив моменты, требующие улучшения [2].

ЛИТЕРАТУРА

1. Хабрахабр [Электронный ресурс]. – 2011. – Режим доступа: <https://habr.com/ru/company/webprofessionals/blog/132564/> – Дата доступа: 4.04.2019.
2. Ашматов и партнеры [Электронный ресурс]. – 2017. – Режим доступа: <https://www.ashmanov.com/education/articles/yuzabiliti/> – Дата доступа: 5.04.2019.
3. Хабрахабр [Электронный ресурс]. – 2009. – Режим доступа: <https://habr.com/ru/post/70787/> – Дата доступа: 4.04.2019.
4. WebforMyself [Электронный ресурс]. – 2009. – Режим доступа: <https://webformyself.com/10-poleznyx-yuzabiliti-principov-i-naxodok/> – Дата доступа: 5.04.2019.
5. WebforMyself [Электронный ресурс]. – 2017. – Режим доступа: <https://webformyself.com/principy-i-pravila-yuzabiliti-sajta-dlya-masterov-veb-dizajna/> – Дата доступа: 5.04.2019.

КАК С ПОМОЩЬЮ ДИЗАЙНА ТЕКСТА, ЦВЕТА И ИЗОБРАЖЕНИЯ ПОВЫСИТЬ ЮЗАБИЛИТИ ИНТЕРФЕЙСА

Задача дизайнеров интерфейсов состоит в том, чтобы найти наилучшее представление, доносящее информацию о поведении программы, в проектировании которой они принимают участие. Они должны стремиться сделать эстетически привлекательный и удобный интерфейс, который мог бы представить поведение и информацию в понятном и полезном виде, который поддерживает маркетинговые цели организации и эмоциональные цели пользователей. Сделать это можно с помощью дизайна свойств элементов интерфейса, таких как текст, цвет и изображение. Они являются основными ключевыми моментами, которые помогают повысить юзабилити интерфейса. Цель данной работы – проанализировать основные компоненты интерфейса, которые позволяют повысить юзабилити сайта.

Дизайн текста. Учитывая размер шрифта для основного текста, можно установить иерархию размеров для различных частей текста и соблюдать эту иерархию везде. В большинстве интерфейсов используется избыточное количество размеров шрифтов. Нередко можно обнаружить, что каждое значение от 10 до 24 пикселей где-то использовалось в UI. Для выбора размера можно использовать соотношения начиная с 1брх, например 4: 5 («большая треть(терция)»), 2: 3 («идеальная пятая часть (чистая квинта)») или, возможно, «золотое сечение», 1: 1,618 которое часто называют «модульной размерностью». В случае, если этот подход не очень удобен, можно воспользоваться таблицей размерности, которая начиная с 1брхи увеличивается на 2 рх. В проекте должны использоваться хорошие шрифты. Если вы не можете доверять своему вкусу, то можно использовать системный стек шрифтов. Пользователи привыкли к ним, и это будет смотреться аккуратно. Пользуясь поиском, игнорируйте шрифты с вариациями толщины меньше 5, полагайтесь на раздел популярные и помните, что существуют шрифты, разработанные для текста и шрифты для заголовков. Хорошим способом является заимствование шрифтов с понравившихся вам сайтов. Важно помнить также и о длине линии текста, она не должна быть слишком длинной, желательно в пределах диапазона 45-75 символов в строке. Это работает так же в случае расширенного содержания. Выравнивайте текстовые элементы по базовой, а не по центральной линии, так как в противном случае это приведет к неравновесной композиции.

Не стоит забывать о высоте линии текста, чтобы избежать потери внимания при переходе на другую строку или повторного прочтения одной строки. Высота линии около 1,5 является хорошей отправной точкой с точки зрения читаемости, однако важно учитывать длину строки и размер шрифта. Это означает, что высота строки и ширина абзаца должны быть пропорциональны – узкий контент может использовать более короткую высоту строки, например, 1,5, а для широкого контента может потребоваться высота строки до 2. Для большого текста заголовка вам может не понадобиться дополнительный интервал между строками, а высота строки в 1 отлично подходит.

Не выделяйте каждую ссылку цветом. Для сохранения вида ссылки как активного элемента подберите гамму цветов и изменяйте яркость или насыщенность, если ссылки идут подряд иерархической моделью. Так же можно добавить подчеркивание при наведении курсора. Важным элементом является выравнивание текста. Стоит учитывать язык, так как например для английского и русского принято выравнивание по левому краю. Выравнивание по центру не подходит для длинного текста, однако отлично подходит для двухстрочных названий. Выравнивание по правому краю чаще используется при заполнении таблиц числовыми значениями, так как так их проще сравнить. Если вы используете выравнивание флагом, не забывайте включить переносы, чтобы избежать пробелов между буквами и словами.

Работа с цветом. Наиболее распространенными схемами для представления цвета в интернете являются Hex и RGB, но они не являются лучшими, т. к. выглядят по-разному в коде. Исправить это, представление цвета можно с помощью схемы HSL с использованием атрибутов человеческого глаза, интуитивно воспринимающего оттенок, насыщенность и яркость. Зачастую вам нужно больше цвета, чем вы думаете, поэтому не стоит слепо доверять генераторам цветовой палитры, которые при введении основного цвета, выдают 5 цветов, которые вы должны использовать для создания своего интерфейса. В результате интерфейс выглядит не очень привлекательно. Нужно сначала определить основные оттенки и изменяя яркость, насыщенность или оттенок собирать палитру с учетом разницы этих параметров для кнопок, основного текста или блоков. Далее необходимо определить цвета акцента, которые будут сообщать об ошибке (красный), верном действии (зеленый) или предупреждении (желтый).

При использовании серого цвета помните, что можно сделать так, чтобы он выглядел как холодный или теплый путем добавления соответственно синих и желтых цветов.

При подборе оттенков не позволяйте яркости убить насыщенность. Помните, что можно визуально изменить яркость цвета изменяя лишь его оттенок, так как по человеческой природе некоторые цвета, например, желтый воспринимаются ярче. Хотя это отличный способ изменить яркость цвета, не влияя на его интенсивность, он лучше всего работает в небольших дозах. Не поворачивайте оттенок более чем на 20-30°, иначе он будет выглядеть как совершенно другой цвет, а не просто светлее или темнее. Чтобы веб-контент был доступен необходимо, чтобы обычный текст (до ~18px) имел коэффициент контрастности не менее 4,5:1, а больший текст имел коэффициент контрастности не менее 3:1. При использовании белого текста на цветном фоне соотношение контрастности должно быть 4,5:1. Такое соотношение может создать проблемы с иерархией, так как темные цветные фоны привлекут внимание пользователя. Эту проблему можно решить, перевернув контраст. Вместо использования светлого текста на темном фоне используйте темный текст на светлом фоне.

Никогда не полагайтесь только на цвет, иначе пользователям с дальтонизмом будет трудно интерпретировать ваш пользовательский интерфейс. Необходимо передавать информацию и каким-то другим способом, например, добавив значки. Если мы работаем с графиком, где каждая линия тренда имеет другой цвет, необходимо полагаться на контраст, а не использовать разные цвета. Дальтоникам гораздо легче показать разницу между светом и темнотой, чем разницу между двумя различными цветами.

Работа с изображениями. Плохие фотографии разрушат дизайн, даже если все остальное в нем выглядит великолепно. Поэтому необходимо использовать качественные фотографии. Если при размещении текста на большом изображении, независимо от того, какой цвет вы выбрали для текста, его все-таки трудно прочитать, то решить эту проблему можно, уменьшив динамику в изображении, чтобы сделать контраст между текстом и фоном более последовательным. Для этого можно добавить наложение, тем самым затемнив фон для белого текста, понизить контрастность изображения для использования темного текста или же раскрасить изображение, например, в фирменный цвет компании добавив на него полупрозрачный цветной слой. Хорошим решением в данном случае будет добавление тени на текст.

Всем известно, что масштабирование растровых изображений до размеров, превышающих их первоначальный размер, является плохой идеей – они сразу же становятся «нечеткими». А качество векторных изображений не ухудшатся, когда вы увеличите их размер. Значки, которые были нарисованы в 16-24px, никогда не будут выглядеть

очень профессионально, если вы увеличите их до 3-х или 4-х их предполагаемых размеров. Им не хватает деталей, и они всегда получаются непропорционально «коренастыми». Если маленькие значки – это все, что у вас есть, то попробуйте заключить их в другую форму и придать форме цвет фона. Это позволит вам увеличить предполагаемый размер, сохранив фактический значок.

Не уменьшайте скриншоты, слишком сильно, например до 70%, так как вы получите изображение, которое пытается втиснуть слишком много деталей в слишком малое место. В результате пользователи смогут понять, что говорится в тексте. Если вам необходимо сделать подобный скриншот, сделайте снимок экрана с меньшим размером экрана (например, макет планшета) и оставьте для него много места, чтобы вам не пришлось его сжимать, либо подумайте о том, чтобы сделать только частичный снимок экрана, чтобы вы могли отображать его в меньшем пространстве без необходимости его масштабирования. Если вам действительно нужно поместить скриншот всего приложения в узкое пространство, попробуйте нарисовать упрощенную версию пользовательского интерфейса с удаленными деталями и замененным небольшим текстом на простые линии. Так же, прерывистыми и нечеткими при масштабировании выглядят и значки, предназначенные для использования в больших размерах. Лучший подход – перерисовать супер упрощенную версию логотипа в целевом размере.

Если вы зависите от загруженного пользователем контента, вы не сможете тонко настроить контраст, цвет или кадр. Поэтому разместите их изображения внутри фиксированных контейнеров, вырезая все, что не подходит. Если пользователь предоставляет изображение с цветом фона, похожим на фон в пользовательском интерфейсе, изображение и фон могут слиться вместе, в результате чего изображение теряет свою форму. Вместо того, чтобы пытаться решить эту проблему с помощью границы, попробуйте использовать тонкую внутреннюю тень рамки. Границы часто сталкиваются с цветами на изображении, в то время как большинство людей едва ли даже поймут, что тень есть. Если вам не нравится небольшая «вставка», которую вы получаете от использования тени рамки, полупрозрачная внутренняя граница тоже отлично работает.

Используя все эти правила можно значительно улучшить юзабилити интерфейса.

ЛИТЕРАТУРА

1. Adam Wathan and Steve Schoger, *RefactoringUI*: учебное пособие / Adam Wathan and Steve Schoger. – 2018. – 252 с.

Студ. Н. В. Левченко

Науч. рук. ст. преп. Т. В. Кишкурно

(кафедра информационных систем и технологий, БГТУ)

ИСПОЛЬЗОВАНИЕ ДИЗАЙН-СИСТЕМЫ ДЛЯ УСКОРЕНИЯ РАЗРАБОТКИ И ПОДДЕРЖКИ

Одним из наиболее распространенных заблуждений о дизайн-системе является чрезмерно узкое понимание термина «дизайн-система», когда за неё принимают какой-то один из составляющих элементов, например, визуальный язык или дизайнерские шаблоны.

Способность к передаче опыта и знаний была одним из толчков к развитию человечества, однако, формализованной системы для этого не существовало в течение очень долгого времени. Всё происходило довольно стихийно, буквально из уст в уста.

Прекрасным примером подлинной реформы в систематизации знания можно считать изобретение в средние века Гвидо Аретинским системы нотного письма, которая впоследствии эволюционировала в то, чем мы пользуемся и сейчас. Она позволила существенно упростить процесс обучения певчих, и сократить его с 10 лет до 2! [1]

Чем сложнее и многограннее система, тем значимее точность передачи опыта. Поэтому, громадный технологический рывок XX века потребовал усовершенствовать то, как мы формализуем знание и контролируем результат.

В середине 50-х с активным ростом коммерческого графического дизайна для корпораций возникает понятие визуальной идентификации бренда (слово «дизайн-система» появляется примерно в то же время).

Классическим (и наиболее близким к нашей теме) примером является исключительно подробная инструкция для метрополитена Нью-Йорка [2], где досконально объясняются все детали создания графики, принципы навигации и т.п. (рисунок 1).

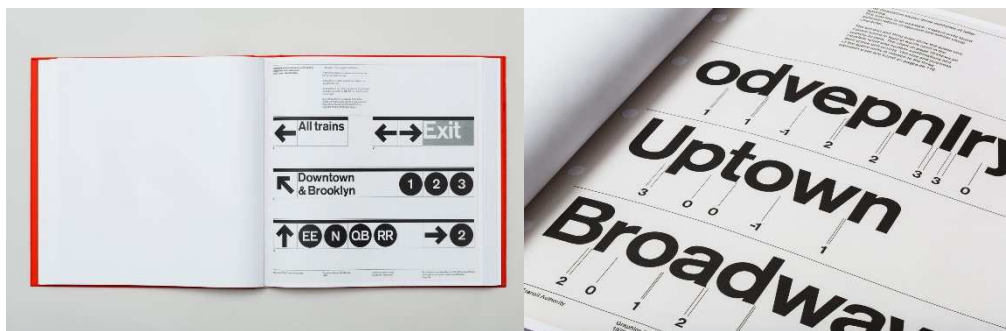


Рисунок 1– Страницы из гайдлайна метро Нью-Йорка

Затем, уже в наши годы, точнее в последние десять лет, экспоненциально быстро выросло количество digital-стартапов, которым требовались быстро масштабируемые решения, основой которым послужил новый взгляд на понятие дизайн-систем. Последней вехой развития в понимании термина стала концепция “Атомарный дизайн” (Термин ввел Брэд Фрост) [1].

Ее суть заключается в атомарности визуального языка продукта как реального объекта. Скажем, у нас есть страница социальной сети Facebook. У многих Facebook ассоциируется с пальцами вверх. Визуально этот жест будет являться “атомом” системы, который входит в состав “молекулы” подвала поста на странице, а “молекула” в свою очередь входит в состав “живого организма” под названием “пост на странице”, который в совокупности с другими “организмами” выступает в роли шаблона, из которых и формируются страницы в данной социальной сети (рисунок 2).

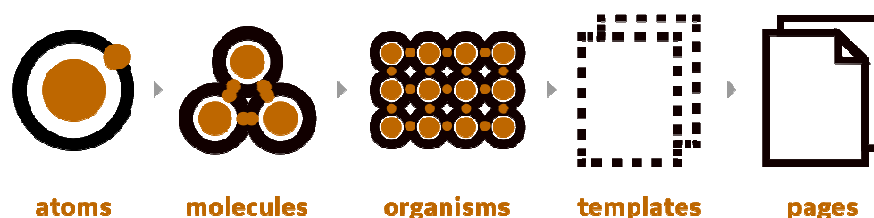


Рисунок 2– Концепция атомарного дизайна

Если говорить терминами, то дизайн система – это визуальный язык продукта, подкрепленный технически реализуемыми методами, содержащий в себе визуальную и техническую библиотеки элементов. Суть каждой дизайн системы – ускорить процесс разработки продукта, упростить поддержку системы, быстро адаптироваться к изменяющимся условиям, создать прочный мост между доверием пользователя и продуктом в кратчайшие сроки.

Полноценная дизайн-система состоит из библиотеки графических элементов, библиотеки элементов в коде, шаблонов для дизайнерских инструментов, команды [3].

Библиотека графических элементов содержит в себе все визуальные атомы, молекулы и организмы системы. Библиотека элементов в коде, соответственно, код для каждого из атомов системы. Шаблоны необходимы дизайнеру для скорейшего построения интерфейса. Команда – группа людей, которая занимается разработкой и поддержкой дизайн системы, исправлением ошибок, ее улучшением.

Многие путают понятие дизайн-системы и гайдлайна. Гайдлайн является лишь описанием визуального стиля чего-нибудь, какого-то конкретного элемента. Дизайн-система – более широкое понятие, скорее источник для формирования гайдлайнов, она содержит их внутри себя.

Дизайн-системы используются и разрабатываются большинством крупных и не очень компаний, потому что дизайн-системы формируются за счет философии компании и помогают продукту достигать цели, поставленные компанией. Интерфейсы, реализованные в контексте дизайн-системы продукта, помогают установить твердую связь между пользователем и продуктом, иными сервисами компании, завлечь его к себе надолго. Наряду с осознанным вещанием философии, хороший продукт будет казаться еще лучше, нежели конкурент, который не использует дизайн-систему.

Одним из примеров эффективности использования дизайн-системы по временному фактору может послужить заранее написанные фреймворки и тулкиты для CSS и JS. Например, Tachyons и Bootstrap. Web-проекты гораздо быстрее реализуются с их помощью, нежели с использованием бессистемного подхода.

С точки зрения визуальной привлекательности – тут все очевидно. Системный подход в дизайне гораздо четче прослеживается и пользователи относятся к нему гораздо теплее, нежели к интерфейсам, построенным по неупорядоченному принципу – разными падингами, размерами и гарнитурами шрифта, и т.д.

Подводя итоги, следует отметить неопровержимую пользу, которую дизайн-системы приносят всем вокруг – от стартапов до транснациональных корпораций и простых пользователей. Атомарный подход к построению системы, сливающей в себе аспекты визуального языка и языка программирования, послужил началом новой эпохи в сфере построения digital-продуктов, нового понимания термина “дизайн-система”. Примерами открытых дизайн-систем могут послужить разработки именитых российских компаний – Paradigm (Mail.ru) [4], Дизайн Альфа-Банка (Альфа-банк) [5].

ЛИТЕРАТУРА

1. Хабр [Электронный ресурс]. – 2019. – Режим доступа: <https://habr.com/ru/company/mailru/blog/351726/> – Дата доступа: 5.04.2019
2. Standarts Manual [Электронный ресурс]. – 2019. – Режим доступа: <https://standardsmanual.com/products/nyctacompactedition/> – Дата доступа: 5.04.2019
3. DesignBetter [Электронный ресурс]. – 2019. – Режим доступа: <https://www.designbetter.co/design-systems-handbook/> – Дата доступа: 5.04.2019
4. Paradigm [Электронный ресурс]. – 2019. – Режим доступа: <https://design.mail.ru/> – Дата доступа: 5.04.2019
5. Дизайн Альфа-Банка [Электронный ресурс]. – 2019. – Режим доступа: <https://design.alfabank.ru/> – Дата доступа: 5.04.2019

Студ. Е. И. Жаденова
Науч. рук. ст. преп. Т. В. Кишкурно
(кафедра информатики и веб-дизайна, БГТУ)

МОТИОН-ДИЗАЙН В ИНТЕРФЕЙСАХ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ ДЛЯ УЛУЧШЕНИЯ ЮЗАБИЛИТИ

Анимационная графика, она же motion-графика – это визуальное оформление, которое оживляет статическое изображение. Интернет, медиа и реклама, телевидение, кино, мобильные приложения, видеоигры – без нее не обходится ни одна из этих сфер. Анимационная графика использует сразу три канала информации: изображение, текст и звук, и зритель лучше усваивает данные. А в случае мобильных интерфейсов зачастую используется ещё и виброотклик.

«Motion-design – это не ключевой аспект пользовательского опыта». Согласно этому утверждению, всё держится исключительно на UX части, а движение – что-то отдельное, не играющее значительной роли в юзабилити. Но это совершенно не так, потому что именно движение в конечном итоге будет иметь значение, когда пользователь в очередной раз уйдёт со страницы, не завершит заказ, удалит приложение, потому что иногда дело может быть всего в одном небольшом элементе и его анимации [1].

«Motion-design – это всего лишь сливки». Второе утверждение заключается в том, что движение может быть добавлено сверху на любые элементы и в самый последний момент. Но проблема заключается в том, что продумывание анимации взаимодействий должно проектироваться на том же этапе, когда отрисовываются экраны, переходы между ними и определяется логика работы приложения, потому что иначе можно добиться совсем неправильного эффекта.

«Motion-design – анимация». В данном случае термин «анимация» включает в себя вещи, которые едва ли соотносятся с самой сутью UX-дизайна. Из-за искажённого понимания термина и вытекает первое утверждение. «Motion-design применим в специфическом контексте». Согласно этому утверждению, некоторые виды анимаций работают лучше в конкретном контексте. Это значит, что motion-design интерфейсов должен существовать лишь в рамках удачных решений, основанных на предыдущем опыте, но это совсем не так. Движение может быть совершенно разным, именно оно и помогает раскрыть продукт с новой стороны [2]. Цель работы – проанализировать возможности анимационной графики в интерфейсах мобильных приложений для улучшения юзабилити.

Движение поддерживает юзабилити четырьмя способами.

Ожидание. Ожидания бывают разными, они определяют то, как пользователи воспринимают объекты и то, как объекты себя ведут на самом деле. Другими словами, дизайнерам хотелось бы минимизировать разрыв между тем, что пользователь ожидает, и тем, какой опыт получает на самом деле.

Непрерывность. Непрерывность одновременно относится и к пути пользователя, и к последовательности пользовательского опыта. Непрерывность может рассматриваться как плавность опыта взаимодействия между интерфейсами, которая и составляет совокупный опыт.

Повествование. Повествование – это линейное развитие событий в пользовательском опыте, в результате которого появляется временная или пространственная система координат. Повествование можно представить себе как серию отдельных моментов и событий, которые связаны друг с другом с помощью пользовательского опыта.

Соотношение. Соотношение – это пространственное, временное и иерархическое представление между объектами интерфейса. Оно формирует понимание и способствует принятию решений пользователями [3].

Когда элементы изменяют своё состояние или позицию, продолжительность анимации должна быть достаточной, чтобы дать пользователям возможность заметить изменения, но в то же время не слишком долгой, чтобы ему не пришлось переживать и ощущать состояние «ожидания» [4].

В то время как принципы Диснея приводят к «более привлекательным анимациям», принципы анимации пользовательского интерфейса приводят к большему количеству полезного опыта.

Динамика. Поведение объекта должно соответствовать ожиданиям пользователя при возникновении временных процессов. Динамика создает и усиливает «естественность», присущую беспрепятственности пользовательского опыта, и создает ощущение непрерывности, когда объекты ведут себя так, как того ожидают пользователи.

Смещение и задержка – определяет связи объектов и иерархии при введении новых элементов и сцен. Значение этого принципа заключается в том, что он неосознанно настраивает пользователя на успех, «рассказывая» ему о природе объектов интерфейса. Даже до того, как пользователь опознает эти объекты, дизайнер уже сообщил ему через анимацию, что объекты каким-то образом отделены.

Родительство – создает пространственные и временные иерархические отношения при взаимодействии с несколькими объектами. Это мощный принцип, который «связывает» объекты в пользовательском ин-

терфейсе. Родительство функционирует лучше всего во взаимодействиях реального времени. Поскольку пользователь непосредственно управляет объектами интерфейса, дизайнер сообщает пользователю – через анимацию – как объекты связаны, и какая взаимосвязь между ними.

Трансформация – создает непрерывное состояние потока нарративов, когда изменяется служебная программа объекта. О UX преобразовании уже написано много. В некотором смысле это самый очевидный и видимый из принципов анимации. Трансформация наиболее заметна, во многом потому, что она выделяется. Преобразование осуществляет беспрепятственный переход пользователя через различные состояния UX.

Изменение значений – создает динамичную и непрерывную повествовательную связь, когда предмет значения изменяется. Объекты интерфейса, основанные на тексте, то есть цифры и текст, могут менять свое значение. Это одна из тех «неуловимых очевидных» концепций. Пользователи, понимая динамический характер своих данных, теперь могут влиять на изменение этих значений и могут стать агентами влияния. Когда значения статичны, связь с реальностью, лежащей в основе значений, уменьшается, а пользователи теряют свое влияние.

Накладывание маски – создает непрерывность в объекте или группе объектов интерфейса, когда функционал определяется тем, какая часть объекта или группы раскрыта или скрыта. Процесс накладывания маски можно воспринимать как связь между формой объекта и его функционалом. Благодаря временному использованию раскрывающих и скрывающих областей объекта функционал проходит непрерывно и плавно.

Оверлей – создает пространственные отношения нарратива и объекта в визуальной плоскости, когда объекты со слоями и местоположение взаимозависимы. Overlay позволяет дизайнерам использовать движение для передачи объектов, зависящих от местоположения, которые существуют за или перед другими в не 3D-пространстве и тем самым способствовать пространственной ориентации своих пользователей.

Клонирование – создает непрерывность, связь и повествование, когда появляются или исчезают новые объекты. Когда новые объекты создаются в текущих страницах, важно учитывать их внешний вид. Накладывание масок, клонирование и протяжность – три подхода, основанных на юзабилити, для создания сильных описаний.

Матирование – позволяет пользователям пространственно ориентироваться в отношении объектов или сцен, не в основной визуальной иерархии. Подобно принципу Masking, Матирование существует в качестве статического и временного явления.

Параллакс – создает пространственную иерархию в визуальной плоскости при скроллинге. Дизайнеры могут использовать смещение для выделения первоочередного содержимого от внешнего и дополнительного контента. Пользователь не только воспринимает интерфейсные объекты, как имеющую иерархию, превышающую ту, что определена в визуальном дизайне, но теперь эта иерархия может быть использована для того, чтобы пользователь понял природу пользовательского опыта, прежде чем сознательно разобрать дизайн/контент.

Многомерность – обеспечивает пространственную структуру повествования, когда новые объекты появляются и исчезают. Люди замечательно разбираются в использовании пространственных структур для навигации, как в реальном мире, так и в цифровом опыте. Предоставление референсов пространственного появления и исчезновения помогает укрепить ментальные модели там, где пользователи находятся в UX.

Наезд и масштаб – сохраняет непрерывность и пространственное повествование при навигации объектов интерфейса и пространств [2].

Пользователям невероятно нравится, когда у приложения или веб-сайта есть смелые и интересные решения в сфере дизайна. Motion-дизайн стал неотъемлемым элементом пользовательских интерфейсов. Он стал интуитивным способом подтверждения успешности действий пользователей, вспомогательным элементом для лучшей ориентации в интерфейсе, улучшает обучаемость и повышает возможность манипулирования вниманием. Плюс ко всему, движение, которое является практически незаметным, стало неотделимым элементом дизайна, ориентированного на пользователя.

ЛИТЕРАТУРА

1. Uxdesign.cc [Электронный ресурс] / Uxdesign.cc – 2019. – Режим доступа: <https://uxdesign.cc/adding-value-to-your-design-by-understanding-motion-and-its-tools-c06418e43a1b> – Дата доступа: 01.04.2019.

2. Telegraf.design [Электронный ресурс] / Telegraf.design – 2019. – Режим доступа: <https://telegraf.design/sozдание-yuzabiliti-s-animatsiej-manifest-ux-v-moushn-chast-1/> – Дата доступа: 06.04.2019.

3. Medium [Электронный ресурс] / Medium – 2019. – Режим доступа: <https://medium.com/ux-in-motion/motion-design-in-digital-products-a-white-paper-48da834261dd> – Дата доступа: 05.04.2019.

4. Medium [Электронный ресурс] / Medium – 2019. – Режим доступа: <https://medium.com/ux-in-motion/the-irony-of-usability-and-linear-motion-88f67f7efcf9> – Дата доступа: 01.04.2019.

Студ. В. О. Завадская
Науч. рук. ст. преп. Т. В. Кишкурно
(кафедра информатики и веб-дизайна, БГТУ)

КАК УЛУЧШИТЬ ВЕБ-ДОСТУПНОСТЬ С ПОМОЩЬЮ ВЫБОРА ПРАВИЛЬНОЙ ПАЛИТРЫ

Сочетание цветов имеет не последнее место в создании хорошего дизайна сайта. Художники используют теорию цвета уже сотни лет. Но история ее применения в бизнесе и маркетинге намного короче. Одна из первых современных трактовок теории цвета была написана около ста лет назад Альбертом Манселлом (Albert Munsell). Манселл придумал для описания цветовых свойств трехкомпонентную модель, в которой цвет выявляется с помощью трех понятий – значения /ценности (англ. value), оттенка (англ. hue) и насыщенности / яркости (англ. chroma)[3].

В последние несколько десятилетий, использование теории цвета пришло и в интернет-маркетинг. Веб-дизайнерам пришлось сильно потрудиться, чтобы описать понимание того, как человеческий мозг воспринимает цвета в цифровом контексте.

В настоящее время теория цвета становится все актуальнее в сфере интернет-маркетинга в связи с ростом значимости показателя отказов и других статистик взаимодействия. Когда посетитель за доли секунды решает, доверять сайту или нет, важна любая мелочь. И применение теории цвета на практике в данном случае может оказать решающее воздействие на сознание посетителя, заставив его остаться на странице.

При создании сайтов дизайнеры используют теорию цвета для улучшения эстетики, читабельности страниц, и, что более важно, для получения нужного психологического воздействия. Цель данной работы – провести подробный анализ основных правил для подбора палитры цветов, отвечающий требованиям юзабилити.

Основные правила для подбора палитры:

1. Системные цвета. Большинство систем закрепляют отдельный оттенок красного для ошибок, зеленого – для успешных операций, желтый – для предупреждений и синий – для информационных сообщений. Цвета отклика критически важны: они выводятся наверху страницы, сочетаясь с остальными ключевыми компонентами, и пользователи могут воспринимать их не очень дружелюбно. Без четкой структуры в системе такие цвета просто прописываются в коде – в это может привести к непостоянству и разброду цветов. На рисунке 1 представлен вариант выбора системных цветов.



Рисунок 1 – Системные цвета

2. Контраст. Возможность реализации доступного приемлемого контраста должна быть продумана в цветовой системе изначально.

Рекомендации по доступности веб-контента (WCAG), разработанные W3C, представляют формулу для расчёта контрастности между двумя цветами, с вычислением коэффициента контрастности.

Коэффициент контрастности варьируется в диапазоне от 1:1 (нет разницы) до 21:1 (максимально возможное значение). Их легко вычислить во многих бесплатных инструментах: Tanaguru, Contrasti плагин Stark Sketch.

WCAG точно определяет коэффициенты контрастности для читаемого текста:

- уровень AA: минимальный стандарт: у мелкого шрифта коэффициент контрастности должен быть 4.5:1 или выше. У крупного шрифта 3:1 или выше;

- уровень AAA: расширенный стандарт: у мелкого шрифта коэффициент контрастности должен быть 7:1 или выше. У крупного шрифта 4,5:1 или выше[2]. Наглядный пример контрастности цветов показан на рисунке 2.

Level AA minimum contrast	Level AAA minimum contrast
Small text (16px, 4.5:1)	Small text (16px, 7:1)
Large text (24px, 3:1)	Large text (24px, 4.5:1)

Рисунок 2 – Пример контрастности цветов

3. Необходимо находить приемлемые цвета, выбирая из спектра. Недостаток спецификаций WCAG – их жесткий предел: цветовые пары либо проходят проверку, либо нет. Дизайнерам и клиентам нужно больше, им нужен целый спектр подходящих цветов.

Выбор необходимого оттенка происходит быстрее, когда появляется спектр опций на выбор, когда линия прохождения или непрохождения пар объяснима и очевидна. Это переводит процесс из режима тестирования и получения ошибок к настройке корректности пар. Пример представлен на рисунке 3.

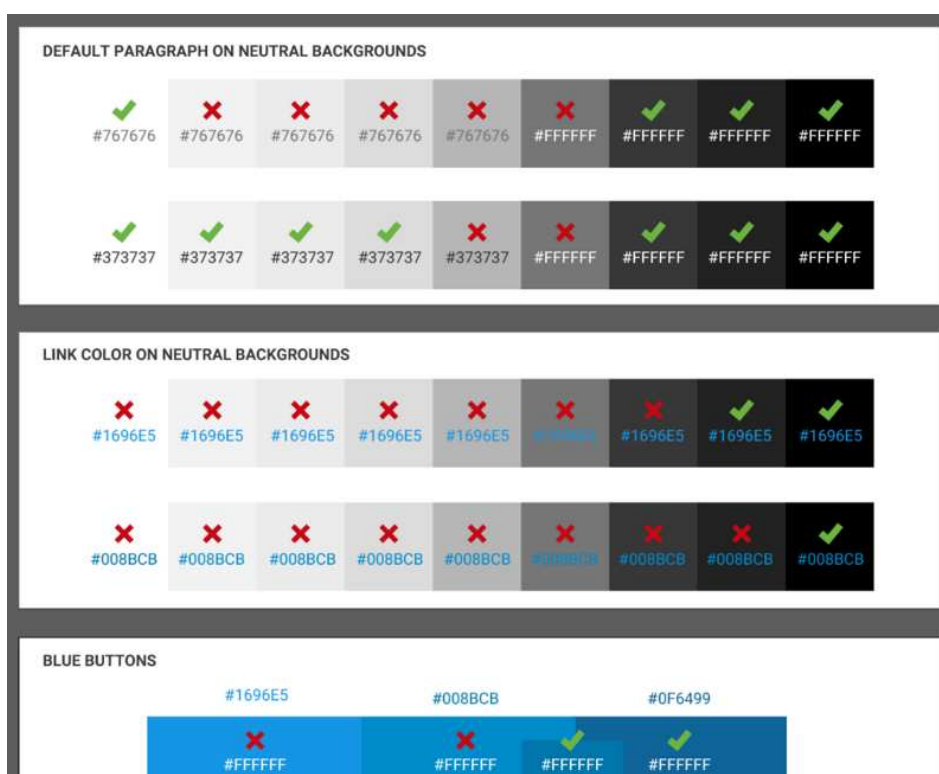


Рисунок 3– Пример цветового спектра

4. Светлое на темном, темное на светлом. При создании системы дизайнеру приходится обдумывать весь набор опций по цвету. Недостаточно просто тестировать проблемы сочетаемости по мере их возникновения. Вместо этого цветовую палитру нужно тщательно проанализировать перед выпуском системы для работы.

Это особенно важно при обратных сочетаниях цветов. Обычно система по умолчанию настроена на темный текст на светлом фоне. Тем не менее, многие меняют эти цвета ролями, будь это черный и белый на светлом и темном фоне, или оттенки другого основного или дополнительного цвета. Пример представлен на рисунке 4.

#000000	#454545	#686868	#848484	#A4A4A4	#EAEAEA	#F2F2F2	#F8F8F8	#FFFFFF
✗ 3.7	✗ 1.7	✗ 1	✗ 1.3	✗ 2.2	✓ 4.7	✓ 5	✓ 5.3	✓ 5.6
#0067C5	#0067C5	#0067C5	#0067C5	#0067C5	#0067C5	#0067C5	#0067C5	#0067C5
✓ 4.7	✗ 2.1	✗ 1.3	✗ 1.3	✗ 1.8	✗ 3.7	✗ 4	✗ 4.2	✓ 4.5
#4379BD	#4379BD	#4379BD	#4379BD	#4379BD	#4379BD	#4379BD	#4379BD	#4379BD

Рисунок 4– Пример сочетаемости темного на светлом и наоборот

5. Необходимо опускать нейтральные оттенки. Нейтральные оттенки – от темного, серого до светло-серого – обеспечивают важную подложку интерфейса. Загрузка системы нейтральными тонами, даже

если их несколько, создает риск, что в интерфейсе повсеместно будут использоваться размытые цвета. Другой частый результат – «каркасообразный» дизайн. Ни светлый, ни темный тип не обеспечивают комфортный контраст на средне-сером фоне.

Выбирайте несколько светло-серых и несколько темно-серых цветов, чтобы достичь комфортного контраста. Избегайте оттенков, которые находятся посередине между выбранными. Пример представлен на рисунке 5 [1].



Рисунок 5 – Спектр оттенков

ЛИТЕРАТУРА

1. 16 советов по разработке палитры интерфейса [Электронный ресурс] – 2018. – Режим доступа: <https://vc.ru/design/15711-colour-system> – Дата доступа: 20.04.2019
2. Как создать палитру, удобную для всех [Электронный ресурс]. – 2018. – Режим доступа: <https://m.habr.com/ru/post/444868/> – Дата доступа: 10.04.2019
3. Как цвет вашего сайта воздействует на психику клиента [Электронный ресурс] – 2018. – Режим доступа: <https://habr.com/ru/company/redhelper/blog/298508/> – Дата доступа: 19.04.2019.

УДК 004.5

Студ. И. С. Целуйко

Науч. рук. доц. О. А. Новосельская
(кафедра информатики и веб-дизайна, БГТУ)

ВЕБ-САЙТ ФРАНЦУЗСКОЙ ВЫПЕЧКИ «DELEPEL»

Сайт «deLepel» предназначен для пользователей, которые хотят купить французскую выпечку или приготовить её сами. Также можно будет почитать историю основных французских шедевров выпечки и выбрать пекарню в Минске, в которой можно купить товар.

Учитывая требования, предъявляемые к сайту и его наполнению, был выбран весьма популярный на данный момент способ структуризации и оформления страниц – многостраничный сайт.

Тщательно проанализировав веб-страницы, был сделан вывод о создании целевой страницы смешанного типа. Такая технология позволяет сочетать более высокую конверсию, по сравнению с обычной

страницей, а также удобна для SEO (поисковая оптимизация для повышения позиций сайта в результатах выдачи по заранее отобранным запросам), сравнимую с обычным сайтом.

Главная цель – создать интуитивно понятный сайт, с привлекательным дизайном и грамотно размещённой информацией.

На веб-сайте «deLepel» есть шесть основные страницы:

- Главная.
- Выпечка.
- История.
- Печём сами.
- Где купить.
- Блог.

Страница «Главная» используется для того, чтобы заинтересовать пользователя историей французской выпечки, высказыванием лучшего шеф-повара в мире, возможностью посмотреть рецепты и почитать статьи в блоге.

«Выпечка» предоставляет информация о продуктах и возможность заказать.

На странице «История» описаны самые основные шедевры французской выпечки с небольшой исторической справкой о них.

В разделе «Печём сами» представлены рецепты с пошаговой инструкцией по приготовлению.

«Где купить» даёт нам информацию о расположении и работе пекарней, где можно приобрести вкусную выпечку.

Почитать новости и интересные статьи можно на странице «Блог».

Логотип был выбран шрифтовой. Для этого подобран специальный шрифт «Scriptina», чтобы предать логотипу специфический вид.

Логотип построен по типу «Логослово», состоящий только из букв. Это один из самых простых типов, но в тоже время запоминающийся. Окончательный вид логотипа представлен на рисунке 1.

The image shows the final logo for 'de Lepel'. The text 'de Lepel' is written in a highly stylized, cursive script. The 'd' and 'e' are connected, and the 'L' is very large and loops around the 'e'. The 'p' and 'e' are also connected. The overall style is elegant and handwritten.

Рисунок 1 – Окончательный вид логотипа

Цветовая палитра сайта приведена на рисунке 2.



Рисунок 2 – Фирменные цвета

При написании текстов на сайте используется три шрифта (рисунок 3).

Alegreya Regular
Alegreya Bold

Playfair Display Bold

Proxima Nova Regular
Proxima Nova Semibold Italic
Proxima Nova Bold

Рисунок 3 – Шрифты сайта



Рисунок 4 – Макет главной страницы «deLepel»

В шапке сайта и заголовках используется шрифт Alegreya в двух начертаниях: Regular и Bold.

На информационных вывесках и рекламных блоках Playfair Display в начертании Bold. И в основных блоках текста используется шрифт Proxima Nova в следующих начертаниях:

- Regular,
- Semibold Italic,
- Bold.

Для того, чтобы начать верстку сайта, были разработаны макеты, по которым в дальнейшем были сверстаны страницы (рисунок 4).

Для разработки сайта использовались следующие технологии:

- HTML и CSS.
- AJAX.
- PHP.
- JavaScript.
- Bootstrap.
- MySQL.

Таким образом, поставленная цель по разработке интуитивно понятного веб-сайта с привлекательным дизайном и грамотно размещённой информацией достигнута.

УДК 004.5

Студ. И. А. Зенькович
Науч. рук. доц. О. А. Новосельская
(кафедра информатики и веб-дизайна, БГТУ)

ИГРОВОЙ ПОРТАЛ «LABYRINTH ESCAPE»

Игровой портал «Labyrinth Escape» предназначен для пользователей, которые хотят поиграть в несложную медитативную игру и для тех, кто хочет каких-то интересных испытаний. В игре предусмотрен соревновательный режим на двух игроков по локальной сети, а также игра с ботами.

Главная цель – создать интересную игру, а также сайт с привлекательным визуалом и продуманным дизайном. В дальнейшем этот проект может масштабироваться в игру с многопользовательским режимом или режимом совместного прохождения.

Целевые посетители сайта точно знают, что хотят приобрести или информацию, которую хотят увидеть на сайте.

Исходя из этого пользователей сайта «LabyrinthEscape» можно разделить на три группы:

- те, кто случайно попали на сайт;
- те, кто интересуется играми независимых разработчиков;
- те, кто ищет конкретную игру.

Возрастного ограничения игра и сайт не предусматривает ввиду отсутствия сцен с насилием, что позволяет распространить игру на широкий рынок как взрослых, так и молодых игроков.

На игровом портале «LabyrinthEscape» есть пять основных блоков.

- Игра.
- Описание игры.
- Скриншоты игры.
- Этапы разработки.
- Форма подписки.

Блок «Игра» используется для того, чтобы заинтересовать пользователя игрой и ее геймплеем с помощью видеоролика и возможностью поиграть.

«Описание игры» предоставляет информацию об игре, основных механиках и управлении.

В блоке «Скриншоты игры» показаны изображения игрового геймплея, интерфейса и основных механик.

Раздел «Этапы разработки» пошагово показывает путь к созданию этой игры и все пройденные этапы.

«Форма подписки» даёт возможность оставаться в курсе новостей от разработчика игры.

Логотип построен по типу комбинирования текста и символа. Это один из самых простых типов, но в тоже время запоминающийся. Окончательный вид логотипа представлен на рисунке 1.



LABYRINTH ESCAPE

Рисунок 1 – Окончательный вид логотипа

Цветовая палитра сайта приведена на рисунке 2.



Рисунок 2 – Фирменные цвета

При написании текстов на сайте используется два шрифта (рисунок 3). В шапке сайта и заголовках используется шрифт DIN 30640 Std Neuzzeit Grotest Bold Cond в начертании Bold. Остальной текст использует шрифт AvenirNext в начертаниях Regular, Semi-Bold, Black. Сочетание двух гротескных шрифтов составляют пару, которая повторяет настроение игры и правильно подает информацию на сайте.

**DIN 30640 STD NEUZEIT
GROTEST BOLD COND**

Avenir Next

Рисунок 3 – Шрифты сайта

Для того, чтобы начать вёрстку сайта, были разработаны макеты, по которым в дальнейшем были сверстаны страницы (рисунок 4).

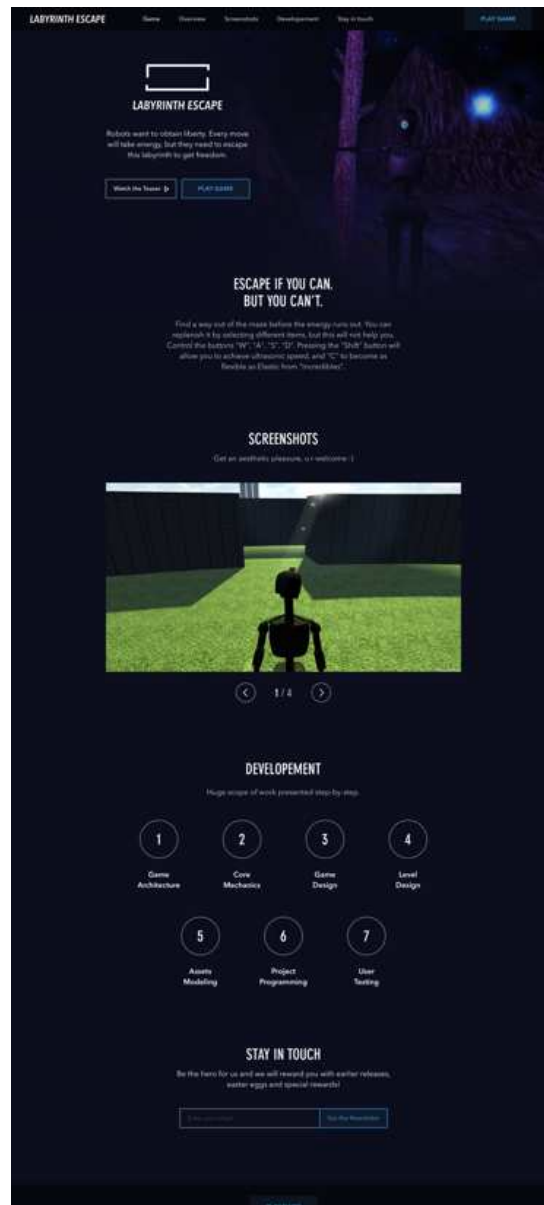


Рисунок 4 – Макет главной страницы «Labyrinth Escape»

Для разработки сайта использовались следующие технологии:

- HTML и CSS.
- PHP.
- JavaScript.
- Bootstrap.

Таким образом, поставленная цель по разработке игрового портала с привлекательным дизайном и грамотно размещённой информацией достигнута.

Студ. Л. С. Юхновец

Науч. рук. доц. О. А. Новосельская
(кафедра информатики и веб-дизайна, БГТУ)

ИСПОЛЬЗОВАНИЕ ВНЕШНИХ ПРОГРАММНЫХ СРЕДСТВ ДЛЯ СТИЛИЗАЦИИ ИЗОБРАЖЕНИЙ

В ходе данной работы будут рассмотрены возможности таких программных средств как Adobe Photoshop и 3DLUTCreator, для их качественной оценки возможности стилизации изображений в цветовой модели RGB и цветовом пространстве HSPne соответственно.

Актуальность задачи состоит в том, что на сегодняшний день у многих фотографов и дизайнеров возникает необходимость кардинального изменения изображения, основанная на его проработке в конкретных тонах.

Важно понимать, какие цвета и оттенки необходимы для стилизации изображений. Выбор цветового пространства, а их шестнадцать в программе 3DLUTCreator [1], основывается на предпочтениях и нуждах пользователей. Данный факт является уже первым преимуществом над программой Photoshop, которая использует всего три цветовых модели: RGB, CMYK, Lab. Перечень предлагаемых пространств первой программы изображен на рисунке 1.

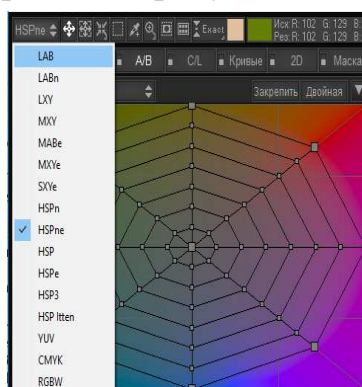


Рисунок 1 – Цветовые пространства

Не сложно заметить, что здесь используются такие пространства как: LXY (представляет собой упрощенное Lab пространство), MABe (где M – длина вектора в RGB кубе, а AB – углы вектора), RGBw (иными словами – CMYK наоборот, то есть убрана черная краска, но меньшим значениям цветовой краски соответствует более темное изображение). Цветовая модель RGB[2] (red, green, blue) – это цветовое пространство, состоящее из всех возможных цветов, которые могут быть получены аддитивным путем, проще говоря, в результате смешения красного, зеленого и голубого цветов. Значения RGB задаются

тройкой целых чисел от 0 до 255. Область применения рассматриваемой модели: телевидение, компьютерная графика. Цветовое пространство HSPne [2] (Hue – оттенок, Saturation – насыщенность, Perceived brightness – воспринимаемая яркость) (рисунок 2) представляет собой пространство HSP (рисунок 3) с расширенной областью нейтральных и слабонасыщенных цветов, а также с лучшим разделением светлых цветов по цветовому оттенку.

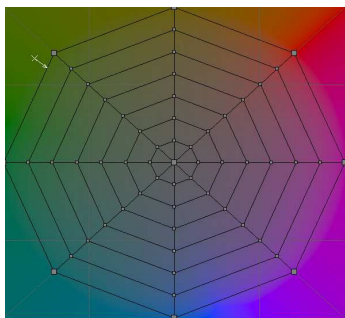


Рисунок 2 – Пространство HSPne

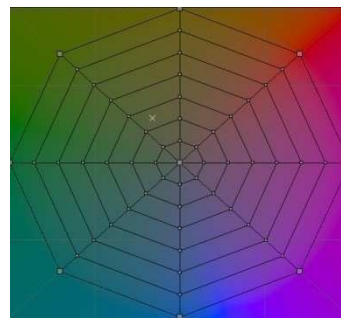


Рисунок 3 – Пространство HSP

Процесс преобразования цветовых погрешностей в программе Adobe Photoshop происходит на уровне каналов [3]. Каналы представляют собой изображения в градациях серого и характеризуются тем, что не используют никакие цвета за исключением черного и белого.

В данном случае будут рассматриваться каналы изображения в цветовой модели, и проводиться их соответствующая настройка.

3DLUTCreator – программа для профессиональной цветокоррекции фото. По своему функционалу и мощности она является более расширенной для возможности проведения стилизации изображения в сравнении с программами линейки Adobe. Одним из основных плюсов программы является также и то, что она не требует предварительной установки.

Цветокоррекция проводится с помощью изгибания сетки, привязанной к цветовой плоскости, содержащей насыщенность и оттенок. Интерфейс данной программы создает все необходимые условия как для полного изменения цветовой гаммы изображения, так и для работы с нужными цветовыми диапазонами по отдельности.

Оба рассматриваемых программных средства подходят для тоновой стилизации изображений (имитация характерных особенностей стиля). К наиболее распространенным средствам тоновой коррекции изображений относятся преобразования под стиль ретро, кино, коллоидный процесс и так далее.

Рассмотрим пример стилизации изображения в программе 3DLUTCreator по определенным тоновым составляющим. Для этого

необходимо выбрать изображение, содержащее достаточное число оттенков для работы с ними (рисунок 4).



Рисунок 4 – Исходное изображение

Для проведения стилизации достаточно использовать A/Всетку, которая позволяет изменить цвет объекта, путем перетаскивания нужного узла на цветовой сетке в область нужного оттенка и насыщенности, не затрагивая при этом других цветов.



Рисунок 5 – Измененный вариант

Выполнение подобного процесса в Adobe Photoshop не является возможным. Использование каналов с целью определенной тоновой стилизации приведет к изменению цвета всего изображения. Решение этой задачи заключается в использовании обтравочных масок и инструментов выделения (быстрое выделение, магнитное лассо, быстрая маска), однако, процесс займет достаточное количество времени, что является существенным недостатком.

Таким образом, можно выделить следующие плюсы и минусы рассматриваемых программных средств по сравнению друг с другом:

1. 3DLUT Creator.

Плюсы: не требует установки, доступность редактирования по любым цветам, большой спектр выбора цветовых профилей, эргономичный и понятный интерфейс, экономия времени.

Минусы: программа платная.

2. Adobe Photoshop

Плюсы: широкий выбор инструментов для работы с цветом, популярность.

Минусы: программа платная, работа всего с тремя цветовыми моделями, затратные по времени процессы обработки фотографий.

ЛИТЕРАТУРА

1. 3DLutCreatorЦветокоррекция по-новому [Электронный ресурс]/3DLUTCREATOR. –Режим доступа: <https://3dlutcreator.ru/>.– Дата доступа: 09.04.2019.

2. Цвет и его модели [Электронный ресурс]/ Сайт журнала Компьюарт. – Режим доступа: <https://compuart.ru/article/23772>. – Дата доступа: 10.04.2019.

3. Основные принципы работы с каналами в AdobePhotoshop [Электронный ресурс]/ Официальный сайт ADOBE. – Режим доступа: <https://helpx.adobe.com/ru/photoshop/using/channel-basics.html>. – Дата доступа: 10.04.2019.

УДК 004.932.72

Студ. А. С. Леонова

Науч. рук. доц. О. А. Новосельская
(кафедра информатики и веб-дизайна, БГТУ)

РАЗРАБОТКА 3D-МОДЕЛИ ПО ФОТОГРАФИИ

Трёхмерная компьютерная графика позволяет создавать объёмные трёхмерные сцены с моделированием условий освещения и установкой точек зрения, созданием анимации[1]. Она применяется при разработке дизайн-проектов интерьера, архитектурных объектов, в рекламе, при создании обучающих компьютерных программ, видеороликов, наглядных изображений деталей и изделий в машиностроении и др. Трёхмерное изображение отличается от плоского необходимостью построения геометрической проекции трёхмерной модели сцены на экране компьютера с помощью специализированных программ.

Достоинства трехмерной графики:

- созданиеобъемной модели с оптимальным числом составляющих деталей;
- наложение необходимых текстур, расстановка освещения, трансформация в процессе моделирования и рендеринга;
- управление качеством полученного изображения;

- возможность задания безфоновом формате.

Несмотря на преимущества трехмерных изображений, они не лишены некоторых недостатков, которые нужно учитывать при разработке графических проектов:

- высокие требования к аппаратной составляющей компьютера: его оперативной памяти, скорости работы процессора и т.д.;
- высокая трудоемкость разработки моделей объектов сцены;
- необходимость постоянно отслеживать взаимное положение объектов в составе сцены [2].

Так как объекты 3D-графики «бестелесны», они легко проникают друг в друга, важно контролировать отсутствие ненужного контакта между ними [3].

Целью работы является построение трехмерной модели архитектурного сооружения на основе двумерной фотографии, добавление анимации движения к отдельным объектам модели и окружения. Для этого в качестве основных задач можно выделить:

- поиск высококачественных изображений архитектурного сооружения.
- трехмерное моделирование сооружения и его окружения и их визуализация.
- добавление анимации движения к отдельным элементам трехмерной модели (нажатие ручки входной двери, открытие двустворчатой двери) и окружения (имитация колыхания листьев и веток деревьев от ветра, линейное движение облаков).

Решение первой задачи. Выбраны изображения архитектурного сооружения (здание историко-краеведческого музея) с различных ракурсов (рисунок 1).



Рисунок 1 – Изображения здания музея

Важно, чтобы изображения были высокого качества, для подробного рассмотрения составных деталей строения, используемых при строительстве материалов, т. е. для качественного всестороннего построения трехмерной модели.

Решение второй задачи. С помощью программного обеспечения для 3D-моделирования, анимации и визуализации при создании

игр и проектировании Autodesk 3ds Max была построена трехмерная модель здания музея.

При построении применялось моделирование сплайнами с целью получения обтекаемых форм, таких как резьба на наличниках, крыше, двери и т.д., EditablePoly для построения и преобразования примитивных составных форм. Для достижения реалистичности были наложены текстуры дерева, бетона, стекла, листвы и т.д. (рисунок 2).



Рисунок 2 – Текстуры для наложения

Для корректировки наложения текстур был использован модификатор UVWMap. Для создания кустов к объектам ChamferBox (куб со скругленными ребрами) применялся модификатор Noise (шум).

Решение третьей задачи. С помощью программных возможностей среды 3DsMax были разработаны анимации движения для ручки двери, створок дверей (с помощью инструмента SelectandMove), крон деревьев (с помощью изменения параметров модификатора Noise). В основе создания фона лежит базовый элемент Plane (плоскость), для которого в качестве текстуры назначен видеоролик плывущих облаков. Анимации назначались в режиме AutoKey (ключи ставятся автоматически сразу после перемещения ползунка и изменения анимируемых параметров).

Перед рендерингом получившейся модели была установлена камера, для которой также был разработан сценарий движения для добавления динамичности ролика. В качестве рендера был выбран ScanlineRenderer. Готовый трехмерный ролик был сохранен в формате *.avi, также было сохранено трехмерное изображение в формате *.jpeg. На рисунке 3 показаны кадры анимационного ролика.



Рисунок 3 – Кадры анимационного ролика

В результате разработан трехмерный анимационный ролик здания музея, созданный на основе двумерной фотографии. Процесс добавления анимации представляет собой трудоемкий процесс, требующий программных устройств больших мощностей для просмотра анимации в процессе работы и рендеринга модели. В ходе дальнейшего развития данной темы возможно создание роликов, показывающих экспозиционные залы музея.

ЛИТЕРАТУРА

1. Трехмерная графика[Электронный ресурс] / Estate.– 2015. – Режим доступа: http://esate.ru/article/cg/dizayn/trekhmernaya_grafika. – Дата доступа: 11.04.2019.
2. Преимущества 3D-графики перед 2D-объектами[Электронный ресурс] /Cpu3d. – 2015. – Режим доступа: <http://cpu3d.com/preimushestva-3d-grafiki-pered-2d-obektami/>. – Дата доступа: 12.04.2019.
3. Средства трехмерной графики[Электронный ресурс] /OpenGL. – 2015. – Режим доступа: <http://www.opengl.org.ru/teachpro-web-dizain/trekhmernaya-grafika.html>. – Дата доступа: 12.04.2019.

УДК 004.738.5:339:004.92

Студ. В. С. Шаплыко

Науч. рук. ст. преп. Т. В. Кишкурно
(кафедра информатики и веб-дизайна, БГТУ)

ВЛИЯНИЕ ДИЗАЙНА НА ПРОДАЖИ ИНТЕРНЕТ-МАГАЗИНА

Сегодня всё больше людей совершают покупки в интернет-магазинах. Их сейчас становится всё больше и больше. Единицы из них имеют хорошую конверсию и высокую прибыль. Кроме грамотно продуманной стратегии и хорошего маркетинга, необходимо создать привлекательную оболочку, внешний вид. По статистике, человек в течение лишь нескольких секунд решает, оставаться ему на сайте или нет. Задача сайта интернет-магазина задержать пользователя, сформировать у него хорошее мнение о магазине, вызвать доверие и привести к покупке.

Многие владельцы коммерческих сайтов часто недооценивают значение дизайна для успеха своего бизнеса. Однако качественный и приятный на вид дизайн вызывает соответствующие положительные ощущения у посетителей сайта. Он привлекает внимание пользователя, вызывает желание и интерес, что приводит к действию, то есть к совершению покупки. Некрасивый и устаревший внешний вид сайта, наоборот – делает единственным желанием пользователя стремление поскорее закрыть вкладку в браузере.

Обновление дизайна сайта, при условии грамотной реализации, обеспечивает целый ряд преимуществ:

- повышение лояльности со стороны существующей аудитории и новых посетителей;
- увеличение коэффициента конверсии;
- рост общего уровня продаж товаров или услуг;
- снижение показателя отказов и улучшение поведенческих факторов;
- соответствие сайта современным технологиям верстки и тенденциям в дизайне;
- оптимизация структуры проекта и т. п.

Ранее адаптивная верстка на сайте рассматривалась как приятный бонус, сейчас является настоящей необходимостью. Доля трафика с мобильных устройств стремительно растет, и в ряде тематик уже превышает переходы с десктопных PC и ноутбуков. Адаптивная верстка – обязательно условие, которому должен соответствовать дизайн современного сайта интернет-магазина. На экране смартфона и планшета интернет-магазин с устаревшим, неадаптивным дизайном, не только некрасиво выглядит, главное – им совершенно неудобно пользоваться и делать покупки.

На главной странице предпочтительно использовать правило 7 ± 2 , которое говорит, что одновременно человек может воспринимать только 7 ± 2 объекта. Данное правило исключает появление информации, назначение которой может быть не понятно. Блоки будут явно выражены. Простота – это то, к чему нужно стремиться при разработке интернет-магазина.

Фильтр и сравнение товаров – неотъемлемая часть покупки товара в интернет-магазине. Только выбирая, клиент может разрешить свои сомнения при покупке. Давая пользователю множество возможностей фильтровать товар, система не помогает, а только путает. Форму фильтрации необходимо сделать максимально лаконичной с дополнительной возможностью погрузиться в детали фильтрации, если основных фильтров окажется недостаточно. В момент фильтрации желательно использовать все экранное пространство, поскольку пользователь в это время все равно ничего другого осознанно делать не сможет.

Подходы к композиции, дизайну интернет-магазина зависят от его масштаба. Интернет-магазин, содержащий 100 позиций, и интернет-магазин, содержащий миллион позиций, будут иметь разные подходы к прототипам дизайна. Если в интернет-магазине присутствует один товар, то необходимо использовать одностраничный сайт с описанием данного товара. С увеличением роста количества товара добавляется обязательное расположение категорий, дополнительных разделов, фильтрации, поиск.

Акценты – визуальные элементы для привлечения внимания, которые должны выделить главное, предложить альтернативу, сфокусировать внимание. Один из акцентов встречается в интернет-магазинах – бирки на товар.

Баннер – это большое изображение в верхней части страниц магазина, которое может рекламировать товары или акции. Яркий баннер призван не только остановить взгляд и вызвать приятные эмоции у покупателя – на него следует добавить ссылку. Баннер должен быть как для красоты, так и для пользы.

Группа товаров, выдаваемая при прямом заходе на главную страницу сайта, именуется витриной. Товары в витрине являются визитной карточкой интернет-магазина. Главная страница – самая посещаемая на сайте, поэтому так важно разместить на ней анонсы самых лучших ваших товаров. Неактуальные цены и товары являются проблемой в любом месте интернет-магазина, но находясь на витрине, они наносят куда больший ущерб продажам. В витрине стоит отображать лучшие предложения, выбирая для них превью наиболее профессиональные фото.

Промопанель – это панель с информацией и кнопкой с призывом к действию, которая выводится над шапкой сайта на всех страницах, где ее точно заметит потенциальный клиент. Там находится наиболее важная информация.

Хороший веб-дизайнер должен практиковать «эстетическое преобразование», чтобы создавать визуально потрясающие веб-сайты, которые сочетают ваши цели увеличить спрос на продукцию или услугу. Они должны разработать дизайн интернет магазина и учитывать весь путь, который должен пройти пользователь, создать привлекательные визуальные эффекты, которые на самом деле помогут вам в достижении высоких показателей конверсии, от заголовков к кнопкам.

ЛИТЕРАТУРА

1. Habr [Электронный ресурс]. – 2018. – Режим доступа: <https://habr.com/ru/company/advantshop/blog/160245/> – Дата доступа: 05.04.2019
2. Sezzam [Электронный ресурс]. – 2019. – Режим доступа: <https://sezzam.com.ua/vliyanie-dizayna-internet-magazina-na-uvelicheniya-prodazh/> – Дата доступа: 05.04.2018
3. Webexpert [Электронный ресурс]. – 2019. – Режим доступа: <https://webexpert.com.ua/design-and-sales> – Дата доступа: 06.04.2018
4. Evo.business [Электронный ресурс]. – 2018. – Режим доступа: <https://evo.business/8-fishek-dizajna-internet-magazina/> – Дата доступа: 06.04.2018

ОБ ИСПОЛЬЗОВАНИИ SVG-МАСОК

SVG (ScalableVectorGraphics), т. е. "масштабируемая векторная графика" – это формат векторной графики, основанный на XML. Этот формат разработан и рекомендован консорциумом W3C. Он знаменит тем, что поддерживает как статичные изображения, так и анимированные, что выгодно отличает его от множества других графических форматов [1].

SVG существует с 1999 года, а с 16 августа 2011 включена в рекомендации W3C. Для веб-сайтов существует SVG, которая согласно официальной спецификации на w3.org является языком для описания двумерной графики в XML.

Область применения SVG в веб-дизайне весьма обширна, например:

- Иллюстрации. SVG позволяет создавать абстрактные фигуры различных форм, что позволяет сделать сайт более привлекательным и интересным.

- Иконки. Чтобы сделать сайт уникальным можно использовать авторские иконки, графики и схемы. Обычно для их создания используется векторный формат, чтобы на различных форматах изображение смотрелось одинаково четко, не теряя качества.

- Текст. Иногда, например, для больших заголовков, удобно перевести текст в кривые и представить на сайте как SVG.

- Декоративные элементы. Различные геометрические фигуры, формы и линии, служащие в качестве декоративных элементов страницы можно представлять с помощью SVG. Это позволит при необходимости быстро изменить их свойства. Также они не будут сильно загружать сайт, что хорошо скажется на скорости загрузки страницы.

- Логотипы. Традиционно для создания логотипов используют векторную графику, чтобы было возможность перенести изображение на полиграфическую продукцию. В веб-дизайне для отображения логотипа можно использовать SVG.

- Анимация. В SVG есть возможность анимации изображения. Это придает сайту больше динамичности, делает его более привлекательным и запоминающимся для пользователя [2].

SVG имеет ряд преимуществ, что выгодно выделяет его на фоне альтернативных форматов для веб-разработки. Это такие преимущества как:

– Масштабирование. SVG–векторный формат, то его масштабирование не является большой проблемой, так как качества не портится независимо от коэффициента масштабирования.

– Уменьшение HTTP-запросов. Использование SVGувеличивает скорость загрузки сайтов, за счет того, что требует меньшего числа обращений к серверу.

– Стайлинг и скриптинг. SVGлегко взаимодействует с CSS, что позволяет без затруднений менять оформление сайта. Объектам SVGприсущи те же свойства, что и любым другим объектам HTML. Например, границы, заливка, прозрачность, позиционирование и т.д.

– Анимация. Взаимодействие с JavaScript позволяет анимировать объекты SVG, а также динамически изменять их. Например, изменение заливки на выбранную пользователем или изменение текстовой информации.

– Малый размер. За счет специфики построения объектов, SVGфайлы весят меньше, чем растровые, несущие в себе то же графическое изображение.

Использование SVG выгодно в проектах, предполагающих использовать на дисплеях как высокого, так и низкого разрешения. В таких случаях необходимо, чтобы графика погружалась быстро и в нужном качестве, поэтому без SVGобойтись трудно. Также перевод уже готового проекта на SVGграфику также не вызывает трудностей или больших временных затрат. Можно даже не заменять старые изображения, а просто добавить новые в SVG формате[3].

Что такое SVG-маска. В данном контексте маска – это дополнительный слой над контентом, который частично его скрывает. На первый взгляд может показаться, что кроме обрезания каких-нибудь картинок в форме шестиугольника применений этому нет. Ведь обычно именно такие примеры приводят в статьях о масках. Но если посмотреть немного шире и добавить движение, то все начинает играть новыми красками.

В базовом варианте маска создается примерно так:

```
<svg>
<defs>
<mask id='mask-1'>
    . . .
</mask>
</defs>
<image mask='url(#mask-1)' />
</svg>
```

Вместо картинки может быть абсолютно любой объект. Маску вполне можно применять ко всему, что мы там рисуем. Сама она так-

же может состоять из каких угодно компонентов, например, мы можем сделать ее из картинки или использовать многоугольники и кривые [4].

Удобнее всего, если элементы маки будут состоять из фигур различных оттенков серого. В таком случае, они будут регулировать прозрачность изображений находящихся под ними. Если цвет маски черный, то элемент она применяется, становится полностью прозрачным. А если цвет белый, то соответственно – полностью непрозрачным. Серые же цвета дают промежуточный эффект.

В данном примере маска накладывается на анимированный круг, ограничивая область его видимости в форме диалогового облака, и текста «SVG» по центру.

Пример применения SVG-маски:

```
<style>
#img{
    fill: url(#fill);
    background-repeat: repeat;
}
</style>
<svg viewBox='0 0 100 60'>
<rect width="100%" height="100%" fill="#000"/>
<pattern id="fill" width="100" height="100" patternUnits="userSpaceOnUse">
    <image xlink:href="5x7ft-xt-2929.jpg" width="80"
height="80"/>
</pattern>
<defs>
<mask id='mask-1' x='0' y='0' width='100%'
height='100%'>
    <rect width="100%" height="100%"
fill="#fff"/>
    <path id="buble" stroke="#fff" transform="scale(0.47)" stroke-width="1" fill="#000" d="M115
15.25c27.614 0 50 18.188 50 40.625 0 22.437-22.386 40.625-
50 40.625-2.652 0-5.255-0.169-7.795-0.493-10.74 10.74-23.56
12.666-35.955 12.949v-2.629c6.693-3.279 12.5-9.252 12.5-
16.078 0-0.953-0.074-1.888-0.211-2.802-11.308-7.448-18.539-
18.824-18.539-31.573 0-22.437 22.386-40.625 50-40.625z"/>
    <text id='text' x='31' y='33' font-
family='Verdana' font-weight='bold' font-size="20"
fill='#fff'>SVG</text>
</mask>
</defs>
<circle id="img" r="30" cx="50" cy="30" fill="#fff"/>
<animateTransform xlink:href="#img" attributeName="transform" attributeType="XML" type="rotate" from="0
50 30" to="360 50 30" dur="5s" begin="0s" fill="freeze"
repeatCount="indefinite"/>
```

```
<rect width="100%" height="100%" fill="#000"
mask='url(#mask-1)'/>
</svg>
```

Результат выполнения кода приведен на рисунке.



Рисунок - Результат выполнения кода

SVG удобна главным образом потому, что не вызывает проблем с разрежением изображений и позволяет обходиться без тяжелых графических файлов. А возможность добавления анимации значительно расширяет область ее применения. Поэтому, на сегодняшний день SVG является оптимальным решением для многих задач.

ЛИТЕРАТУРА

1. Что такое SVG? [Электронный ресурс]. – Режим доступа: – 21.03.19.
2. Полное руководство по SVG: векторная графика в веб-дизайне и этапы ее создания [Электронный ресурс]. – Режим доступа: <https://idbi.ru/blogs/blog/polnoe-rukovodstvo-po-svg> – Дата доступа: 21.03.19.
3. Причины нарастающей популярности SVG в Photoshop и веб-дизайне [Электронный ресурс]. Режимдоступа: <https://wayup.in/blog/why-svg-is-popular-in-photoshop-and-web-design>. – Датадоступа: 21.03.19.
4. SVG маски и вау-эффекты: о магии простыми словами [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/349362>. – Дата доступа: 21.03.19.

Студ. Г.А. Панасик

Науч. рук. ст. преп. Н. И. Потапенко
(кафедра информатики и веб-дизайна, БГТУ)

ВЕБ-САЙТ ВЕТЕРИНАРНОЙ КЛИНИКИ «ПЁССИКОТ»

На сегодняшний день во многих сферах человеческой жизнедеятельности используются ресурсы всемирной паутины. В итоге, разработка и создание подобных порталов относится к категории важнейших и наиболее актуальных мероприятий, которые позволяют фирмам и организациям достичь поставленных целей.

Актуальность создания сайта любого вида – донесение информации до широкого круга пользователей интернета. Реклама – двигатель прогресса, поэтому актуальность создания сайта не подвергается сомнениям.

«Пёссикот» – веб-сайт, на котором пользователи могут просматривать информацию о предоставляемых услугах, ценах и специалистах, а также реализовать онлайн запись на приём или консультацию.

Пользователи могут легко найти информацию, которая им необходима, а элементы и структура сайта стимулируют их к обращению в клинику.

Ниже будут проанализированы аналоги сайтов ветеринарных клиник. Анализ будет осуществляться путём оценки сайтов по различным критериям: внешний вид, содержание контента, юзабилити, технология разработки, простота и понятность в нахождении интересующей информации, креативность, собственный стиль. Поэтому были проанализированы следующие конкуренты:

1. Доктор Вет – веб-сайт белорусской ветеринарной клиники[1]. В ходе анализа были выявлены его основные функции:

- онлайн запись на приём;
- поиск товаров в одноимённой сети аптек;
- публикация новостей клиники и статей, касающихся домашних животных;
- подробное описание услуг;
- предоставление цен;
- просмотр адресов и контактов клиник;
- просмотр акций;
- оплата через сайт;
- просмотр часто задаваемых вопросов;
- выбор языка.

Сайт выполнен в несуразном стиле, слишком заполнен бесцельной информацией различного характера, много разных цветов, плохо

прорисованные иконки и грязные тени. У сайта нет выдержанного и уютявшегося стиля. Из-за большого количества неправильно поданной информации на странице сложно найти необходимую информацию. Весовая композиция подобрана не правильно: заголовки не привлекают внимания. Сайт содержит всю информацию, которая может понадобиться пользователю, однако эстетическая составляющая находится на низком уровне. Всё это в совокупности отталкивает пользователей от сайта.

Однако немалым плюсом является то, что сайт адаптивный.

Проведя анализ, можно сделать вывод, что у сайта очень много возможностей и наполненный контент, однако довольно запутанная структура и большое количество бесцельной информации. Со стороны пользователя можно сказать, что искать что-то на этом сайте очень затруднительно и оставаться на нём не хочется, несмотря даже на богатый функционал и наполненность.

2. PolandAnimalHospital – веб-сайт польской ветеринарной клиники [2]. В ходе анализа были выявлены следующие его основные функции:

- подписка в социальных сетях;
- подробного описания услуг;
- информация о ветеринарах;
- онлайн запись к ветеринару;
- просмотр адресов и контактов клиник;
- обратная связь.

На сайте отсутствует внутренняя сетка элементов. Всю правую часть занимает не сразу понятный элемент, который нельзя закрыть и который занимает много места. Также довольно непонятная логика открытия ссылок и расположения вкладок, неправильный перевод и композиция. Имеются некоторые пустые вкладки. Сайт содержит минимум информации, которая может понадобиться пользователю и эстетическая составляющая находится на низком уровне. Всё это в совокупности отталкивает пользователей от сайта.

Сайт адаптивен. После просмотра с мобильного устройства сложилось впечатление, что сайт разрабатывался под мобильные устройства, так как в этом случае расположение элементов более логичное.

Проведя анализ, можно сделать вывод, что у сайта почти отсутствует необходимый контент, довольно запутанная структура и большое количество бесцельной информации. Со стороны пользователя можно сказать, что искать что-то на этом сайте очень затруднительно и практически бесполезно, поэтому оставаться на нём не хочется.

В ходе проведенного анализа были выявлены достоинства и недостатки веб-сайтов. Далее при разработке веб-сайта «Пёссикот» большинство из выявленных недостатков были устранены, а достоинства позаимствованы.

Веб-сайт одинаково корректно отображается во всех популярных браузерах как с картинками, так и без.

Веб-сайт представляет собой интернет ресурс, соединяющий в себе интернет-магазин и сервис по предоставлению информации. Разработанный ресурс имеет следующий функционал:

- в наглядной форме представляет полный и актуальный ассортимент предоставляемых услуг с их описанием и ценой;
- в наглядной форме представляет полный и актуальный список принимающих ветеринаров с описанием предоставляемых ими услуг;
- обеспечивает возможность зарегистрированным пользователям оставлять комментарии о приёме и влиять на рейтинг ветеринаров;
- обеспечивает удобный инструмент для записи на приём;
- предоставляет моментальную обратную связь;
- обеспечивает возможность однократного ввода информации о питомце, которая впоследствии будет использоваться для получения талона;
- обеспечивает возможность регистрации и сохранения истории посещений;
- обеспечивает возможность онлайн покупки специализированных товаров;
- обеспечивает возможность вызова ветеринара на дом;
- обеспечивает возможность онлайн консультации.

Потенциальными пользователями ресурса являются люди, имеющие домашних питомцев. Возрастных ограничений, либо нацеленности на определённый пол нет.

Во время разработки постоянно проводились различного рода анализы, опросы и анкетирование потенциальных пользователей, для выявления несовершенств веб-сайта и способов их устранения.

Было изучено много статей про пользовательский опыт в дизайне. По итогам, были сделаны выводы, которые повлияли на концепцию дизайна веб-сайта и последующее размещение элементов страниц.

Макеты страниц веб-сайта (рисунки 1–3). Макеты и прототипы платформы разрабатываются в приложении Adobe Photoshop. При разработке учитываются правила и принципы дизайна.

При выборе цветов опирались на психологию цвета и особенности его восприятия пользователями.

Все иллюстрации были разработаны самостоятельно. В качестве основы выступали несколько скачанных иллюстраций, которые в последствии потерпели ряд изменений:

1. некоторые иллюстраций были объединены между собой, для получения желаемого результата;
2. в последствии все иллюстрации были перекрашены в цвета, используемые в фирменном стиле;
3. некоторые иллюстрации были отрисованы самостоятельно.

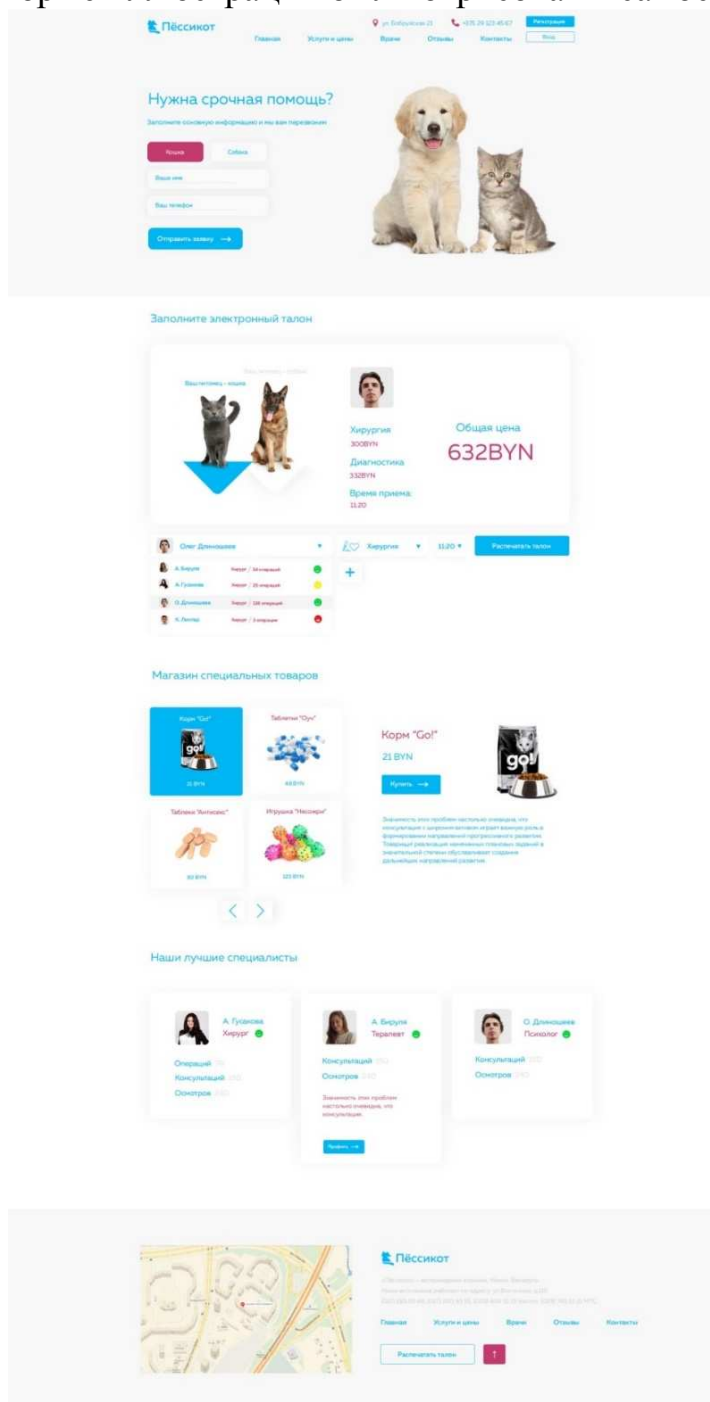


Рисунок 1 – Макет главной страницы

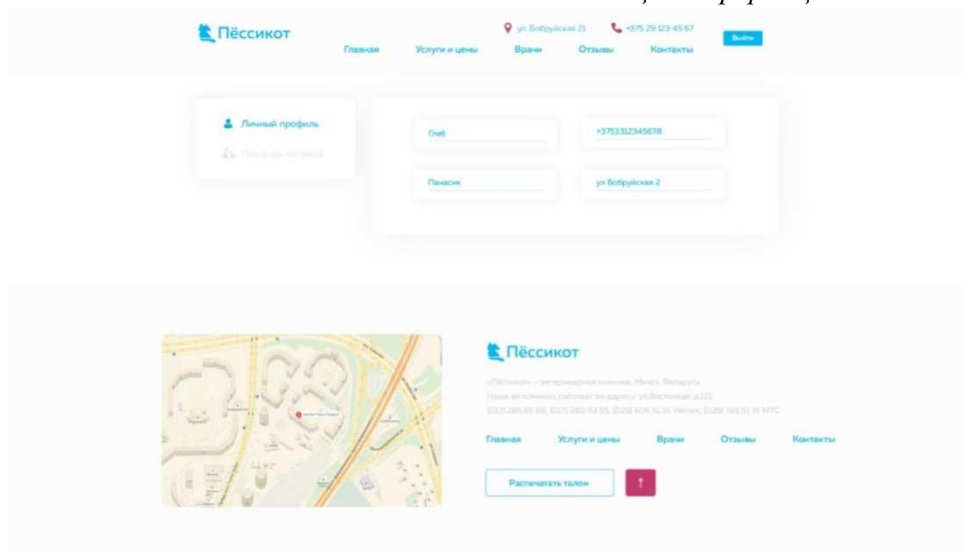


Рисунок 2 – Макет личного кабинета

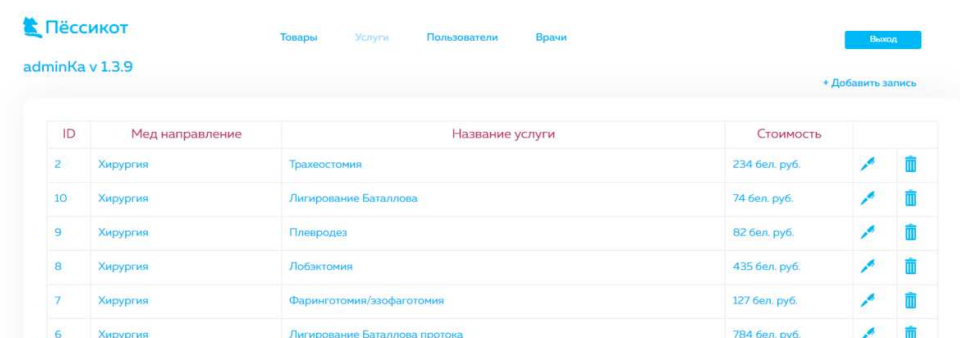


Рисунок 3 – Макет административной части

Вся функциональная часть разработана с помощью языка PHP. Каркасы страниц созданы с помощью языка разметки HTML 5. Внешний вид описывается при помощи формального языка CSS. Интерактивность сайта создаётся с помощью языка программирования JavaScript и библиотеки jQuery, которые так же по умолчанию подключены в Yii2.

База данных для проекта создавалась в СУБД MySQL с использованием веб-приложения phpMyAdmin.

Разработанный веб-ресурс является персональным сайтом ветеринарной клиники и содержит всю необходимую информацию для владельцев питомцев и имеет возможность администрирования, что существенно упрощает его дальнейшую техническую поддержку.

ЛИТЕРАТУРА

1. <http://dv.by>
2. <http://www.polandanimalhospital.com>

УДК 339.138

Студ. А. Н. Гусакова, С. В. Жуков
Науч. рук. ст. преп. Н. И. Потапенко
(кафедра информационных технологий, БГТУ)

ИНТЕРНЕТ-ПЛАТФОРМА КОНСАЛТИНГОВЫХ УСЛУГ «CONSU»

Емкость мирового консалтингового рынка на конец 2017 года составляет 250 миллиардов долларов США. За рассматриваемый период с 2011 по 2017 года мировой рынок консалтинговых услуг рос со среднегодовым темпом в 4,1.

Более 80 % мировых консультантов являются фрилансерами, что существенно затрудняет как поиск соответствующего требованиям и потребностям консультанта со стороны бизнеса, так и поиск работы для консультантов.

Создание платформы «Consu» может помочь нивелировать существующие недостатки отрасли, а именно затруднение в поиске квалифицированных консультантов, поиск консультантов для решения определенной задачи соответствующих целям и бюджету бизнеса, централизация консультантов-поставщиков услуг и бизнес-покупателей услуг.

Consu – интернет-площадка, которая позволит консультантам предложить себя бизнесам в качестве помощников в рамках своей области компетенций, а бизнесу – возможность найти профессионалов, которые решат актуальную проблему.

В настоящий момент на рынке восточной Европы нет профессионального агрегатора консалтинговых услуг. В основном, там представлены только платформы электронной коммерции (далее маркетплейсы) одним из видов товара на которых представлены услуги, в которые входит и консалтинг.

Сейчас основными конкурентами платформы являются:

- консалтинговые агентства;
- неспециализированные маркетплейсы.

Анализ консалтинговых агентств не проводился, так как все их сайты больше ознакомительного характера. Функционально платформа больше схожа с сайтами фриланса. Поэтому были проанализированы следующие конкуренты:

1. FL.ru – крупнейшая русскоязычная биржа удаленной работы [1]. В ходе анализа были выявлены следующие плюсы:

– **для консультантов:** удобный интерфейс; регистрация через соц. сети (быстрый вход); личный аккаунт пользователя; редактор информации в аккаунте; чат с пользователями; выбор специализации из

каталога; возможность добавить портфолио; возможность добавить резюме; возможность добавить в избранное; раздел статистика переходов в аккаунт; возможность добавить (удалить) в команду (из команды); рейтинг; детальный поиск; конкурсы; возможность оставлять комментарии к проектам; возможность написать сообщение другому исполнителю; возможность удалить аккаунт; возможность написать администратору; отображение режима занятости (принимаю заказы/занят); отображение режима офлайн/онлайн.

– **для представителей бизнеса:** гарантия возврата денег; способы резервирования и выплат; возможность опубликовать заказ; возможность выбрать исполнителя; возможность найти услугу в каталоге; возможность найти услугу при помощи фильтра и поиска; кнопка предложить заказ исполнителю; контакты исполнителя; возможность написать отзыв и оценить исполнителя; возможность разместить вакансию; возможность разместить конкурс; просмотр сделок (в том числе безопасных); есть форум; есть база знаний.

Так же были выявлены следующие минусы:

– **для консультантов:** очень много различных тематик; нет выбора специализации из каталога; нет чата с администратором; нет всплывающих подсказок возле иконок.

– **для бизнесов:** нет чата для связи с администратором; очень много различных тематик.

2. UpWork – агрегатор фрилансерских услуг [2]. В ходе анализа были выявлены следующие плюсы:

– **для консультантов:** удобный интерфейс; личный аккаунт пользователя; выбор специализации из каталога не на этапе регистрации; выбор уровня опыта работы при регистрации; возможность добавить навыки при регистрации; чат с администратором; редактор информации в аккаунте; чат с пользователями; выбор специализации из каталога; возможность добавить портфолио; возможность добавить видео в аккаунт; возможность добавить (удалить) в команду (из команды); статистика и отчеты в лк; детальный поиск; уведомления с настройками; рейтинг; биржа вакансий и заказов с фильтром; возможность оставлять комментарии к проектам; возможность написать сообщение другому исполнителю; возможность добавлять в избранное; возможность удалить аккаунт; статистика посещения аккаунта; отображение режима занятости (принимаю заказы/занят); отображение режима офлайн/онлайн; приложение для установки на компьютер (для работы офлайн с функцией отслеживание времени); почасовая ставка за работу; модерация аккаунта после регистрации; указание навыков, образования и трудовой деятельности

сти;тесты, повышающие шансы получить заказ;двухэтапная аутентификация;форум.

– **для представителей бизнеса:**возможность при регистрации внести данные о компании;модерация аккаунта после регистрации;указаниенавыков, образования и трудовой деятельности;тесты для исполнителей, проверяющие их знания и умения;двухэтапная аутентификация.

И следующие минусы:

– **для консультантов:** очень много различных тематик;нет регистрации через соц. сети (быстрый вход);нет возможности добавить ссылки на другие соц. сети;нет возможности прикрепить резюме;возможность заказать работу у выбранного фрилансера.

– **для представителей бизнеса:**нет каталога по тематикам;необходимо указывать телефон при регистрации;необходимо указывать e-mail при регистрации;неудобный фильтр.

В ходе проведенного анализа были выявлены достоинства и недостатки платформ. Далее, при разработке платформы Consu, выявленные недостатки были устранены, а большинство достоинств позаимствованы.

Целью данного проекта является создание профессионального маркетплейсаконсалтинговых услуг, где консультанты/опытные сотрудники смогут предложить себя бизнесам/заказчикам в качестве помощников в рамках своей области компетенций, а бизнес сможет легко найти профессионалов, которые решат его проблему.

Платформа даст следующие возможности для заказчика:

1. Экономия времени.

2. Высокие качества услуг, поскольку предварительно модераторами проводится подтверждение опыта и компетенций соискателя по различным критериям (образование, практический опыт, проекты и т.д.).

3. Удобство:

– одна площадка, где точно есть нужный заказчику консультант;

– интуитивно-понятный и удобный интерфейс облегчает выбор и установление контакта с поставщиком услуг;

– на платформе удобно и быстро можно найти исполнителя, стоимость прописана, много специализаций;

– не надо формировать ТЗ, чтобы найти исполнителя.

4. Гарантии – заказчики получают гарантированно высокого качества услугу. Если в процессе оказания услуги появляется необходимость в консультанте другого направления, платформа решает и эту задачу.

5. Безопасные сделки – исполнители получают оплату только после подтверждения выполнения услуги заказчиком. До этого деньги резервируются на счету.

Для поставщиков услуг платформа даёт следующие возможности:

– наличие единого специализированного маркетплейса для предложения своих услуг;

– условия стабильны, условия сделок ясны – поставщик понимает, что получит гарантированную оплату за оказанные услуги при их выполнении;

– экономия времени и денег – нет необходимости вкладывать время и деньги в поиск клиентов, клиенты сами находят исполнителей благодаря платформе. Более того, клиенты в первую очередь обращаются к платформе, понимая высокое качество услуг.

Был проанализирован рынок консалтинговых услуг и бизнесов. По итогам анализа была определена целевая аудитория платформы:

1. Описание целевой аудитории консультантов/поставщиков услуг:

1.1. Поставщиками услуг будут являться физические лица в возрасте от 28 до 70 лет, имеющие практический опыт профессиональных консультантов, с высшим образованием, желающие себя продвигать на рынке консалтинговых услуг.

1.2. Юридические лица, оказывающие консалтинговые услуги (малый и средний бизнес).

2. Описание целевой аудитории заказчиков:

2.1. Возраст – 30-55 лет, пол значения не имеет, являются ЛПР и ВЛПР, которые понимают задачи/проблемы своего бизнеса, осознают необходимость их решения, но не могут их решить без помощи внешних специалистов (например, директора, хозяева бизнеса).

По итогам проведённых анализов были выявлены минимальные необходимые функциональные возможности:

1. Платформа будет иметь административную панель для управления площадкой. Администратор будет иметь следующие возможности:

1. Работать с отзывами:

- проверять;
- подтверждать;
- удалять;
- редактировать.

2. Работать с пользователями:

- блокировать;
- изменить данные;

- удалить.
- 3. Рассылать уведомления.
- 4. Добавлять и редактировать новости.
- 2. Незарегистрированный пользователь может:
 - 1. Просматривать биржу заказов.
 - 2. Просматривать профили заказов.
 - 3. Просматривать профили бизнесов.
 - 4. Искать заказы с использованием фильтров и сортировок.
 - 5. Предлагать новости администратору.
 - 6. Общаться с администратором по средствам обратной связи.
 - 7. Просматривать общую информацию о платформе, её работе и предоставляемых услугах.
 - 8. Зарегистрироваться как консультант или представитель бизнеса (в том числе и с использованием социальных сетей).
 - 9. Авторизоваться как консультант или представитель бизнеса.
- 3. Авторизованный как консультант пользователь может:
 - 1. Редактировать информацию своего профиля:
 - общую информацию о себе;
 - добавлять и редактировать образование;
 - добавлять и редактировать резюме;
 - добавлять и редактировать портфолио;
 - редактировать контакты;
 - изменить пароль.
 - 2. Удалить свой профиль.
 - 3. Перейти в кабинет бизнеса.
 - 4. Изменять статусы (занят, принимаю заказы).
 - 5. Просматривать оставленные ему отзывы (сортировать их).
 - 6. Просматривать выполненные проекты (сортировать их).
 - 7. Откликаться на заказы.
 - 8. Общаться с заказчиком.
 - 9. Получать уведомления о новых отзывах, заказах, одобренных откликах и сообщениях от заказчиков.
 - 10. Оставлять отзывы бизнесам после завершения сделки.
 - 11. Выйти из аккаунта.
- 4. Авторизованный как представитель бизнеса пользователь:
 - 1. Редактировать информацию своего профиля:
 - общую информацию о себе;
 - добавлять и редактировать резюме;
 - редактировать контакты;
 - изменить пароль.
 - 2. Удалить свой профиль.

3. Публиковать проекты.
4. Редактировать проекты.
5. Удалять проекты.
6. Перейти в кабинет Консультанта.
7. Просматривать оставленные ему отзывы (сортировать их).
8. Просматривать выполненные проекты (сортировать их).
9. Просматривать профили откликнувшихся консультантов.
10. Общаться с откликнувшимися консультантами.
11. Получать уведомления о новых отзывах, откликах и сообщениях от консультантов.
12. Заключать сделки.
13. Завершать сделки.
14. Оставлять отзывы консультантам, которые выполняли проект, после завершения сделки.
15. Выйти из аккаунта.

Каждый пользователь будет видеть время последней активности другого, либо онлайн, если пользователь находится на платформе сейчас.

Во время разработки постоянно проводились различного рода анализы, опросы и анкетирование потенциальных клиентов, для выявления несовершенств платформы и способов их устранения.

Было изучено много статей про пользовательский опыт в дизайне. По итогам, были сделаны выводы, которые повлияли на концепцию дизайна платформы и последующее размещение элементов страниц.

Скриншоты внешнего вида страниц платформы представлены на рисунках 1–3.

Все макеты и прототипы платформы были разработаны в приложении Figma, которая позволила редактировать онлайн как макеты, так прототипы. При разработке учитывались правила и принципы дизайна.

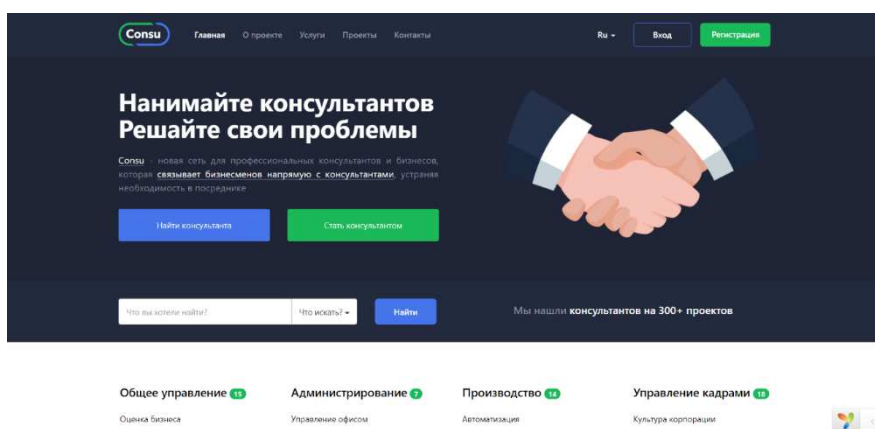


Рисунок 1 – Скриншот первого экрана главной страницы

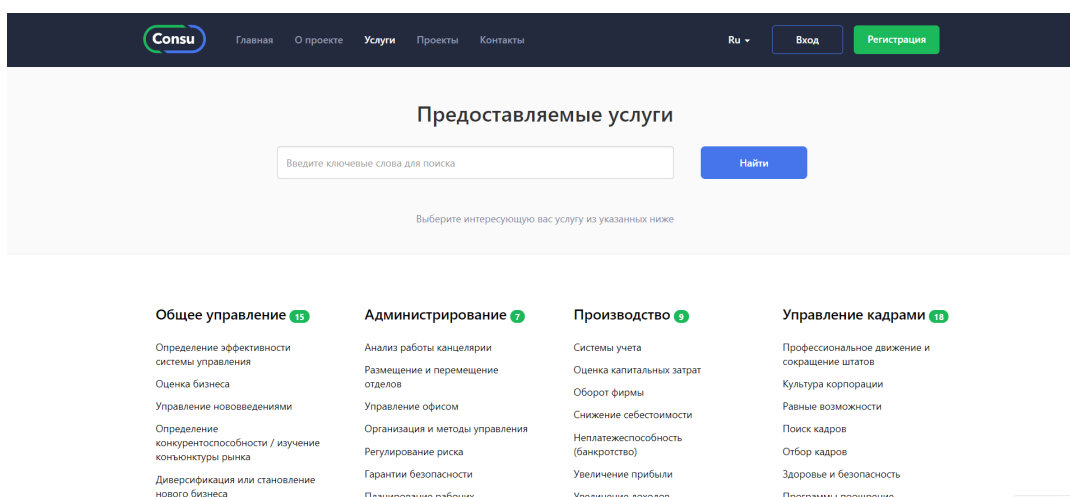


Рисунок 2 – Скриншот первого экрана страницы услуг

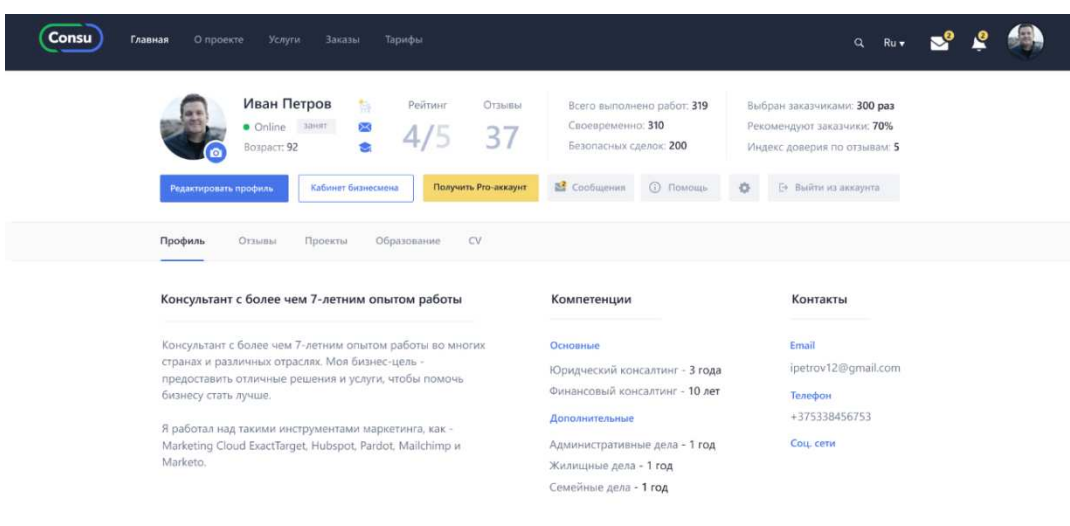


Рисунок 3 – Скриншот страницы профиля кабинета консультанта

При выборе цветов опирались на психологию цвета и особенности его восприятия пользователями.

Все иллюстрации были разработаны самостоятельно. В качестве основы выступали несколько скачанных иллюстраций, которые в последствии потерпели ряд изменений:

- некоторые иллюстраций были объединены между собой, для получения желаемого результата;
- в последствии все иллюстрации были перекрашены в цвета, используемые в фирменном стиле;
- некоторые иллюстрации были отрисованы самостоятельно.

Каркасы страниц создаются с помощью языка разметки HTML 5. Внешний вид описывается при помощи формального языка CSS. Адаптивность страниц создаётся с помощью фреймворка Bootstrap 3, который по умолчанию содержится в УИ2. Интерактив-

ность сайта создаётся с помощью языка программирования JavaScript и библиотеки jQuery, которые так же по умолчанию подключены в YII2.

База данных для проекта создавались в СУБД MySQL с использованием миграций и веб-приложения phpMyAdmin. В связи с тем, что изменение структуры базы данных часто требует изменение исходного кода, была использована так называемая возможность миграции баз данных, которая позволяет отслеживать изменения в базах данных при помощи терминов миграции баз данных, которые являются системой контроля версий вместе с исходным кодом.

Вся функциональная часть разработана на php-фреймворке Yii2.

В результате была разработана платформа, призванная решить проблему поиска профессиональных консультантов для представителей бизнеса и проблему поиска работы для консультантов.

ЛИТЕРАТУРА

1. <https://www.fl.ru>
2. <https://www.upwork.com>

УДК 004.9, 004.438

Студ. Н. Н. Красовская
Науч. рук. доц. О. А. Новосельская
(кафедра информатики и веб-дизайна, БГТУ)

ИНТЕРНЕТ-ПРИЛОЖЕНИЕ «КОНСАЛТИНГ ФРИЛАНС»

Более 80% мировых консультантов являются фрилансерами, что существенно затрудняет как поиск соответствующего требованиям и потребностям консультанта со стороны бизнеса, так и поиск работы для консультантов[1].

Создание интернет-приложения «Консалтинг фриланс» может помочь нивелировать существующие недостатки отрасли, а именно затруднение в поиске квалифицированных консультантов, поиск консультантов для решения определенной задачи соответствующих целям и бюджету бизнеса, централизация консультантов-поставщиков услуг и бизнес-покупателей услуг.

Консалтинг фриланс – интернет-приложение, которое позволит консультантам предложить себя бизнесам в качестве помощников в рамках своей области компетенций, а бизнесу – возможность найти профессионалов, которые решат актуальную проблему.

В настоящий момент на рынке восточной Европы нет профессионального агрегатора консалтинговых услуг. В основном, там пред-

ставлены только платформы электронной коммерции (далее маркетплейсы) одним из видов товара на которых представлены услуги, в которые входит и консалтинг.

Целью данного проекта является создание профессионального маркетплейса консалтинговых услуг, где консультанты/опытные сотрудники смогут предложить себя бизнесам/заказчикам в качестве помощников в рамках своей области компетенций, а бизнес сможет легко найти профессионалов, которые решат его проблему.

Для консультантов платформа даёт следующие возможности:

- наличие единого специализированного маркетплейса для предложения своих услуг;
- условия стабильны, условия сделок ясны – поставщик понимает, что получит гарантированную оплату за оказанные услуги при их выполнении;
- экономия времени и денег – нет необходимости вкладывать время и деньги в поиск клиентов, клиенты сами находят исполнителей благодаря платформе. Более того, клиенты в первую очередь обращаются к платформе, понимая высокое качество услуг.

Был проанализирован рынок консалтинговых услуг и бизнесов. По итогам анализа была определена целевая аудитория консультантов интернет-приложения:

1. Консультантами будут являться физические лица в возрасте от 28 до 70 лет, имеющие практический опыт профессиональных консультантов, с высшим образованием, желающие себя продвигать на рынке консалтинговых услуг.

2. Юридические лица, оказывающие консалтинговые услуги (малый и средний бизнес).

Авторизованный как консультант пользователь может:

1. Редактировать информацию своего профиля:

- общую информацию о себе;
- добавлять и редактировать образование;
- добавлять и редактировать резюме;
- добавлять и редактировать портфолио;
- редактировать контакты;
- изменить пароль.

2. Удалить свой профиль.

3. Изменять статусы (занят, принимаю заказы).

4. Просматривать оставленные ему отзывы (сортировать их).

5. Просматривать выполненные проекты (сортировать их).

6. Откликаться на заказы.

7. Общаться с заказчиком.

8. Получать уведомления о новых отзывах, заказах, одобренных откликах и сообщениях от заказчиков.
9. Оставлять отзывы бизнесам после завершения сделки.
10. Выйти из аккаунта.

Каждый пользователь будет видеть время последней активности другого, либо онлайн, если пользователь находится в интернет-приложении сейчас.

Во время разработки постоянно проводились различного рода анализы, опросы и анкетирование потенциальных клиентов, для выявления несовершенств платформы и способов их устранения.

Было изучено много статей про пользовательский опыт в дизайне. По итогам, были сделаны выводы, которые повлияли на концепцию дизайна платформы и последующее размещение элементов страниц.

Все макеты и прототипы платформы были разработаны в приложении Figma, которое позволило редактировать онлайн как макеты, так прототипы. При разработке учитывались правила и принципы дизайна. Итоговый макет приведен на рисунке 1.

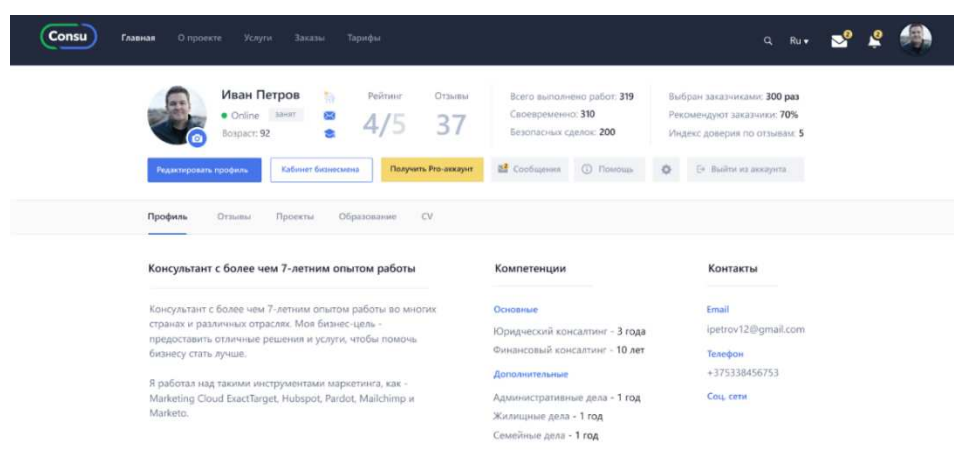


Рисунок 1 – Страница профиля кабинета консультанта

При выборе цветов опирались на психологию цвета и особенно-сти его восприятия пользователями.

Все иллюстрации были разработаны самостоятельно. В качестве основы выступали несколько скачанных иллюстраций, которые в последствии потерпели ряд изменений:

1. Некоторые иллюстраций были объединены между собой, для получения желаемого результата.
2. В последствии все иллюстрации были перекрашены в цвета, используемые в фирменном стиле.
3. Некоторые иллюстрации были отрисованы самостоятельно.

Вся функциональная часть разработана с помощью php-фреймворка Yii2. Каркасы страниц создаются с помощью языка разметки

ки HTML 5. Внешний вид описывается при помощи формального языка CSS. Адаптивность страниц создаётся с помощью фреймворка Bootstrap 3, который по умолчанию содержится в YII2. Интерактивность сайта создаётся с помощью языка программирования JavaScript и библиотеки jQuery, которые так же по умолчанию подключены в YII2.

База данных для проекта создавалась в СУБД MySQL с использованием миграций и веб-приложения phpMyAdmin. В связи с тем, что изменение структуры базы данных часто требует изменение исходного кода, была использована так называемая возможность миграции баз данных, которая позволяет отслеживать изменения в базах данных при помощи терминов миграции баз данных, которые являются системой контроля версий вместе с исходным кодом.

В результате было разработано интернет-приложение, призванное решить проблему поиска работы для консультантов.

ЛИТЕРАТУРА

1. Разговор с «матёрым» фрилансером: «Работать под пальмами неудобно – там песок и экран бликует»[Электронный ресурс] / Сайт IMAGURU. – Режим доступа: <https://imaguru.by/razgovor-s-matyorym-frilanserom-rabotat-pod-palmami-neudobno-tam-pesok-i-ekran-blikuet/>. – Дата доступа: 02.04.2019.

УДК 339.138

Студ. А.С. Байденко

Научн. рук. ст. преп. Н. И. Потапенко

(кафедра информатики и веб-дизайна)

РАЗРАБОТКА ИГРОВОГО ВЕБ-ПРИЛОЖЕНИЯ «РЕМОНТ КОСМИЧЕСКОЙ СТАНЦИИ»

Данный проект – это игра, в ходе которой игроку предлагается взять на себя управление человекоподобным роботом, задачей которого является восстановление питания на космической станции.

Сюжет. Действие игры происходит на исследовательской станции Оксима (рисунок 1). Станция была сооружена и экипирована для длительного исследования Второго Солнца, открытого в 8974 году. Автономность систем станции была нарушена вследствие ошибки в программном коде бортового компьютера, и космический комплекс не смог принять решение о своевременном возвращении на Землю. Аккумуляторы станции были перманентно испорчены вследствие длительного разряда. Резервный аккумулятор заряжался от солнечных

батарей последние 30 лет, так как Второе Солнце потеряло 80% своей мощности в первые 10 лет наблюдения.

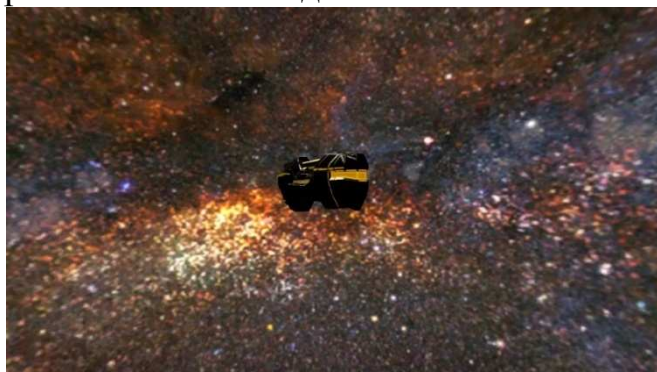


Рисунок 1 – Станция Оксима в космосе

Накопленной энергии было достаточно для того, чтобы запитать станцию на полчаса: зарядить робота NF-78, осветить станцию и открыть двери, чтобы можно было найти урановые стержни для подзарядки корабля, на котором этот робот должен покинуть станцию, чтобы вернуть на Землю батарею жёстких дисков с результатами исследований и сигналом бедствия.

Внешнее оформление. Для создания некоторых игровых ассетов была использована программа Houdini (Гудини), предоставляющая моушен-дизайнерам разнообразные средства наглядности и эффектов (рисунок 2). Это обосновано его процедурным подходом к работе. В отличие, например, от Blender или 3dsMAX Houdini не имеет деструктивных операций.

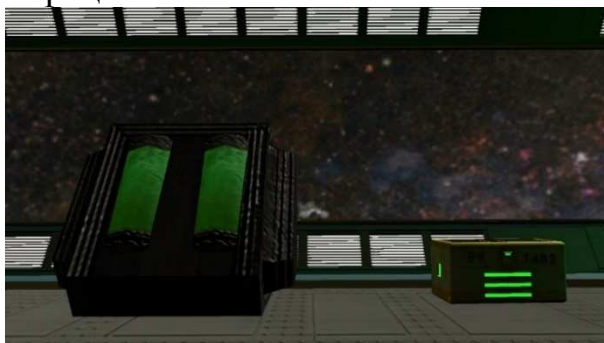


Рисунок 2 – Пример игрового ассета

Все операции могут быть изменены в любом времени в будущем. Каждая операция представлена нодой – небольшим блоком в отдельном экране, которая имеет ряд настроек. Ноды связываются по цепочке друг с другом образуя дерево. Ноды могут иметь несколько входов или выходов в зависимости от их назначения. При изменении размеров одного элемента модели все остальные будут меняться соответственно. Модель перестраивается в предусмотренных местах. Отверстия, ручки, ножки, панели – всё копируется нужное количество раз и распределя-

ется в нужных местах. Если модель стала длиннее, значит в нужных местах появятся дополнительные элементы. А если изменить элемент-родитель, то поменяются все элементы предки. Например, если речь идёт об операции копирования. Если работать в соответствии с таким подходом, то это позволяет принципиально менять финальную модель за пару кликов. Таким образом, процедурно сгенерировав один ящик, моделлер получает сотню разных ящиков. Один длиннее, другой выше, у третьего в два раза больше ручек, у четвёртого ручка одна, но большая, пятый имеет восемь прорезей, а шестой вообще открыт. И любой необходимый ящик можно получить из одной пользовательской ноды за несколько секунд. Всё это обеспечивает вариативность окружения без необходимости моделировать все вариации ящиков самостоятельно.

Ход игры. В ходе игры игрок может поднимать найденные капсулы с помощью телекинеза, притягивать их к себе, хранить их у себя в инвентаре, а также бросаться ими (рисунок 3).

При попадании первой капсулы в генератор, он загорается более ярко и открывает дверь в следующий отсек станции. При попадании в генератор третьей капсулы эвакуационный корабль становится заряженным и на нём можно покинуть станцию.

При посадке на корабль воспроизводится кат-сцена, в которой игра поздравляет игрока с прохождением игры (рисунок 4).



Рисунок 3 – Начало игры



Рисунок 4– Синяя капсула, парящая в воздухе благодаря телекинезу

Если игрок не успевает выполнить цель, то энергия на станции кончается и она отключается (рисунок 5).

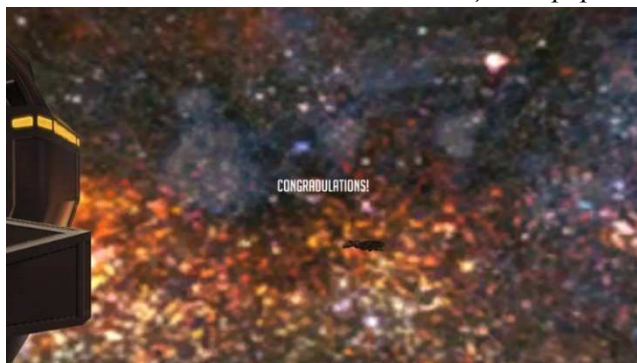


Рисунок 5 – Финальная кат-сцена

Планы по нововведениям. В ближайшем будущем в игру будет добавлен дополнительный уровень, на котором игроку предстоит управлять краном, смоделированным также в Houdini (рисунок 6).

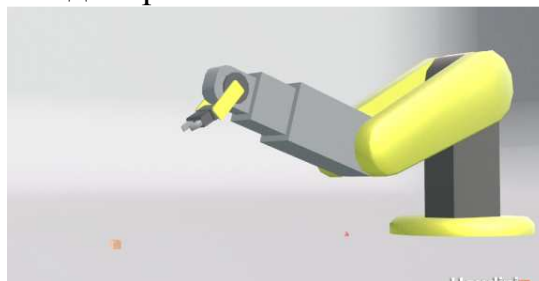


Рисунок 6 – Процедурно-смоделированный кран

Также в игру будет добавлено несколько механик передвижения, дополнительные сложности и опасности, чтобы сделать игру интереснее.

Игровые веб-приложения пользуются популярностью не только у подростков, но и у людей более старшего возраста. Игры, основанные на «умном» управлении, расчете стратегий и имеющие элементы опасности, будут несомненно популярны у потенциальных любителей игр.

УДК 339.138

Студ. В. С. Шаплыко

Науч. рук. ст. преп. Н. И. Потапенко
(кафедра информатики и веб-дизайна, БГТУ)

ТРЕНДЫ ДИЗАЙНА 2019

От качества дизайна зависит успех всего сайта, поэтому очень важно следить за трендом, понимать какие инструменты вошли в моду в сфере сайтостроения, а какие уже считаются устаревшими и остаются позади. Тренды чаще всего можно отнести к графическим ходам, которые кажутся свежими и неординарными. Зачастую такие ходы рождают крупные продуктовые бренды или агентства, а через некоторое время,

путем заимствования или переосмысления другими дизайнерами, такие графические ходы многократно повторяются в продуктах и концептах, что и порождает новый тренд. Также тренды в дизайне могут быть продиктованы развитием определенных технологий и желанием их применения в своих продуктах, таких как, например, использование дополненной и виртуальной реальности. Поэтому целью настоящей работы является определение наиболее актуальных технологий и образов, которые следует применять в современном веб-дизайне.

В первую очередь, тенденции в дизайне коснулись мобильных гаджетов. Мобильный трафик обогнал десктопные приложения, сервисы и сайты и составляет 57% [1]. Сегодня при разработке стало обязательным делать мобильную версию веб-ресурса. По данным Google мобильные версии сайтов наиболее посещаемы, поэтому мобильный дизайн стал тенденцией. Теперь дизайна сайта начинается с мобильной версии, а десктопная уже на втором месте. Сегодня, когда смартфоны и планшеты стали высокоинтеллектуальными, когда система распознает запрос голосом и появились голосовые боты, а производительность гаджета и размер диагонали его экрана позволяет включать в дизайн сайта абсолютно любые инструменты, для разработчиков появилось больше возможностей.

Иллюстрация – это основная тенденция, которая продолжала расти в последние годы. С целевых страниц, на целые веб-сайты, сделанные на заказ иллюстрация делает страницу уникальной и запоминающейся [2]. Хотя тенденция хороша для энергичного представления бренда, творческая абстракция подойдет и для более серьезных областей. Какой бы айдентика бренда ни была, для нее всегда можно создать уместный иллюстрированный образ.

Геометрические формы – это свежесть веб-дизайна, благодаря стандартным примитивным фигурам: круг, квадрат, треугольник. Такие формы очень хорошо реализуются в большинстве случаев, тем самым привнося более интересный вид стандартным элементам в дизайне, например, при оформлении информационных блоков или карточек товара, а также в качестве подложки к фото или иллюстраций. Все больше такая реализация встречается в качестве фонового изображения, иллюстрациях и паттернах. Чаще сейчас используют жидкие формы. Смешанные со смелыми цветами и интенсивными градиентами, плавные формы являются важной тенденцией веб-дизайна в 2019 году [3].

Иллюстративные значки – динамические линейные значки, которые люди естественно связывают с услугами, могут использоваться для установления визуальной иерархии и привлечения внимания к области, которую вы хотите, чтобы пользователь заметил [3]. Такие иконки при-

влекают внимание, быстро помогают пользователю понять свое назначение.

В этом году анимация вездесуща [3–5]. Вы можете найти их на веб-сайтах, плагинах или в дизайне приложений. Плавные переходы для кнопок или хороший эффект параллакса для разделов могут действительно изменить ситуацию, они передают много информации, которая в противном случае была бы потеряна. Движение куда красноречивее, чем освещение, позиционирование и материалы. Движение рассказывает историю. Таким образом, можно сделать логотип частью своей истории, вложив в него дополнительный смысл с помощью анимации.

3D графика в интерфейсах размывает границу между реальностью и цифровой анимацией [4]. Трехмерная графика является самым ярким трендом этого года. 3D персонажи, анимированные, мультяшные, рассказывающие истории, располагающие, будут чаще появляться на сайтах и рекламных роликах в 2019-ом. Эта тенденция будет особенно полезна для продуктовых компаний со сложными, не очень наглядными процессами. Используя 3D визуализацию, можно задействовать любой технологический процесс и создать более глубокий уровень понимания.

Комбинация реального 3D и CG выявила реакцию клиентов на реалистичные объекты, с которыми они могут взаимодействовать. Плоский дизайн также способен сделать это уникальным и еще не изученным способом. Его называют псевдо 3D. Это плоские слои, складывающиеся вместе, чтобы создать ощущение трехмерности. Основным ресурсом добавления размерности к плоскому дизайну, будет расположение теней и света, а также отражения. Существует также фальшивый 3D-тренд, который использует традиционные инструменты дизайна, такие как Principle и AfterEffects, для создания паттерна движения, эмулирующего 3D.

Микровзаимодействия – это небольшие анимации, используемые на мобильных платформах для создания тонких визуальных эффектов. Микровзаимодействия – важнейшая часть разработки интерфейсов. Они влияют на поведенческие факторы сайта [4, 6]. В дальнейшем будут также использоваться анимации интерфейса, реакции на какие-либо действия на сайт. Все для того, чтобы пользователь ощущал функционал на интуитивном уровне, а также получал удовольствие при взаимодействии с сайтом или интерфейсом приложения.

Традиционная модульная сетка не уйдет никогда, но дизайнерам хочется в обоих смыслах выйти за границы привычных форм и создать эффект недосказанности. Пользователям это нравится все больше

и больше. Разбитая сетка и перекрытия придают очень интересный визуал на фоне обычной верстки. Такая реализация очень хорошо подходит минималистичным решениям в дизайне [5]. Если же предполагается большое количество контента, стоит подумать, подойдет ли подобная сетка проекту. В целом тенденция есть, и она появилась на фоне того, что множество сайтов последнее время имеют стандартные решения и во многом идентичны.

Градиенты больше нужны не для привлечения внимания, они теперь приносят глубину и размерность в интерфейс. Новые экраны умеют фантастически передавать цвета. Поэтому дизайнеры исследуют границы доступности и пробуют использовать впечатляющие градиенты в интерфейсе. Градиент 2.0 утонченный и простой. Он не использует конфликтующие цвета. Яркие цвета никуда не денутся. Мы увидим больше комбинированных цветов и слоев. Фактически, даже монохромная палитра представит некоторую визуальную эстетику посредством глубины и размерности. Яркие цвета и значимые градиенты лучше выделяются на темном фоне. Лучший подход – это найти баланс между доступностью темного интерфейса и эмоциональным откликом ярких цветов [5].

Голосовой интерфейс реализует концепцию отсутствия пользовательского интерфейса в ее лучшем виде [6]. Дизайн не должен быть визуальным или видимым для работы. Инструменты не имеют большого значения, если мы не создаем тактильный опыт. Разработка логики означает операционную психологию. Создание голосового управления означает работу с обработкой естественного языка. Голосовой интерфейс настолько же чувственный, насколько и сложный. Этот процесс является внутренним и больше связан с копирайтингом, построением контекста и синтезом данных, чем с реальным дизайном. Несмотря на это, дизайнеры одержимы поиском способов представления голосового интерфейса. В основном это реализуется посредством впечатляющих нестандартных интерфейсов и анимации.

Все перечисленные тренды следует учитывать при разработке дизайна веб-ресурса и применять к его элементам. При этом не следует перенимать все абсолютно, а учитывать направленность и содержание разрабатываемого веб-сайта.

ЛИТЕРАТУРА

1. ТехТерра [Электронный ресурс]. – 2018. – Режим доступа: <https://texterra.ru/blog/trendy-dizayna.html> – Дата доступа: 12.04.2019

2. Impulse-design [Электронный ресурс]. – 2019. – Режим доступа: <https://impulse-design.com.ua/osnovnye-trendy-veb-dizajna-2018.html> – Дата доступа: 12.04.2018
3. Artweb [Электронный ресурс]. – 2019. – Режим доступа: <https://artweb.ua/ru/blog/view/20-trendov-veb-dizajna-v-2019-godu> – Дата доступа: 12.04.2018
4. Vc [Электронный ресурс]. – 2018. – Режим доступа: <https://vc.ru/design/50568-2019-god-trendy-v-dizayne> – Дата доступа: 12.04.2018
5. Designpub [Электронный ресурс]. – 2019. – Режим доступа: <https://designpub.ru/> – Дата доступа: 12.04.2018
6. Ux.pub [Электронный ресурс]. – 2018. – Режим доступа: <https://ux.pub/trendy-ui-i-ux-dizajna-na-2019-god/> – Дата доступа: 12.04.2018

УДК004.41

Студ. В. С. Бриль

Науч. рук. ст. преп. И. А. Миронов
(кафедра информатики и веб-дизайна, БГТУ)

ВЕБ-ПЛОЩАДКА ДЛЯ СБОРА ПОМОЩИ ТЯЖЕЛОБОЛЬНЫМ ДЕТЯМ «WAYSTONEHELP»

С каждым годом все больше и больше возрастает количество больных детей, которым не всегда могут помочь те методы лечения, которые существуют в Республике Беларусь. К счастью, количество детей, нуждающихся в лечении за рубежом, не очень много, но стоимость операций столь высока, что не позволяет родителям самостоятельно справиться с обрушившейся на них бедой. И именно благотворительные фонды помощи детям помогают им найти необходимые средства в борьбе со своим недугом.

Но сегодня рыночные отношения поставили фонды в такое положение, что они сами должны искать возможности для ведения своей деятельности. И чем лучше будет организована деятельность фонда по ее продвижению, тем быстрее она сможет собрать деньги на осуществление своей миссии. И чем больше форм активности по привлечению благотворительных средств будет использоваться, тем лучше она будет работать.

С развитием Интернета появилось огромное множество новых инициатив по организации помощи. Пожалуй, нет другой технологии, которая дала бы больший импульс развитию благотворительности, чем информационные сети. И самое важное то, что желание помогать нуж-

дающимся стало приобретать массовый и повсеместный характер. Поэтому усовершенствование и разработка новых инструментов продвижения благотворительных организаций в сети Интернет, которые позволят охватить еще большую аудиторию заинтересованных людей в оказании помощи, чем имеется сегодня, – главная задача таких организаций.

Всеми этими вопросами занимается фандрайзинг, который как раз обеспечивает организацию надежными потоками доходов. В Республике Беларусь понятие фандрайзинга медленно, но верно начинает набирать обороты.

Поэтому, выступая от лиц благотворительных организаций, предлагаю реализовать социальную платформу по сбору средств, для тяжелобольных детей в Интернете.

Каждый благотворитель выбирает наиболее удобный и интересный для него способ жертвовать. Поэтому мы предлагаем еще один способ фандрайзинга для благотворительных организаций и еще один способ пожертвования, который придется многим благотворителям по душе.

Данный социальный проект предполагает создание международной платформы, которая позволяет:

- благотворительным организациям публиковать информацию о ребенке, которому необходима транспортировка за границу, чтобы пройти лечение или курс реабилитации;

- благотворителям приобретать мили за свои личные средства или жертвовать часть накопленных миль своих авиаперевозчиков.

Описание потенциальной целевой аудитории:

- частные благотворители (платежеспособное население, которое готово оказывать спонсорскую помощь фондам);

- участники корпоративных программ авиакомпаний (мужчины и женщины, регулярно пользующиеся услугами авиаперевозчиков).

Проект Waystohelp – уникальный для Беларуси. Поэтому главная задача для нас на этапе продвижения – это рассказать о нем пользователям сайта, благотворителям и постоянным партнерам. Для этого мы вынесли всю необходимую информацию на главную страницу сайта сразу же после блока «кому нужна помощь».

Стилистика портала выполнена в голубом цвете решения. По исследованиям британских ученых, голубой цвет – это один из популярных цветов. Этот цвет является любимым для 57% мужчин и 35% женщин, а еще его могут видеть люди, страдающие дальтонизмом. Мы взяли его за основу еще по одной причине: голубой цвет дарит ощущение свободы, продуктивности и доверия. Именно поэтому его

используют организации здравоохранения, энергетики, финансов, самолетов, техники. Так как главная миссия организации – помогать детям, а главная миссия проекта – убеждать людей делиться милыми, то нашим дизайнерским решением мы смогли решить поставленные задачи.

При попадании на главную страницу сайта пользователь сразу же увидит две кнопки: авторизацию и регистрацию. При нажатии на одну из этих кнопок, пользователю будет необходимо заполнить форму, внося туда свои данные.

На первом блоке главной страницы размещён баннер с детьми, которые нуждаются в помощи, на данном баннере размещена информация о ребёнке и имеется кнопка «хочу помочь», при нажатии на которую пользователь сможет пожертвовать мили для ребёнка. Данный баннер представлен на рисунке 1

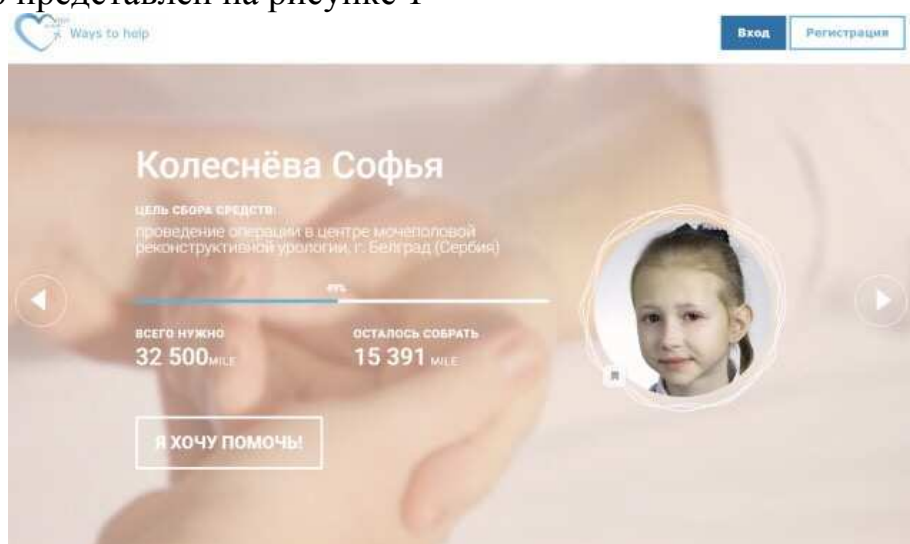


Рисунок 1 – Баннер

Во втором блоке также представлены дети, нуждающиеся в помощи, и имеется кнопка «хочу помочь». Пользователь может нажать на нужного ему ребёнка и тем самым перейдёт на страницу (рисунок 2), где будет предоставлена вся информация о выбранном им ребёнке.

На главной странице размещена информация о данной платформе, для того чтобы случайный пользователь понимал куда он попал. И блок с новостями, из которых можно более подробно узнать о благотворительности, а также найти информацию о детях, которым уже была оказана помощь. В подвале сайта можно найти информацию, с помощью которой можно связаться лично и уточнить все интересующие вопросы.

Также необходимо привлечь аудиторию и в дальнейшем извлекать из прибывшей аудитории выгоду, для этого требуется, произвести рекламные действия.

Дети, которым нужна помощь



Рисунок 2 – Баннер

Реклама очень важна, так как она позволяет приводить посетителей, которые дальше смогут превратиться в клиентов.

Рассмотрим список возможных каналов распределения. Во-первых, так как наше приложение тесно связано с авиаперевозками, то первое, что приходит на ум это бортовой журнал. Для примера мы рассмотрели бортовой журнал авиакомпании «Белавиа» – OnAir. Данный журнал выпускается раз в месяц, имеет большой объём и огромное количество рекламы. Мы считаем, что размещение информации о партнёрской программе здесь позволит охватить потенциальных потребителей, которые пользуются услугами авиакомпаний.

Во-вторых, сайты благотворительных организаций, которые будут размещаться на нашей фандрайзинговой площадке. Они являются одними из наиболее заинтересованных лиц в том, чтобы как можно больше людей узнали о ещё одном способе пожертвования.

И как ещё один канал распределения мы выбрали запуск контекстной рекламы, которая перенаправляет посетителей на сайт, что как следствие, повлечёт за собой новых жертвователей.

УДК 004.41

Студ. С. А. Кузьмина
Науч. рук. ст. преп. И. А. Миронов
(кафедра информатики и веб-дизайна, БГТУ)

ВЕБ-ПЛОЩАДКА «PICKOFFPIC»

Многие талантливые художники прошлого жили впроголодь и довольствовались малым. Их талант был оценен обществом только после смерти и во многом благодаря предприимчивым арт-дилерам, которые искусственно раздували стоимость работ. Пожалуй, самый показательный пример это Винсет Ван-Гог. При жизни ему удалось продать всего лишь одну картину. Свою жизнь он закончил в клинике для душевных больных, где покончил жизнь самоубийством. А после смерти стоимость его произведений взлетела в разы, благодаря расчетливым немецким дилерам.

Но к счастью у большинства художников жизнь складывалась не так трагично. Многие зарабатывали себе на хлеб обычными мирскими профессиями: клерками, кондукторами, поварами. К примеру, Поль Гоген плавал на торговых судах, служил в военном флоте, а после службы устроился на работу биржевым брокером, что обеспечило ему солидный доход. Свою страсть к искусству он обнаружил в зрелом возрасте, уже имея за плечами солидный капитал.

Но были примеры художников, чей талант оценили современники. Яркий пример – Рубенс. Еще при жизни он получил должность придворного живописца. Многие знатные и богатые люди, того времени считали за честь заказать свой портрет у мастера.

Талант, удача, усердие, жизненные обстоятельства вот из чего складывается успех. Таким образом, можно сделать вывод, что путь к славе художников прошлого, мало чем отличается от нынешних творцов.

Есть несколько вариантов продвижения и продажи картин:

- личные контакты и связи;
- арт-дилеры;
- интернет-площадки.

Стоит уделить внимание последнему способу. Речь идет о различных интернет аукционах и галереях. Большинство арт-галерей предлагают бесплатную регистрацию и размещение ваших работ. Зарабатывают они на процентах от продаж до 50%. Конечно у этого способа есть недостатки, из-за многообразия выбора картина может потеряться.

Однако, продажа картин через интернет несет в себе и массу преимуществ по сравнению с традиционными способами реализации предметов искусства. Прежде всего это – широкий охват целевой аудитории.

Как правило, такие сайты посещают люди, напрямую заинтересованные в приобретении произведения конкретной категории и жанра, а потому шансы авторов продать свое творение в разы возрастают.

Экономия времени, так как люди искусства постоянно находятся в творческом поиске или заняты процессом создания очередного шедевра, то времени для разъездов по галереям с целью договориться о размещении своего творения у них практически не остается. Интернет-площадка избавляет художников от такой необходимости. Все, что требуется – это пройти элементарную процедуру регистрации на сайте, предварительно сфотографировать свои работы и выставить их на обозрение посетителей портала.

Рациональный расход средств. Большинство галерей, как известно, взимают с художников плату за размещение в стенах выставочных залов предметов искусства. При продаже работ в интернете необходимость платы за презентацию отсутствует. Портал взимает с авторов произведений, согласно заключенному договору, незначительную комиссию непосредственно в момент продажи произведения искусства.

PickofPic – это портал для художников, который позволяет:

- художникам размещать и продавать свои картины;
- ценителям искусства находить и покупать картины;
- арт-дилерам находить новые таланты.

В основе его дизайна заложены основные принципы минимализма, который подчеркивает уникальность каждого продукта, представленного на сайте.

Зайдя на главную страницу можно увидеть три основных раздела: картины, художники и новостная лента.

В разделе «Картины» представлены картины всех пользователей. Для удобства поиска интересующей картины на странице присутствует сортировка по различным критериям. Все выполнено в стиле минимализм, поэтому первое что привлекает внимание – это сами картины. Данный раздел представлен на рисунке 1. Более подробная информация о картине находится на странице с ней, там же находится кнопки «Хочу купить» и «Связаться с художником». Нажав первую кнопку, художник узнает, что вы заинтересованы конкретно этой картиной и свяжется с вами. Но также вы можете связаться с художником для других целей, например, хотите авторскую картину по своей задумке, и вам очень нравится, как работает конкретный художник.

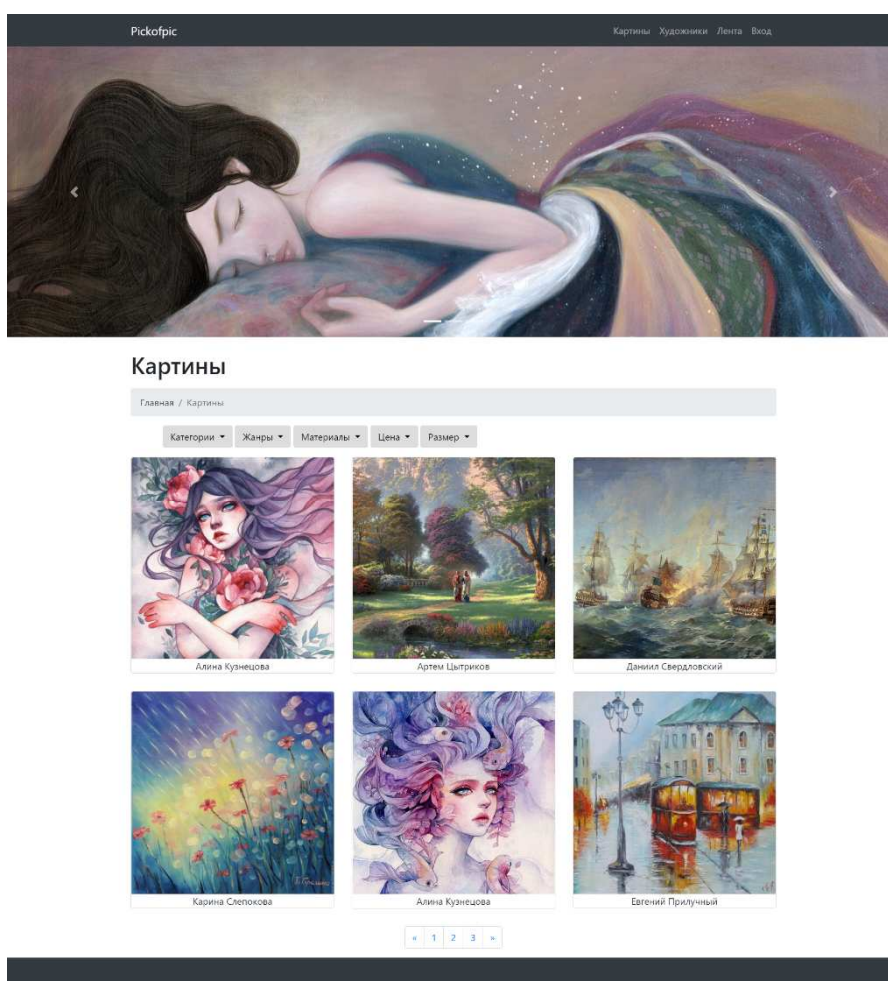


Рисунок 1 – Раздел сайта – «Картины»

Во втором разделе «Художники» можно найти интересующего вас художника. Перейти в этот раздел можно с главной страницы (рисунок 2). Также вся навигация присутствует в шапке сайта. Страница выполнена в той же стилистике что и раздел с картинами, отличие в том, что там не сортировка присутствует, а поиск по имени. Выбрав нужного художника, вы попадете на его личную страницу, где представлены его картины.

Художники



Еще

Рисунок 2 – Ссылка на раздел с художниками

Ну и остается последний раздел – это «Новостная лента». Там будут показаны последние новости. А именно: кто зарегистрировался новый, кто выложил новую картину, какие-то посты об искусстве от администратора. В дальнейшем этот раздел можно расширить. Например, можно сделать чтобы сами пользователи смогли выкладывать не только картины, но и посты писать. Или пойти дальше и сделать не рандомную ленту, а по подпискам.

Проект Pickofpic может и не уникальный, но точно станет востребованным. Особенно примечательна его масштабируемость. Он может разрастись до полноценной социальной сети, то есть до весьма перспективной площадки с будущим.

УДК 004.056+003.26

Студ. К.М. Процукович
Науч. рук. проф. П. П. Урбанович
(кафедра информационных систем и технологий, БГТУ)

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМИЧЕСКИХ СВОЙСТВ КРИПТОВАЛЮТНЫХ ТЕХНОЛОГИЙ

Ниже будет проведен сравнительный анализ наиболее широко распространенных криптовалютных платформ, и, в частности, мы исследуем Bitcoin, Ethereum и Ripple. Проанализируем консенсусные и стимулирующие механизмы, используемые различными протоколами. Затронем некоторые аспекты безопасности [1-4].

Криптографическая валюта *Bitcoin* [3] впервые популяризировала концепцию Blockchain. Вплоть до того, как Bitcoin и его распределенная глобальная бухгалтерская книга были изобретены, цифровые активы полностью управлялись централизованными органами, отслеживающими все транзакции.

Ethereum – одна из технологий Blockchain второго поколения, целью которой является создание новой модели для построения децентрализованных приложений.

Ripple является типом консенсус-ориентированной распределенной базы данных. Эта система не является технологией Blockchain, поскольку транзакции не добавляются в виде блоков, которые связаны друг с другом хеш-цепью. Однако, подобно моделям согласования блоков, сеть Ripple обновляет регистр посредством согласованного процесса между узлами [1-2].

Технология Blockchain позволяет организациям осуществлять транзакции напрямую, без необходимости отправлять предложенные транзакции централизованной третьей стороне, действующей в каче-

стве доверенного посредника. Каждый узел хранит отдельную копию цепочки блоков, состояние которой обновляется по мере добавления новых блоков. Однако узлы могут иметь совершенно непоследовательное представление данных, записанных в цепочке блоков, из-за расходящегося порядка, в котором транзакции перечислены в репликах. Поэтому участникам необходимо координировать действия друг с другом, чтобы определить законный регистр и гарантировать согласованность системы на всех узлах.

В таблице 1 приведены для сравнения данные о механизмах подтверждения рассмотренных технологий.

Proof-of-Work (PoW – дословно: доказательство работы) — алгоритм защиты главной особенностью заключается в асимметрии затрат времени решения — они значительны на нахождение решения и весьма малы для проверки.

Схема Bitcoin PoW заложила теоретическую основу для современных консенсусных моделей, и все еще появляются новые, обеспечивающие инновационные функции и свойств.

Таблица 1 – Сравнительный анализ механизмов подтверждения

	Bitcoin	Ethereum	Ripple
Метод верификации	PoW	PoW (Ethash)	Консенсус
Вовлеченные узлы	Все	Все	Валидаторы
Алгоритм хеширования	SHA-256	Кеccak-256	SHA-512Half
Время на подтверждение блока	10 минут	14 секунд	5-10 секунд
Скорость транзакции	Низкая	Низкая	Высокая
Награда майнерам	?	?	X
Комиссия за перевод	?	?	?

Хотя доказательство, используемое в Ethereum, аналогично протоколу Bitcoin PoW, в Ethash он использует другой криптографический примитив для своей функции хеширования, называемый Кеccak-256, вместо того чтобы полагаться на алгоритм двойного хеширования SHA-256.

Однако есть модели, которые используют другой подход для демонстрации согласованности транзакций. Среди них Ripple, которая выбрала схему голосования, называемую консенсусом. Цель состоит в том, чтобы все узлы могли договориться о том, какие транзакции следует включить в последнее закрытие реестра.

Поскольку в Ripple отсутствует процесс решения криптографической загадки с подтверждением работы, консенсус быстр, и регистры проверяются за считанные секунды. Поскольку новая транзакция закрывается примерно каждые 5 секунд. Это значительно отличается от времени блокировки Bitcoin. Действительно, новый блок генериру-

ется примерно каждые 10 минут. Сеть Ethereum, для сравнения, производит блок в среднем каждые 14 секунд.

Важной функцией в поддержании неизменности системы Blockchain является стимулирование. Поскольку процесс консенсуса требует сотрудничества между участниками с неприсоединившимися интересами. Bitcoin включает в себя механизмы стимулирования, которые приходят в форме вознаграждений за майнинг и комиссионных за транзакции. Первый узел, который успешно решает PoW и получает возможность добавить свой блок в цепочку блоков, может получить *вознаграждение* за блок.

Подход, принятый Ripple, полностью отличается от только что описанных моделей, поскольку он не обеспечивает прямого денежного вознаграждения за поддержку узлов. Причина такого отсутствия механизма стимулирования заключается в том, что в Ripple не используется процесс майнинга.

По замыслу, как только транзакция добавлена в цепочку блоков и подтверждена, она никогда не может быть отменена. Однако, несмотря на то, что целостность данных является одним из ключевых моментов Blockchain, они не являются неуязвимыми для кибератак, потому что законные изменения в глобальной записи могут быть сделаны в принципе любым.

Таблица 2 – Сравнительный анализ аспектов безопасности

	Bitcoin	Ethereum	Ripple
Проблема двойной траты	?	?	?
51% атаки	X	X	?
Атака «отказ в обслуживании»	?	?	?

Проблема двойной траты относится к случаю повторная продажа одних и тех же активов. Bitcoin решает эту проблему путем хронологического упорядочения блоков транзакций в непрерывную цепочку проверок работы, которая видна всем пользователям. Точно так же Ethereum использует свой алгоритм на основе PoW, чтобы предотвратить риск того, что пользователь может одновременно потратить одну и ту же единицу валюты в нескольких транзакциях.

Ripple предлагает альтернативное решение проблемы двойных расходов через процесс консенсуса. Так как процесс требует согласования порядка транзакций, если две транзакции представляют собой двойные затраты, атака решается простым согласованием того, какая из двух транзакций будет первой (другая считается недействительной и, следовательно, не применяется).

В нынешних условиях дизайн Blockchain по своей природе уязвим для атаки на 51%. Если злоумышленник контролирует более 50% мощ-

ности майнинга, он может создать независимую ветвь. Таким образом, если эта атака успешна, злоумышленник может манипулировать реестром в свою пользу. Эмпирические данные показывают, что эта атака неосуществима для любого отдельного пользователя, поскольку для ее перерасчета все доказательства для всех предыдущих блоков в цепочке потребуют огромных вычислительных ресурсов. Однако, хотя сам Bitcoin является чисто децентрализованным, снижение стимула для майнинга приводит к централизации функции майнинга. Утверждают, что эта тенденция к централизации увеличивает риск атаки на 51%. Ethereum обладает теми же слабостями, что и Bitcoin, в отношении атаки на 51%.

Ripple, как известно, не полагается на распределенную вычислительную мощность для защиты целостности сети. Он заменяет голосование за вычислительную мощность понятия майнеров о механизмах консенсуса на основе PoW на голосование валидатор. Основное предположение заключается в том, что большинство узлов Ripple не будут вступать в сговор с целью манипулирования результатом голосования. На самом деле, если 80% проверяющих серверов вступают в сговор, можно подтвердить мошенническую транзакцию. Однако в случае, если большинство валидаторов становятся вредоносными, они могут переписать всю историю транзакций системы.

Наконец, поскольку технология Blockchain основана на общедоступном реестре информации, поддерживаемой сетью компьютеров по всему миру, злоумышленники могут транслировать большое количество спама в транзакциях, пытаясь нарушить нормальную работу сети. Наиболее заметным последствием такой атаки является создание избыточной нагрузки на сеть [5], что вызывает трудности в обработке законных транзакций.

Чтобы смягчить атаки типа «отказ в обслуживании», Ripple вводит плату за транзакцию. Это создает сильный сдерживающий фактор против спама в журнале, потому что любые атаки, направленные на потерю пропускной способности сети, становятся очень дорогими для злонамеренных агентов

Мы рассмотрели консенсусные и стимулирующие механизмы, а также некоторые аспекты безопасности трех платформ: Bitcoin, Ethereum и Ripple. Bitcoin является первой полноценной криптовалютой, в некоторых аспектах он отстает от аналогов, в связи с тем, что конкуренты предлагают решение тех проблем, с которыми Bitcoin справиться не в состоянии, будь то атака 51% или скорость транзакций, но

даже так он всё же остается самой капитализированной и популярной криптовалютой в мире.

ЛИТЕРАТУРА

1. Ethereum Wiki, Ethash. [Электронный ресурс]. – Режим доступа: <https://github.com/ethereum/wiki/wiki/Ethash>. – Дата доступа: 15.04.2019.
2. Schwartz, D., Youngs, N. and Britto, A. 2014. The Ripple Protocol 2.Consensus Algorithm. White Paper. [Электронный ресурс]. – Режим доступа: https://ripple.com/files/ripple_consensus_whitepaper.pdf. – Дата доступа: 15.04.2019.
3. Nakamoto, S. 2008. Bitcoin: A peer-to-peer electronic cash system. [Электронный ресурс]. – Режим доступа: <https://bitcoin.org/bitcoin.pdf>. – Дата доступа: 15.04.2019.
4. The free encyclopedia Wikipedia. [Электронный ресурс]. – Режим доступа <https://en.wikipedia.org/wiki/Bitcoin>. — Дата доступа: 15.04.2019.
5. Урбанович, П. П. Компьютерные сети: учебное пособие для студентов высших учебных заведений по техническим специальностям / П. П. Урбанович, Д. М. Романенко, Е. В. Кабак. - Минск: БГТУ, 2011. – 399 с.

УДК 004.056+003.26

Студенты К. В. Клицунова, Е. И. Дубовик
Науч. рук. проф. П. П. Урбанович
(кафедра информационных систем и технологий, БГТУ)

СТЕГАНОГРАФИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРНЫХ СЕТЯХ

Некоторые основные понятия из предметной области [1,2], которые будут использованы в докладе.

Стеганография — это наука о скрытой передаче информации путём сохранения в тайне самого факта передачи.

Стеганографическая система – объединение методов и средств, используемых для создания скрытого канала для передачи информации.

Контейнер – любая информация, используемая для сокрытия тайного сообщения.

Стеганографический канал – канал передачи стегоконтейнера.

Ключ – секретная информация для сокрытия стегоконтейнера и извлечения информации из него.

Сообщение — контейнер с осажденной тайной информацией.

Цель нашей работы: проанализировать практические аспекты методов сетевой стеганографии.

Сетевая стеганография представляет собой группу методов, в которых скрытая информация передается через компьютерные сети с использованием особенностей работы протоколов передачи данных.

Методы сетевой стеганографии включают изменение свойств одного из сетевых протоколов. Так же, может использоваться взаимосвязь между двумя или более различными протоколами с целью более надежного сокрытия передачи секретного сообщения. Под *сетевой стеганографией* на сетевом уровне модели OSI [3] (далее ip-стеганография), понимается группа методов сетевой стеганографии, в которых стегоконтейнером могут являться неиспользуемые места в заголовках и полях данных ip-дейтаграмм.

Метод DF – это метод, основанный на модификации полей «identification» и «fragment-offset» при установленном флаге «DF» (don't fragment – не фрагментировать).

Если установлен флаг DF, модуль IP не станет фрагментировать дейтаграмму. Если IP-дейтаграмма была фрагментирована, то каждый фрагмент становится отдельным пакетом со своим собственным IP-заголовком. Такие пакеты маршрутизируются независимо, и, как следствие, фрагменты дейтаграммы могут приходить в точку назначения с нарушением их очередности.

Однако в IP-заголовках фрагментов содержится вся необходимая информации для их правильной сборки в пункте назначения. Фрагментация в IP выполняется независимо от транспортного уровня модели OSI. Несмотря на такую «прозрачность», фрагментация может привести к нежелательным последствиям, которые сказываются на уровнях выше IP.

Дело в том, что из-за потери одного фрагмента потребуется передать повторно всю дейтаграмму, а поскольку в самом протоколе IP не предусмотрены таймаут и повторная передача, то эти функции должны быть возложены на более высокие уровни. Протокол TCP осуществляет повторную передачу по таймауту, а UDP — нет. Если окажется, что потерян некоторый фрагмент сегмента TCP, то по таймауту будет повторена передача всего сегмента TCP. Повторная передача отдельного фрагмента ip-дейтаграммы невозможна в принципе. Действительно, если фрагментацию произвел не хост источника дейтаграммы, а один из промежуточных маршрутизаторов, то источник не может знать, каким именно образом было выполнено разбиение на фрагменты. Уже по одной этой причине желательно

принимать меры для предотвращения фрагментации. Фрагментация пакетов в IP является штатной ситуацией, поэтому использование ранее указанных полей заголовков ip-пакетов в качестве стекоконтейнера является вполне целесообразным методом для организации скрытого канала связи (рис.1).

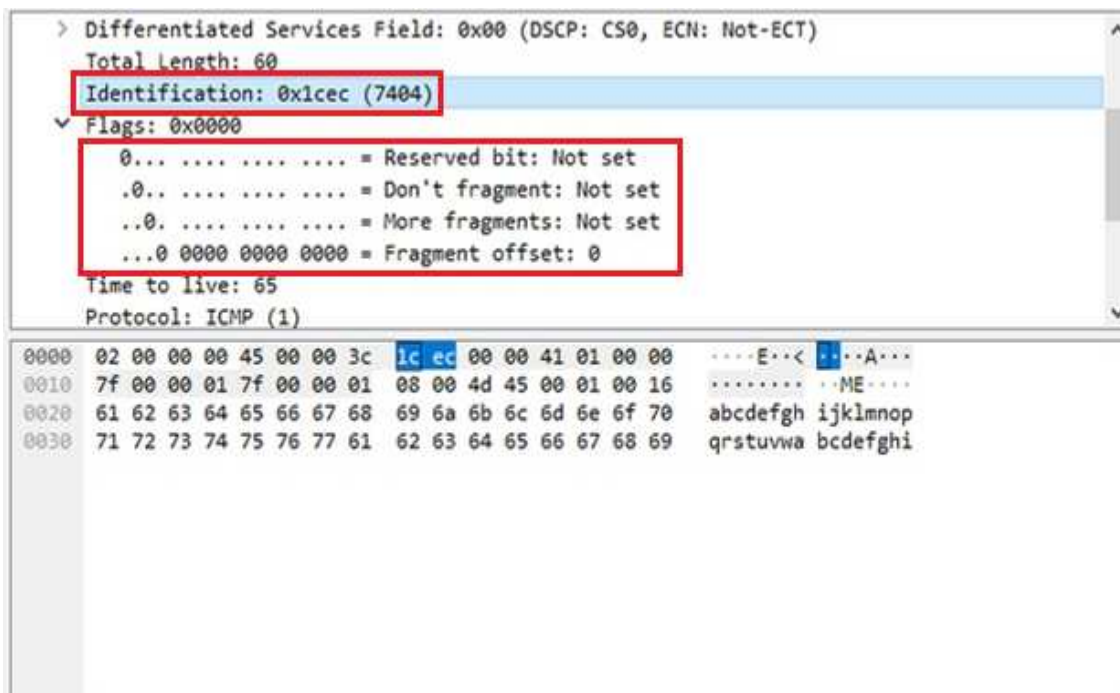


Рисунок 1 – Заголовки IP-дейтаграммы, используемые в качестве контейнера

Важно понимать, что при правильном выборе размера пакета возможно добиться выполнения условия $L \leq PMTU$, где L – длина пакета; MTU – maximum transmission unit (максимальный размер полезного блока данных одного пакета); $PMTU$ – MTU трассы от источника до конечного адресата пакета; $PMTU = \min MTU_i$, где $\min MTU_i$ – минимальное значение MTU среди интерфейсов маршрута, по которому пройдёт пакет.

При выполнении условия ($L \leq PMTU$) необходимость фрагментации пакетов на интерфейсах маршрута отсутствует. Это означает, что поля «identification», «flags», «fragmentoffset» обрабатываться промежуточными маршрутизаторами не будут. В таком случае, при использовании этих полей в качестве стекоконтейнера информация, встроенная в него, будет передана без изменений. Ёмкость такого стекоконтейнера может составлять до 32 бит.

Для исследования метода DF выбрана программная реализация утилиты *ping* (утилита отправляет запросы (ICMP Echo-Request) протокола ICMP указанному узлу сети и фиксирует поступающие ответы (ICMP Echo-Reply)). Так как частой практикой является непрерывная

отправка запросов для проверки соединения, большое количество ICMP-пакетов не вызовут подозрения.

В программной реализации утилиты имеются переменные, соответствующие всем полям заголовка. При помощи данных переменных можно модифицировать значения полей.

В качестве стеганоконтейнера было выбрано поле «identification».

Стеганосообщением является текстовый документ с тайной информацией.

Для реализации данного метода требуется:

- изменить значение поля «fragment-offset» на 0x00 (don't fragment),
- считывать информацию из документа,
- помещать значение считавшегося символа в поле «identification»,
- производить отправку запроса (см. рис.2).

```
const char text[20] = "Tqwepasdfghjsdfghbk"; // сообщение

for (int i = 0; i < 20; ++i)
{
    IpHead.id = (int)text[i]; // заносим символ в поле идентификатора
    memcpy(FullPack, &IpHead, sizeof(IpHeader));
    memcpy(FullPack + sizeof(IpHeader), Packet, icmp_size);

    int bytes = sendto(listn, (char*)PACKET_, size, 0, (sockaddr*)&list_adr, sizeof(list_adr)); // отправляем пакет
}
```

Рисунок 2 – Фрагмент кода, отвечающий за занесение информации и отправку пакета

На рис. 3 можно видеть пример пакета с осажженным символом.

Два последних пункта выполняются до тех пор, пока вся считанная информация не будет отправлена.

Для получения секретной информации стороне получателя потребуется специализированный софт для «отлова» пакетов.

Исследуя метод DF, была использована программа *Wireshark*, которая позволяет «отловить» полученные пакеты и считать в полях заголовков нужную информацию.

Выводы: в ходе исследовательской работы мы освоили технологию сетевой стеганографии, а также на практике реализовали один из методов скрытной передачи информации в сети.

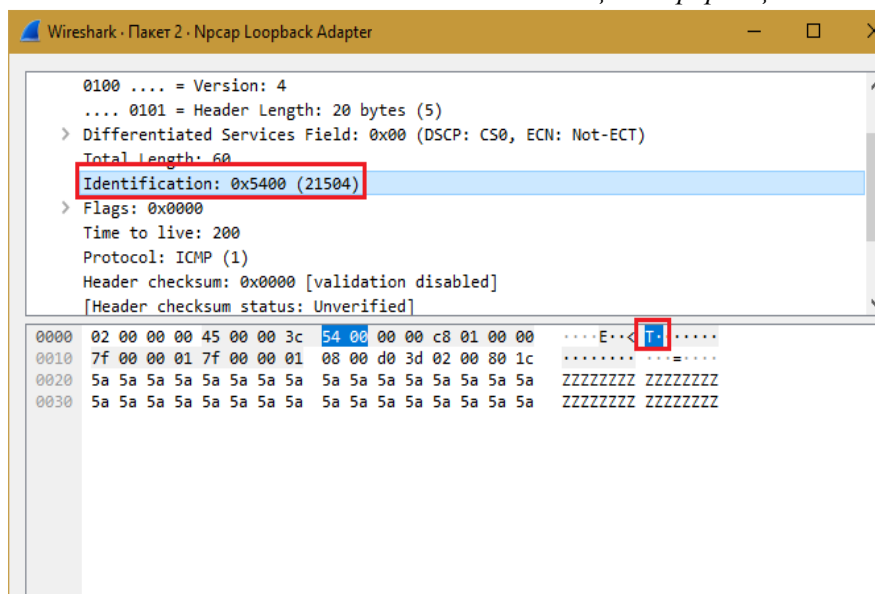


Рисунок 3 – Пакет с осажденным символом “Т”

Стеганографические методы могут применяться для решения других задач [5-6].

ЛИТЕРАТУРА

1. Урбанович, П. П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие для студ./ Урбанович П.П. – Минск: БГТУ, 2016. – 220 с.
2. Урбанович, П. П. Информационная безопасность и надежность систем: учебно-методическое пособие по одноименному курсу для студентов специальности 1-40 01 02-03 "Информационные системы и технологии" / П. П. Урбанович, Д. М. Романенко, Е. В. Романцевич. – Минск: БГТУ, 2007. – 87 с.
3. Урбанович, П. П. Компьютерные сети: учебное пособие для студентов высших учебных заведений по техническим специальностям / П. П. Урбанович, Д. М. Романенко, Е. В. Кабак. – Минск: БГТУ, 2011. – 399 с.
4. Internet protocol – DARPA Internet Program Protocol Specification/ RFC-791 USC/Information Sciences Institute, September 1981. [Электронный ресурс]. – Режим доступа: <https://tools.ietf.org/html/rfc791>. – Дата доступа: 13.04.2019.
5. Urbanovich, P. Theoretical Model of a Multi-Key Steganography System / P. Urbanovich, N. Shutko // Recent Developments in Mathematics and Informatics. Contemporary Mathematics and Computer Science. Vol. 2, Chapter 11. – Lublin : KUL, 2016. – P. 181-202.
6. Text steganography application for protection and transfer of the information / Pavel Urbanovich, Konstantin Chourikov, Andrey Rimorev, Nadzeya Urbanovich // Przegląd elektrotechniczny. – 2010. – R. 86.– № 7.– P. 95-97.

ОСОБЕННОСТИ ДИФФЕРЕНЦИАЛЬНОГО И ЛИНЕЙНОГО КРИПТОАНАЛИЗА

Криптоанализ – это наука, изучающая математические методы нарушения конфиденциальности и целостности информации, также включает методы выявления уязвимости криптографических алгоритмов и протоколов. На практике криптоанализ основан на 3 вещах:

- изучение системы шифрования в целом,
- изучение особенностей исходного текста,
- изучение особенностей ключевой системы.

Рассмотрим основные виды атак [1,2].

Пассивная атака – это перехват и прослушивание передаваемых по сети данных, т.к. атакующий не воздействует на протокол, алгоритм, ключ, само сообщение, какие-либо части системы шифрования.

Активная атака – это изменение сообщений, изменение системных файлов, попытки выдать себя за другого человека. К основным видам атак относятся следующие.

1) Атака «только шифротекст» (cipher-only attack): при выполнении атаки этого типа, атакующий имеет шифротекст нескольких сообщений. Каждое из сообщений зашифровано одним и тем же алгоритмом.

2) При выполнении атаки типа «известный открытый текст» (known-plaintext attack), у атакующего есть открытый текст и соответствующий ему шифротекст одного или нескольких сообщений.

3) При выполнении атаки типа «выбранный открытый текст» (chosen-plaintext attack), у атакующего также есть открытый текст и соответствующий ему шифротекст, но он имеет возможность самостоятельно выбирать открытый текст и получать его в зашифрованном виде.

4) При выполнении атаки типа «выбранный шифротекст» (chosen-ciphertext attack), атакующий может выбирать шифротекст для расшифрования и имеет доступ к получаемому в результате открытому тексту.

Дифференциальный криптоанализ разработан в 1990 году израильскими криптографами Э. Бихамом (E. Biham) и А. Шамиром (A. Shamir) [3]. Первая такая атака была проведена в 1990 году против алгоритма DES. Дифференциальный криптоанализ — атака с подобранным открытым текстом. Это означает, что для применения ДК вы должны иметь возможность зашифровать абсолютно любые тексты в абсолютно любом количестве.

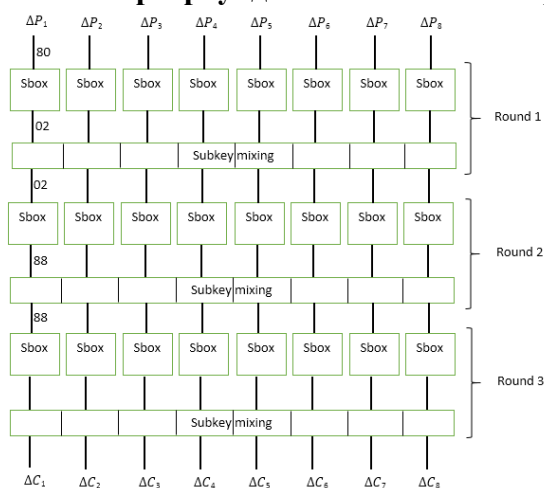
Проанализируем данный криптоанализ на примере трехраундового блочного шифра, представленного на рис. 1. Для двух заранее подобранных шифротекстов P_1 и P_2 злоумышленником вычисляется «дифференциал» $\Delta P = P_1 \oplus P_2$. И с помощью ΔP пытаются определить каким должен быть «дифференциал» шифротекстов $\Delta C = C_1 \oplus C_2$.

Данный шифр имеет 64 битный размер блока и 128 битный ключ. На каждом раунде входной блок делится на 8 байт, каждый из которых проходит через функцию подстановки Sbox. После этого данные перемешиваются с 64 битным подключом Subkey. Функция перемешивания представляет собой XOR операцию.

Предположим, что злоумышленник решил проверить дифференциал $0x80$. Для этого он генерирует произвольный байт X_1 , и вычисляет $X_2 = X_1 \oplus 80$. Далее атакующий прогоняет X_1 и X_2 через функцию Sbox и получает значения Y_1 и Y_2 . Для каждой такой пары X_1 и X_2 , дифференциал которых равен 80, атакующий в состоянии получить дифференциал ΔY . Анализируя полученные значения, атакующий выбирает такое значение ΔY , которое имеет большую вероятность возникновения.

Возвращаясь к нашему примеру, предположим, что из всех 256 пар X_1 и X_2 , в 192 случаях $Y_1 \oplus Y_2 = 02$. Таким образом, вероятность того, что при заданном $\Delta X = 80$, значение $\Delta Y = 02$, составляет $192/256 = 3/4$. Это в свою очередь означает, что при заданном $\Delta X = 80$, с вероятностью $P_1 = 3/4$ на вход второго раунда попадут два значения U_1 и U_2 , такие, что $\Delta U = 02$.

Рисунок 1 – Трехраундовый блочный шифр [4]



Ключ не влияет на значение дифференциалов. Так как при шифровании разных текстов ключ не изменяется и перемешивание с ключевой последовательностью осуществляется с помощью XOR, то при вычислении ΔU байты ключа взаимно исключаются. Для раскрытия свойств второго раунда, злоумышленник генерирует новые 256 пар

входных байт X_1 и X_2 , таких, что $X_1 \oplus X_2 = 02$. Произведя вычисление функции S_{box} для каждой пары X_1 и X_2 , атакующий замечает, что в 64 случаях из 256 $\Delta Y = 88$. Т.е. вероятность того, что $\Delta Y = 88$, для заданного $\Delta X = 02$, составляет $P_2 = 64/256 = 1/4$. Таким образом, произведя нехитрый подсчет вероятностей, атакующий понимает, что для указанного шифра для каждой пары байт X_1 и X_2 , таких, что $\Delta X = 80$, с вероятностью $P = P_1 P_2 = 3/4 * 1/4 = 3/16$, дифференциал внутреннего состояния шифра перед последним раундом составляет $\Delta Y = 88$. Обладая этим знанием, атакующий генерирует несколько пар текстов таких, что $\Delta P = 808080808080$ и приступает к побайтовому подбору подключа третьего раунда.

Покажем, каким образом осуществляется вскрытие первого байта подключа. Для каждого из 256 возможных вариантов первого байта $\text{Subkey}[0]$ и для каждой пары шифротекстов $\{C_1, C_2\}$, злоумышленник вычисляет $U_1 = S_{\text{box}}(C_1 \oplus \text{Subkey}[0])$ и $U_2 = S_{\text{box}}(C_2 \oplus \text{Subkey}[0])$.

Если $\text{Subkey}[0]$ угадан правильно, то приблизительно 3 из каждых 16 пар U_1 и U_2 при вычислении ΔU будут равны 88. Подобрав таким образом наиболее вероятный первый байт подключа Subkey , атакующий может перейти ко второму байту и действуя аналогичным образом вскрыть весь ключ третьего раунда.

После того, как ключ последнего раунда будет раскрыт, злоумышленник может приступить к атаке на предпоследний раунд и действуя подобным образом в конечном итоге получит информацию о всех раундовых ключах шифра.

Недостатки метода:

- высокие требования к времени и объему данных (на практике),
- chosen plaintext attack,
- необходим большой объем памяти для хранения возможных ключей,
- может быть использован только для вскрытия с известным открытым текстом,
- для 16-этапного DES этот метод менее эффективен, чем вскрытие грубой силой.

Линейный криптоанализ разработан М. Мацуи (M. Matsui) в 1993 г. [4]. Является вариантом атаки, направленным на выявление наиболее вероятного значения ключа, использованного в процессе шифрования блочным алгоритмом. Атакующий выполняет атаку «известный открытый текст» на несколько различных сообщений, зашифрованных на одном и том же ключе. Атакующий анализирует входящие и исходящие значения для каждого S-блока. Он анализирует вероятность того, что определенные входящие значения дают в ре-

зультате определенную комбинацию. Выявление таких результирующих комбинаций позволяет ему оценивать вероятность для различных значений ключа, пока он не найдет повторяющийся шаблон, имеющий высокую вероятность.

Алгоритм заключается в следующем [4]:

1) пусть T — количество текстов, для которых левая часть уравнения (ф.1) равняется 0, тогда *если* $T > N/2$, где N — число известных открытых текстов,

2) предположить, что $K_{I1} \oplus K_{I2} \oplus \dots \oplus K_{Ic} = 0$ (когда $P > 1/2$) или 1 (когда $P < 1/2$), *иначе*,

3) предположить, что $K_{I1} \oplus K_{I2} \oplus \dots \oplus K_{Ic} = 1$ (когда $P > 1/2$) или 0 (когда $P < 1/2$).

Из анализа алгоритма можно сделать вывод, что успех алгоритма напрямую зависит от значения $|P-1/2|$ и от количества доступных пар открытый/закрытый текст N .

Защита от линейного криптоанализа:

1) минимизирование вероятностных смещений,

2) выбор высоко нелинейных S -блоков и усиление диффузии,

3) учитывать эффект линейных оболочек,

4) применение бент-функций в построении стойких S -блоков.

Проанализировав два примера, можно сделать вывод, что дифференциальный криптоанализ не подходит для применения на практике из-за того, что для него существуют высокие требования ко времени и объему данных, и подходит только для атаки типа «выбранный открытый текст». Линейный криптоанализ подходит только для атаки типа «известный открытый текст», из чего можно вывести зависимость чем больше вероятность P равенства (ф.1) отличается от $1/2$, тем меньшее количество открытых текстов N необходимо для атаки.

ЛИТЕРАТУРА

1. Урбанович, П.П. Защита информации методами криптографии, стеганографии и обфускации / П.П. Урбанович. – Минск: БГТУ, 2016.

2. Heys, H. M. A Tutorial on Linear and Differential Cryptanalysis/ H.M. Heys. [Электронный ресурс]. – Режим доступа: http://www.engr.mun.ca/~howard/PAPERS/ldc_tutorial.pdf. – Дата доступа: 15.04.2019.

3. Biham, E. Differential cryptanalysis of the Data Encryption Standard/E. Biham, A. Shamir. [Электронный ресурс]. – Режим доступа: <http://bookre.org/reader?file=1242706>. – Дата доступа: 15.04.2019.

4. Matsui, M. Linear Cryptanalysis method of DES cipher/ M. Matsui. [Электронный ресурс]. – Режим доступа: <https://www.cs.bgu.ac.il/~beimel/Courses/crypto2001/Matsui.pdf>. – Дата доступа: 15.04.2019.

УДК 004.4

Студ. А.В. Глушко

Науч. рук. доц. Н.И. Гурин

(кафедра информационных систем и технологий, БГТУ)

СИСТЕМА УЧЕТА РАБОЧЕГО ВРЕМЕНИ СОТРУДНИКОВ ИТ-КОМПАНИИ

ИТ-технологии занимают большую часть рынка в современном мире. ИТ-технологии охватывают области создания, передачи, хранения и восприятия информации, не ограничиваясь только компьютерными технологиями. Для производства программных продуктов организуется и разрабатывается множество проектов.

В крупном бизнесе лидирующее место занимает компания SAP[1]. Множество программных продуктов разрабатывается данной компанией, и для ведения учета проекта, и потраченного времени работников существует не так много программного обеспечения. Сейчас основным программным обеспечением для учета рабочего времени сотрудников компании и управлением проектом являются программные продукты Jira [2] и VersionOne [3].

Основными минусами Jira и VersionOne является перегруженность и непонятность интерфейса пользователя, а также избыточность функционала. В результате было принято решение разработать систему с понятным интерфейсом пользователя и необходимым функционалом, который будет решать поставленные задачи, и не будет запутывать пользователя.

В ходе изучения документации по патентам сделан вывод о том, что многие из них не специализируются в области учета рабочего времени сотрудников компании. В связи с этим является логичным вывод о том, что в настоящее время актуальность и необходимость в таком программном продукте высока.

Областью использования разрабатываемого программного обеспечения являются ИТ-компании, ориентированные на автоматизацию крупного бизнеса. В качестве базы программной системы использованы SAP-системы. Основной причиной использования данных систем, является ориентирование на удобство программиста, и учета его рабочего времени в рамках единой системы, в которой он будет работать в данный момент.

Пользовательский интерфейс построен с использованием сервиса SAPWebIDEFull-Stack облачной платформы SAPCloudPlatform, и далее все приложения опубликованы в общий доступ с использованием облачного сервиса Portal.

Основной список используемых средств для разработки системы учета рабочего времени сотрудников ИТ-компаний представлен в таблице 1.

Таблица 1 – Средства разработки

Инструмент разработки	Описание
S4H System (Linux x86_64), release 1.00.122.06.1485334242	Система, для реализации внутренней логики приложения и базы данных, базирующаяся на архитектуре S/4 HANA (SAP HANA, облачные инфраструктуры, пользовательский интерфейс на основе Fiori)
S4G System (Linux x86_64), release 16.0.02.04	Система для реализации моста между системой S4H и приложением, посредством OData сервиса
CDS (Core Data Services)	Это инфраструктура, которая может использоваться разработчиками баз данных для создания базовой (постоянной) модели данных, которую службы приложений предоставляют клиентам пользовательского интерфейса
BOPF (Business Object Processing Framework)	Предоставляет набор общих сервисов и функциональных возможностей для ускорения, стандартизации и модулизации разработки
OData services	Используется для определения лучших практик, необходимых для создания и использования RESTful API
SAP Cloud Platform	Открытая платформа как услуга (PaaS), которая предоставляет возможности в памяти, услуги базовой платформы и уникальные микросервисы для создания и расширения интеллектуальных облачных приложений с поддержкой мобильных устройств.

Данные средства разработки являются зарекомендованным стеком программного обеспечения необходимого для разработки полноценного Web-приложения, использующего базу данных и инфраструктуру, расположенную в SAP-системе.

После анализа необходимых ресурсов для управления проектом и временем работника были выделены следующие роли пользователя, которые представлены в таблице 2.

Таблица 2 – Роли пользователя

Роль	Описание
Employee	Сотрудник компании, работающий на проекте и выполняющий разработку программного обеспечения для заказчика
PM(Project Manager)	Руководитель проекта – сотрудник компании, главной задачей которого является создание, получение проекта и предоставление кадров для данного проекта. Также может участвовать в непосредственной разработке программного обеспечения для заказчика
HR	Сотрудник компании, выполняющий функции управления кадрами предприятия.
Office Manager	Сотрудник компании, выполняющий функции управления офисами и офисными помещениями компании

Каждая роль пользователя имеет свой уникальный набор функций. Роль пользователя, имеющая свой набор функций, необходима для запрета, либо разрешения доступа конкретному пользователю.

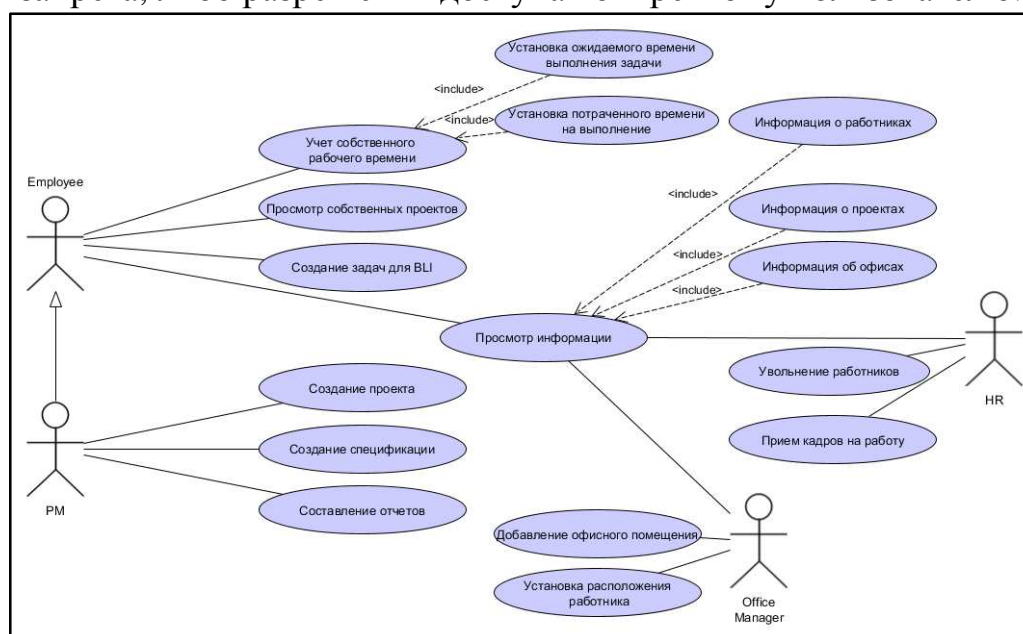


Рисунок 1 – UML Use Case диаграмма

Роль пользователя дает право на выполнение одной либо нескольких операций. То есть роль может давать право на редактирование, просмотр, либо создание какого-либо объекта, и уже при наличии роли, пользователь сможет выполнить разрешенную операцию.

Роль пользователя, в свою очередь, содержит перечень авторизационных объектов, каждый из которых отвечает за тот, или иной объект данных. Список спроектированных авторизационных объектов представлен в таблице 3.

Таблица 3 – Авторизационные объекты

Авторизационный объект	Описание
ZANL_BLI	Регулирует доступ на изменение, чтение, удаление бэклогов
ZANL_OFC	Регулирует доступ на изменение, чтение, удаление офисов
ZANL_PRJ	Регулирует доступ на изменение, чтение, удаление проектов
ZANL_TLG	Регулирует доступ на изменение, чтение, удаление логов времени
ZANL_WRK	Регулирует доступ на изменение, чтение, удаление работников проекта
ZANL_EMPL	Регулирует доступ на изменение, чтение, удаление работников

Таким образом, используя SAP-системы и предложенный инструментарий, разработана система учета рабочего времени сотрудников, которая предоставляет не только необходимый функционал пользователю, а также разделяет функции, исполняемые работниками.

ЛИТЕРАТУРА

1. SAPDefinition [Электронный ресурс] / TechTarget – Режим доступа: <https://searchsap.techtarget.com/definition/SAP> – Дата доступа: 14.04.2019.
2. JiraSoftware [Электронный ресурс] / Atlassian – Режим доступа: <https://www.atlassian.com/software/jira> – Дата доступа: 14.04.2019.
3. VersionOne [Электронный ресурс] / VersionOne – Режим доступа: <https://community.versionone.com/VersionOne> – Дата доступа: 14.04.2019.

УДК 004.4

Студ. В.В. Дмитрачёнок
Науч. рук. ст. преп. Ю.О. Герман
(кафедра информационных систем и технологий, БГТУ)

ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ ОЗВУЧИВАНИЯ ЭЛЕКТРОННЫХ ПОЧТОВЫХ СООБЩЕНИЙ

Сегодня информационные технологии занимают важную часть нашей жизни. Они ежедневно сопровождают нас в выполнении большинства задач: поиск маршрута на картах города, оплата услуг через мобильный банкинг, заказ такси и продуктов на дом. И одной из задач для современных приложений является доступность для людей в любой момент времени. Сегодня человек получает в день в 5 раз больше информации, чем несколько десятилетий назад.

Одним из главных способов восприятия электронной информации у человека является зрение. На данное средство восприятия приходится 90% полученной информации. На втором месте находится слух [1].

Огромное количество времени уходит на чтение электронной информации ежедневно, из-за этого могут возникнуть проблемы со зрением. Однако современные технологии совсем не делают упор на использование второго по популярности способа восприятия информации, ведь это может сэкономить время на чтение, так и облегчить глаза от постоянного напряжения. Для реализации целей, описанных выше, можно использовать голосовые модули для автоматического чтения текстовой информации. Из современного, довольно популярного, способа использования слуха, как основным средством восприятия какой-либо информации, являются аудиокниги. С помощью данного средства пользователи могут погружаться в мир художественной литературы без использования зрения. Если речь идет о способах получения информации, то не стоит забывать о людях с нарушениями

зрения, для которых задачи восприятия визуальной информации становятся достаточно затруднительными.

Так как важным атрибутом современного человека является электронная почта, то почему бы не сделать процесс чтения электронных писем более простым и удобным как обычному человеку, так и пользователю с некоторыми проблемами со зрением.

Цель работы: разработка модуля чтения электронных писем, который позволит упростить ежедневное чтение почты, а также поможет сделать пользователям с ограниченными зрительными способностями более комфортную работу с почтой.

Для начала был проведен патентный поиск по данной теме, в результате которого не было найдено ни одного патента.

В ходе поиска аналогов было обнаружено несколько продуктов, среди которых два самых популярных будут рассмотрены ниже.

Почтовый клиент Microsoft Outlook является одним из самых популярных средств для просмотра почты. Имеются как версии для настольных и мобильных операционных систем, так и веб версия, доступная в браузере. Однако данное приложение может не только читать и отправлять письма, но и составлять расписания по полученным сведениям о событиях из писем. Таким образом, Outlook является универсальным органайзером с огромным количеством функций.

Что касается функции озвучивания писем, то данное средство имеет функции по озвучиванию выделенного текста, которая по умолчанию отключена.

Особенности:

- большой спектр возможностей при работе с почтой;
- календарь событий.

Недостатки:

- сложный в освоении интерфейс.

Веб-приложение Checker Plus является расширением для браузеров от Google и Mozilla, которое позволяет управлять несколькими учетными записями электронной почты, настроить уведомления о входящих письмах на почту, а также включить функцию голосового чтения письма. Всплывающее окно предварительного просмотра почты позволит читать, архивировать, пометить прочитанные или удалять электронные письма, не покидая текущей вкладки.

Это расширение помогает облегчить работу с почтой, без постоянной загрузки страницы почты в браузере.

Особенности:

- удобное управление и быстрый доступ при использовании браузера.

Недостатки:

постоянная нагрузка в виде фонового процесса браузера.

Проанализировав аналоги, было решено реализовать в данном веб-ресурсе следующее:

реализовать максимально удобное использование почтового клиента;

выполнение задач чтения и написания писем;

реализация озвучивания писем;

возможность поиска писем с помощью строки поиска;

возможность прослушивания писем с интересующей пользователю темой.

Для доступа к почтовому ящику необходимо знать адреса IMAP и SMTP серверов, используемого сервиса для получения и отправления писем соответственно.

Самым простым в использовании почтовым сервисом является mail.ru. К нему можно подключиться с любого пользовательского приложения, не меняя настройки аккаунта. Для использования более распространённого и надежного сервиса Gmail.com, необходимо внести некоторые изменения в параметры безопасности учетной записи.

В качестве инструмента для работы с почтой используется удобная библиотека mail.dll. С помощью данной библиотеки можно получать и отправлять письма с довольно подробной настройкой.

Для озвучивания содержимого электронного письма необходимо для начала знать язык, на котором оно написано, после чего идет настройка синтезатора речи и далее озвучка текста.

Используемые в разработке ресурсы помогли достичь цели в создании простого в использовании почтового клиента с модулем озвучивания электронных писем.

ЛИТЕРАТУРА

1. Электронный журнал “Livejournal” [Электронный ресурс] / Режим доступа: <https://arsenische.livejournal.com/51864.html> - Дата доступа: 20.10.2018.

УДК 004.4

Студ. А.А. Костюк
Науч. рук. ст. преп. Ю.О. Герман
(кафедра информационных систем и технологий, БГТУ)

ВЕБ-РЕСУРС ДЛЯ РАСПРОСТРАНЕНИЯ АУДИОКНИГ

Из всех проявлений человеческого творчества самое удивительное и достойное внимания — это книги. В книгах живут думы прошедших времен; внятно и отчетливо раздаются голоса людей, прах которых давно разлетелся, как сон. Все, что человечество совершило, передумало, все, чего оно достигло, — все это сохранилось на страницах книг [1].

Термин “аудиокнига” вошёл в употребление в 1970-х, когда записи начали производиться на аудиокассетах. Как отраслевой стандарт этот термин утвердила Audio Publishers Association в 1994 году.

Сегодня большинство бестселлеров сразу издаются как в бумажном, так и в аудиоварианте.

Некоторые издательства аудиокниг предоставляют выбор между женским или мужским голосом, языками, изменением тона, тембром и интонацией дикторов [2].

Цель данного проекта: реализация веб-ресурса, предоставляющего возможность ознакомления как с самими аудиокнигами, так и с их авторами, а также выбор наиболее подходящих книг с учетом предпочтений пользователя.

На текущий момент реализовано множество различных Интернет-ресурсов, предоставляющих так или иначе доступ к аудиокнигам. Например, онлайн прослушивание, покупка и бесплатное скачивание. В качестве платного аналога было решено выбрать самый популярный сайт Litres.ru, а в качестве бесплатного – Rutracker.org.

Компания ЛитРес, основанная в 2005 году, сегодня является лидером на рынке лицензионных электронных книг в России и странах СНГ [3].

Достоинства ЛитРес:

- возможность прослушивания фрагментов книг;
- удобный поиск по жанрам;
- присутствует собственная аннотация;
- реализован рейтинг книг.

Недостатки ЛитРес:

- плохое описание книги;
- преобладают платные товары.

Rutracker.org — крупнейший русскоязычный BitTorrent-трекер, насчитывающий почти 15,8 миллиона зарегистрированных учётных

записей. На трекаре зарегистрировано почти 1,88 миллиона раздач (из которых более 1,52 миллиона — «живых»), суммарный размер которых составляет 3,632 петабайта [4].

Достоинства:

- максимально подробное описание товара;
- заранее известен размер файла;
- быстрое и удобное скачивание книг.

Недостатки:

- нет возможности прослушивания фрагмента книги для определения качества записи;
- отсутствует рейтинг книг.

Платные варианты нецелесообразно использовать по причине того, что существует возможность бесплатного получения того же товара.

Проанализировав аналоги, было решено реализовать в данном веб-ресурсе следующее:

- подробное описание книги;
- рейтинг книг;
- возможность подбора книг по нескольким критериям, таким как, автор, исполнитель, жанр и название;
- возможность прослушивания фрагментов книги в целях ознакомления с содержимым, оценивания качества записи и озвучивания;
- разделение пользователей на администраторов с возможностью редактирования и добавления книг и простых пользователей;
- вывод ближайших книг серии и подбор аналогичных книг методом анализа иерархий;
- добавление страницы “Хочу”, на которой пользователи могут оставлять заявки на интересующие их книги, которые затем администраторы смогут добавить.

Для скачивания аудиокниг необходимо иметь либо архив с книгой на сервере, либо ссылку на сторонний ресурс, с которого существует возможность скачивания файлов.

Самым лучшим и наиболее эффективным способом скачивания аудиокниг является использование торрент-файлов, которые позволят установить связь между программой на машине клиента и физическим расположением файлов для осуществления процесса скачивания необходимых файлов.

Главная страница веб ресурса представлена на рисунке 1.



Рисунок 1 – Главная страница веб-ресурса

Как видно на рисунке 1, книги подбираются по 4 критериям и каждую из книг возможно изучить подробнее, добавить в свою библиотеку или удалить, если у данного пользователя есть права администратора.

Пример страницы с описанием книги приведен на рисунке 2.



Рисунок 2 – Страница описания книг

Для удобства использования данного веб-ресурса был реализован вывод 5 ближайших книг серии. Кроме того, в данном веб-ресурсе

реализован метод анализа иерархий, который позволяет подбирать наиболее подходящие по теме книги.

ЛИТЕРАТУРА

1. Творческая группа “СамИздат” [Электронный ресурс] / Режим доступа: <https://www.livelib.ru/publisher/37837/books-tvorcheskaya-gruppa-samizdat/> - Дата доступа: 07.04.2019.

2. Аудиокнига [Электронный ресурс] / Режим доступа: http://allaudiobook.ru/load/fakty/audiokniga_ehto/18-1-0-625/ – Дата доступа: 10.04.2019.

3. ЛитРес [Электронный ресурс] / Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D1%82%D0%A0%D0%B5%D1%81/> – Дата доступа: 11.04.2019.

4. RuTracker.org [Электронный ресурс] / Режим доступа: <https://ru.wikipedia.org/wiki/RuTracker.org/> – Дата доступа: 11.04.2019.

УДК 004.4

Студ. А.А. Рыжковский

Науч. рук. ст. преп. Ю.О. Герман

(кафедра информационных систем и технологий, БГТУ)

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЯ ПРИ ВЫБОРЕ ЗЕМЕЛЬНОГО УЧАСТКА

С каждым днем применение информационных технологий становится все шире и шире.

Можно привести множество примеров использования информационных технологий, но в большинстве случаев цель одна – стремление облегчить труд. Широко распространённое применение информационных технологий осуществляется при анализе данных.

Анализ данных необходим для того, чтобы в большом наборе информации обнаружить ранее неизвестные, нетривиальные, а главное практически применимые знания, необходимые для принятия решений в различных сферах человеческой деятельности.

В основе анализа данных лежит подход принятия решений, который является наиболее массовой операцией практически в любой сфере целенаправленной деятельности человека. Потенциально возможные варианты действий называются альтернативами, а характеристики качества альтернатив – критериями. При решении сложных систем задач рассматривают множество альтернатив (минимум две) и множество критериев, причем учет многих критериев приближает постановку задачи к реальной действительности.

Таким образом, процесс принятия решений не может быть полностью формализован и состоит в использовании процедур многокритериального анализа альтернатив в условиях неполной информации о внешних факторах, влияющих на эффективность решений с учетом субъективных предпочтений лиц, принимающих решение [1].

Анализ данных можно применить к такой сфере, как выбор земельного участка. Обычно, перед тем, как человек решает купить себе земельный участок, он анализирует по определенным критериям оптимальное и качественное решение. Среди этих критериев человек, в первую очередь, обращает внимание на личные возможности и требования, такие как бюджет на покупку земли, параметры, которые он бы хотел иметь у своего земельного участка и т.д. Затем уже среди списка предложений он пытается выявить наиболее лучший для него результат.

Вопрос выбора земельного участка сегодня становится все актуальнее. На территории Республики Беларусь появляется все больше доступных земельных участков и все больше людей хочет или планирует застроить земли.

Кто-то хочет построить свой дом, дачу или даже коттедж. Но как уже говорилось ранее, каждый участок характеризуется своими параметрами, наиболее важными из которых является наличие электричества, газа, хорошо-развитой инфраструктуры и других удобств для комфортной жизни. Причем эти параметры не равнозначны, какой-то из параметров может быть более предпочтительный, чем другой.

Целью данного проекта является создание приложения, которое позволяет автоматизировать выбор земельного участка на основе предпочтений, заданных пользователем.

Были проанализированы сайты, с помощью которых возможно искать объявления о продаже земельных участков. Существует несколько Интернет-ресурсов по поиску земельных участков и все они похожи. Среди наиболее популярных выделяется Kufar.by и Realt.by.

Kufar – крупнейшая площадка объявлений Республики Беларусь.

Особенности Kufar:

- объявления различных категорий;
- широкий набор параметров для поиска;
- простой и понятный интерфейс.

Недостатки Kufar:

- отсутствие учета предпочтений пользователя.

Проект Realt.by создан в 2005 году на основе общегородской базы агентств недвижимости «Недвижимость 3.0». Ресурс предоставляет посетителям актуальную, ежедневно обновляемую информацию от агентств.

Особенности:

- различные категории недвижимости;
- большое количество критериев поиска.

Недостатки:

- сложный интерфейс;
- отсутствие важности критериев.

Проанализировав аналоги, было решено реализовать в данном веб-приложении следующее:

- учет важности критериев;
- подбор наиболее оптимального варианта земельного участка;
- фильтры земельных участков по критериям;
- возможность пользователю самому определять важность каждого критерия.

При разработке веб-приложения надо определиться с тем, какой метод принятия решения используется для поиска лучшей альтернативы.

В проекте задача представлена в неструктурированном виде, т.е. описываются только на содержательном уровне (в словесной форме).

Методы экспертного анализа (экспертных оценок) предназначены в основном для решения неструктурированных задач.

Методы экспертного анализа представляют собой совокупность процедур, направленных на получение от специалиста (эксперта) информации, необходимой для принятия решения. Эти методы основаны на опыте, знаниях и интуиции специалистов-экспертов. Процесс принятия решения на основе этих методов не является строго формализованным и упорядоченным. Сам эксперт не всегда может четко объяснить процесс решения проблемы. Информация, полученная от экспертов, подвергается обработке на основе математических методов.

Существует несколько экспертных методов принятия решений и все они классифицируются по разным признакам. Сравнительная характеристика приведена в таблице [2].

Для проекта будет лучше всего использовать модифицированный алгоритм Кемени-Снелла. Это связано с тем, что в нашем приложении предпочтения будет задавать только один эксперт, поэтому все методы с классификацией «коллективный» не подойдут. Алгоритм Кемени-Снелла является методом на основе попарных сравнений аль-

тернатив, суть которого заключается в том, что он позволяет анализировать и выбирать альтернативы, оцениваемые по критериям различных видов: числовым, качественным, «да-нет» и т.д. Алгоритм также позволяет учитывать важность критериев. Алгоритм Саати тоже является индивидуальным методом принятия решений на основе попарных сравнений альтернатив. Но был выбран алгоритм Кемени-Снелла из-за того, что на начальном этапе алгоритма вычисляются веса критериев с помощью алгоритма Саати, т.е. алгоритм является более полным и точным.

Таблица – Классификация методов принятия решений

Признак для классификации	Виды методов экспертного анализа	Описание	Примеры
Количество участвующих экспертов	Индивидуальные	Решение принимается на основе суждений одного эксперта	Алгоритм Саати, Модифицированный алгоритм Кемени-Снелла
	Коллективные	Решение принимается на основе суждений группы экспертов	Метод ранга, метод предпочтений, метод Дельфи, метод мозгового штурма

Таким образом, алгоритм Кемени-Снелла является основой веб-приложения, с помощью которого пользователь на основе своих предпочтений получит наиболее подходящий для него земельный участок среди предложенных.

ЛИТЕРАТУРА

1. Смородинский, С. С. Методы и системы принятия решений / С. С. Смородинский; Н. В. Батин – Ч. 2 – Минск: БГУИР, 2001. – 3 с.
2. Смородинский, С. С. Методы анализа и принятия решений в слабоструктурированных задачах / С. С. Смородинский; Н. В. Батин – Минск: БГУИР, 2002. – 13 с.

ЧЕЛОВЕЧЕСКОЕ МЫШЛЕНИЕ ИЛИ МАШИННЫЙ АЛГОРИТМ

Современные технологии ежедневно проникают в нашу жизнь и становятся её неотъемлемой частью. Роботизация, 3D-печать, блокчейн, искусственный интеллект – и это далеко не весь перечень инноваций, которые не только помогают людям решать различные задачи, но и заставляют их менять род деятельности, осваивать новые специальности. Предсказывают, что к окончанию века 70% современных профессий не будут требовать участия человека. Говоря иначе, проникновение искусственного интеллекта во многие сферы жизни человека неизбежно и замена им специалистов – только вопрос времени.

Цель данной работы: проанализировать, насколько тяжело создать алгоритм, имитирующий человеческое мышление в конкретной области. Может ли человеческий мозг со всей его сложностью быть смоделирован, может ли его суть быть схвачена и реализована в машине [1]?

Основой нашего исследования стала работа индийского математика Inder J. Taneja, в которой он представляет числа, используя лишь одну цифру и простейшие алгебраические операции: сумму, разность, произведение, деление, и логическую операцию – конъюнкцию. Но главным является то, что он делает это, применяя навык устного счета и не имея четкого алгоритма.

Пример таких вычислений:

$$\begin{aligned} 29 &= (1+1+1) \times (11-1) - 1 = 22 + 2 + 2 + 2 + 2 / 2 = 3 + 33 - 3 / 3 = \\ 44 - 4 \times 4 + 4 / 4 &= 5 \times 5 + 5 - 5 / 5 = 6 \times 6 - 6 - 6 / 6 = 77 - 7 \times 7 + 7 / 7 = \\ &= 8 + (88 + 88 - 8) / 8 = 9 + 9 + 99 / 9. \end{aligned}$$

Стоит обратить внимание на то, что он записывает нужное число с помощью минимального количества знаков.

В ходе исследования нами была изучена литература по созданию алгоритмов искусственного интеллекта, а также классификация методов машинного обучения, базовые виды нейросетей, их разграничение и создание. Кроме того, нами был написан код, который реализует операцию представления числа с помощью заданной цифры.

Алгоритмом называется точная и понятная инструкция исполнителю совершить последовательность действий, направленных на решение поставленной задачи [2]. В качестве исполнителя алгоритма может выступать как человек, так и техническое устройство. Различные машины являются формальными исполнителями алгоритмов. От

них не требуется понимание сущности решаемой задачи, но требуется безошибочное выполнение последовательности команд.

Машинное обучение — класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач.

Классификация алгоритмов машинного обучения:

- линейная регрессия;
- логистическая регрессия;
- линейный дискриминантный анализ (LDA);
- деревья принятия решений;
- наивный Байесовский классификатор;
- к-ближайших соседей (KNN);
- сети векторного квантования (LVQ);
- метод опорных векторов (SVM);
- бэггинг и случайный лес;
- бустинг и AdaBoost.

Нейросеть – это обучаемая система. Она действует не только в соответствии с заданным алгоритмом и формулами, но и на основании прошлого опыта. Этаким ребенком, который с каждым разом складывает пазл, делая все меньше ошибок.

Искусственный интеллект (ИИ) — способность интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека.

Компьютерная теория разума, один из сильнейших аргументов в пользу ИИ, утверждает, что мозг можно декодировать и что его работа сводится к коммуникации между нейронами и образованию операционных кластеров. То есть не существует никакой великой загадки сознания, есть лишь непонимание принципов его работы.

В ходе проведенного исследования мы пришли к выводу, что тех знаний, которые имеются у нас на данный момент недостаточно для того, чтобы в полной мере раскрыть данный вопрос. С нашей точки зрения создание алгоритма, копирующего человеческое мышление в данной области весьма трудоемкая задача, которая под силу только специалистам высокого уровня.

ЛИТЕРАТУРА

1. Kurzweil R. The Singularity Is Near: When Humans Transcend Biology/ R. Kurzweil. New York, Viking, 2005. 434p.
2. Chomsky N. Language and Problems of Knowledge / N. Chomsky Cambridge, MA: MIT Press, 1988. 29p.

УДК 004.031

Студ. В.А. Колодко
Науч. рук. ст. преп. Н.А. Жиляк
(кафедра информационных систем и технологий, БГТУ)

БЛОКНОТ 2.0

Вопрос об информационной безопасности на сегодняшний день стоит очень остро. Сегодня у каждого человека есть множество личных данных (такие как: пароли от банковских карт, пароли от аккаунтов, личные счета) и их конфиденциальность интересует каждого. Передавать данные по сети в «голом» виде не безопасно, так как существует множество программ для перехвата их. В этом и стоит вопрос о защите и способе шифровании данных.

Криптография зародилось еще до нашей эры. Уже в то время при передаче информации, пытались максимально защитить ее, чтобы недруги не смогли завладеть этой информацией. Одни из первых техник защиты текста: скитала, квадрат Полибия, шифр Цезаря, диск Энея.

Цель работы: разработать приложение по защите личной информации пользователя.

Реализованы следующие задачи и этапы для достижения поставленной цели:

1. **Архитектура сетевого приложения.** Приложение должно состоять из клиентской части и серверной. Для удобства пользователя клиентская часть будет состоять из приложения для ПК и мобильного устройства. Сервер будет выполнять задачи хранилища, а клиент – отображать информацию и взаимодействовать с пользователем.

При разработке были использованы такие технологии как: язык программирования C#, WPF для написания клиентской части на ПК, Xamarin для написания клиентской части на мобильное устройство, EntityFramework для взаимодействия сервера и базы данных.

2. **Передача зашифрованных данных.** Отправка осуществляется по принципу HTTPS. У клиента и сервера есть пара ключей (публичный и приватный) благодаря которым можно зашифровать и расшифровать данные. Данные передаются по протоколу TCP, после подключения сервер и клиент обмениваются публичными ключами, затем происходит общение между ними в зашифрованном виде.

```

byte[] bytes = new byte[1024];
int bytesRec = handler.Receive(bytes);

publicClientK = Encoding.GetEncoding(1251).GetString(bytes, 0, bytesRec);
handler.Send(Encoding.GetEncoding(1251).GetBytes(publicK));

bytesRec = handler.Receive(bytes);

```

Рисунок 1 – Обмен публичными ключами

3. **Шифрование и дешифрование.** Когда клиент подключился к серверу, идет выборка из имеющихся шифровок. С помощью функции RandomEncryption, случайным образом выбирается способ шифрования на один сеанс. Так же, каждое сообщение перед отправкой подвергается транспозиции функцией TransPosition. То есть каждое сообщение, уходит в сеть задом наперед.

```

public String RandomEncryption()
{
    Random rand = new Random();

    switch (rand.Next(0, 3))
    {
        case 0: return "RSA";
        case 1: return "DES";
        case 2: return "AES";
        case 3: return "RC2";
    }

    return "";
}

public byte[] TransPosition(byte[] buff)
{
    buff.Reverse().ToArray();
    return buff;
}

```

Рисунок 2 – Функции RandomEncryption и TransPosition

4. **Дополнительная защита.** Сервер, получивший сообщение, сохраняет его в файле, название которого совпадает с логином пользователя, файл имеет формат “.nb2”.

```

byte[] bytes = new byte[1024];
int bytesRec = con.Receive(bytes);
String mess = Encoding.GetEncoding(1251).GetString(rsa.RSADecrypt(Encoding.GetEncoding(1251).GetBytes(Encoding.GetEncoding(1251)
con.Send(rsa.RSAEncrypt(Encoding.GetEncoding(1251).GetBytes("FileSave"), Encoding.GetEncoding(1251).GetBytes(publicClientK)));
con.Shutdown(SocketShutdown.Both);
con.Close();

StreamWriter savefile = new StreamWriter(@"Client\" + Login + ".nb2", false, System.Text.Encoding.Default);
savefile.WriteLine(mess);
savefile.Close();

```

Рисунок 3 – Сохранение информации в файл

Перед сохранением в файл, сообщение проходит шифрование, метод которого выбирается с помощью функции RandomEncryption. Вся информация о пользователе хранится в таблице БД, там же есть информация об времени последнего шифрования. Каждый час, сервер перезаписывает файл пользователя, без потери данных. Проверка времени производится с помощью функции TimeCheck. Вызов этой функции происходит каждые 5 минут. Благодаря этой системе, производится дополнительная защита информации пользователя.

```
public void ReEncrypt(String Login)
{
    byte[] buff = Decrypt(Login);
    buff = Encrypt(buff, Login, RandomEncryption());
    TransPosition(buff);
    ReSave(buff);
}

public void TimeCheck()
{
    var clients = se.Client.ToList();
    DateTime date = DateTime.Now;
    foreach(var client in clients)
    {
        if(date.Subtract(client.DataLastSave.Value).Minutes == 60)
        {
            ReEncrypt(client.ClientLogin);
        }
    }
}
```

Рисунок 4 – Функции ReEncrypt и TimeCheck

Таким образом, программа будет полезна для людей всех возрастов, которые хотят защитить личные данные и всегда иметь их под рукой, так как записи на бумаге не надежны.

ЛИТЕРАТУРА

1. WindowsPresentationFoundation [Электронный ресурс] / Режим доступа: <https://docs.microsoft.com/en-us/dotnet/framework/wpf/> - Дата доступа: 15.11.2018.
2. XamarinDocumentation [Электронный ресурс] / Режим доступа: <https://docs.microsoft.com/en-us/xamarin/> - Дата доступа: 10.04.2019.

УДК 557.114:616-006

Студ. Е.А. Бурая
Науч. рук. ассист. К.С. Патапенко
(кафедра информационных систем и технологий, БГТУ)

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ ПЕРСОНАЛЬНЫМИ ДАНЫМИ КАНДИДАТОВ И СОТРУДНИКОВ КОМПАНИИ

В каждой компании есть целые отделы кадров. Здесь постоянный большой поток информации и данных. Каждый день добавляется новая информация, удаляется или меняется старая. Сотрудникам в данном отделе необходимо получить быстрый, простой способ для манипулирования данными сотрудников.

ИТ-компания не является исключением. Здесь ещё больший поток данных, который постоянно обновляется. Трудности вызывает еще и тот факт, что при подборе новых сотрудников проводится большая работа, которая делится на несколько этапов.

При подборе кандидатов принимается участие не только менеджер по персоналу, но и технический специалист. Техническим специалистом является сотрудник компании, который принимает участие в разработке программных продуктов. Менеджер по персоналу ставит целью подбор новых сотрудников, которые легко войдут в уже сложившийся коллектив и будет поддерживать ценности, которые сложились в компании.

Технический специалист очень важен, т.к. он оценивает уровень навыков, умений и знаний кандидатов в той области, в которую необходим сотрудник. Важно, чтобы будущий сотрудник понимал процесс разработки, имел определённый опыт работы. Это необходимо для снижения затрат на введение нового специалиста в процесс разработки.

Деятельность каждого из участников процесса мало связана, но полученные данные о кандидате должны быть собраны в одном месте для удобного получения необходимой информации. Это позволит получить целостное представление о будущем сотруднике.

Веб-приложение позволит решить данную проблему. Также оно поможет осуществить поиск необходимой информации удобно и быстро, позволит отслеживать процесс рассмотрения кандидатов. Всё это упростит и повысит работу по подбору новых сотрудников.

Проанализировав особенности работы в данной области, я выделила 4 участника процесса: кандидат, технический специалист, менеджер по персоналу и остальные сотрудники. Каждая группа имеет свои особенности, которые учтены при разработке. Так, например, кандидаты не имеют доступа к приложению, более того, они и не по-

дозревают о его существовании. Пользоваться им могут только сотрудники.

Для каждого вида пользователей свой функционал. Минимальный функционал у разработчиков. Пользователи данного вида не имеют возможности получить данные о кандидатах и других пользователях, а также зарплаты других сотрудников. Такие пользователи не могут менять информацию, кроме своего пароля.

Большими правами обладают HR-менеджеры. Они могут добавлять людей, кандидатов и сотрудников, изменять данные о кандидате, удалять все объекты, кроме пользователей. У них есть возможность просмотреть данные пользователей.

Менеджеры имеют право добавлять людей, кандидатов и сотрудников, просматривать и изменять данные о каждом типе объектов. Изменять запрещено только данные пользователей. Также менеджеры могут удалять существующих пользователей.

Также должна присутствовать возможность вносить изменения в данных кандидатов не только HR-менеджерами, но и другими сотрудниками, проверяющие практические и теоретические знания. Также будет реализована возможность оставлять этими сотрудниками комментарии по кандидату, с которым ведётся работа.

В данный момент база данных имеет модель, представленную на рисунке 1.

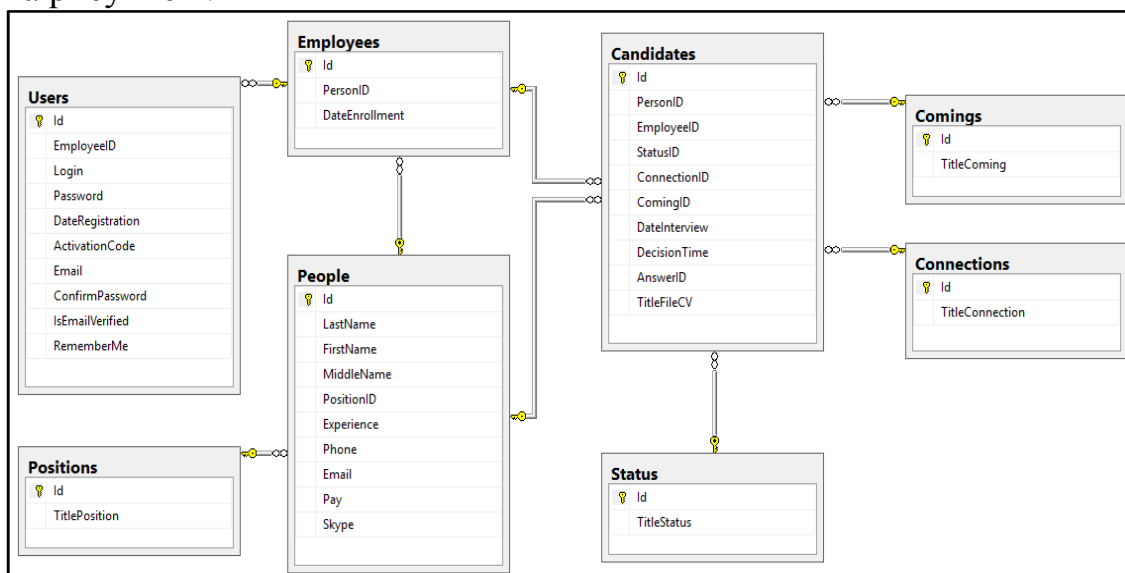


Рисунок 1 – Модель данных веб-приложения

В ней отображены объекты, которые являются их участниками. Таблица Candidates соответствует участнику кандидат, хранит всю информацию о нём. Аналогично проведена работа с пользователями приложения. Таблица Employees соответствует остальным участникам.

Таблица People является основой для всех Candidates и Employees. В ней содержатся поля, которые являются общими для кандидатов и сотрудников. Для разделения возможностей сотрудников используется поле в таблице People, которое обозначает их роль в компании.

ЛИТЕРАТУРА

1. Базаров Т.Ю. Управление персоналом М.: Юнити, 2010 – 560 с.
2. Иванова С.В. Искусство подбора персонала. Как оценить человека за час М.: Альпина Паблишер, 2012 – 112 с.
3. <https://worldsellers.ru/kto-takoj-hr-menedzher/>

УДК 004.03

Студ. Е.И. Дуденкова
Науч. рук. зав. кафедрой В.В. Смелов
(кафедра информационных систем и технологий, БГТУ)

ЭЛЕКТРОННАЯ ОЧЕРЕДЬ НА КОНСУЛЬТАЦИЮ К ПРЕПОДАВАТЕЛЮ ВУЗА

Информационные технологии плотно вошли в нашу жизнь, все больше образовательный процесс обретает электронный вид, решение большого количества обыденных задач становится невозможным без наличия интернета. Неотъемлемым условием повышения эффективности управленческого труда является оптимальная информационная технология, обладающая гибкостью, мобильностью и адаптивностью к внешним воздействиям. Образовательная сфера не является исключением, и использование информационных технологий в развитии и функционировании этой сферы играет большую роль. Введение современных информационных технологий в область образования дает возможность качественно поменять методы и организационные формы обучения, сделав его более комфортным и доступным. Информационные технологии – главная часть процесса модернизации образования.

У современного преподавателя время расписано по минутам, поэтому ему необходима помощь в структурировании расписания. Удаленное взаимодействие между преподавателем и студентом эффективнее и быстрее, чем стандартные методы. Электронная очередь на консультацию не только экономит время преподавателя и студента, но и помогает развивать организованность. Автоматизация этого процесса поможет лучше организовать очередь, а также исключить вероятные ошибки.

Цели проекта:

- облегчить процесс обучения для преподавателей и студентов;
- организовать расписание консультаций у преподавателя.

В результате работы был создан программный продукт, компонентами которого являются: сервер, веб-сайт и мобильное приложение под платформой Android. Веб-сайт предоставляет функционал для администратора, преподавателя, и студента. Мобильное приложение предоставляет функционал для преподавателя и студента.

Администратор получает доступ к списку преподавателей, студентов и предметов. В его обязанности входит создание преподавателей, редактирование и удаление их, удаление просроченных консультаций и студентов, окончивших вуз. Чтобы избавить администратора от монотонной работы, список преподавателей загружается из файла. Администратор вправе вносить изменения в консультации, при этом преподавателю и студенту приходит оповещение о внесенных изменениях на электронную почту и телефон.

Преподаватель имеет возможность редактировать свою личную информацию, в том числе изменить пароль, создавать консультацию и определять ее статус.

Студенту предоставляется доступ к личному кабинету, с возможностью его изменения, регистрации и авторизации, список доступных и личных консультаций.

Для разработки серверной части был выбран широко используемый язык сценариев PHP[1]. Мобильное приложение разработано на основе языка программирования Java[2].

Для хранения данных на удаленном сервере, была выбрана база данных MySQL[3]. Для хранения данных на мобильном устройстве использовалась локальная база данных – SQLite[4].

ЛИТЕРАТУРА

1. PHP [Электронный ресурс]. – 2019. – Режим доступа: <https://www.w3schools.com/php/default.asp>. – Дата доступа: 15.01.2019.
2. Java[Электронный ресурс]. – 2019. – Режим доступа: <https://metanit.com/java/android>. – Дата доступа: 20.01.2019.
3. MySQL [Электронный ресурс]. – 2019. – Режим доступа: <https://metanit.com/sql/mysql/1.1.php>. – Дата доступа: 20.01.2019.
4. SQLite [Электронный ресурс]. – 2019. – Режим доступа: <https://metanit.com/java/android/14.5.php>. – Дата доступа: 20.01.2019.

ОБУЧЕНИЕ ПРОГРАММИРОВАНИЮ ПОСРЕДСТВОМ АВТОРСКОГО ПРОГРАММНОГО ПРОДУКТА

По данным сайта dev.by в Беларуси 60 тысяч разработчиков, из которых только 20% составляют девушки. IT-специальности не пользуются популярностью у абитуриенток, так как существует ряд стереотипов насчет девушек в IT:

- IT-сфера, традиционно считается мужской;
- типичный программист – это парень в очках и свитере с оленями, который 24 часа в сутки проводит за компьютерным столом;
- девушки не настолько умные;
- женской логики не существует;
- девушка не может стать разработчиком;
- девушки недавно начали осваивать IT;
- back-end – это мужское.

Однако большинство стереотипов не имеет рациональных оснований. В IT-сфере девушек меньше, но это не значит, что сфера полностью закрыта. Компаниям нужны специалисты, и не важно, парни или девушки.

Цель проекта: разработать приложение для обучения программированию для девушек.

В ходе работы над проектом были решены следующие задачи:

- разработка мобильного приложения для пользователей;
- разработка веб-приложения для администраторов;
- разработка серверной части приложения;
- проектирование базы данных;
- разработка дизайн-макета приложения;
- реализация удобного, интуитивно-понятного интерфейса.

Разработанное программное средство ProGirl позволяет девушкам в любое время осваивать новые технологии. Приложение кросс-платформенное, доступно для самых популярных мобильных платформ, а именно: iOS и Android, разработано на основе технологий React Native [1], Redux [2], JavaScript [3], Realm [4]. Мобильное приложение предназначено для совершения операций с правами пользователя. Приложение локально хранит загружаемые данные для более быстрого доступа к ним. Данные, хранящиеся в локальной базе данных, будут доступны при отсутствии подключения к сети Интернет.

Для входа в приложение необходимо зарегистрироваться и ввести свои данные (рис. 1).

The image shows two side-by-side mobile app screens. The right screen is titled 'Registration' and has a blue '< Back' button. It features two large pink circular logos with the text 'PROGIRL'. Below the logos are three input fields: 'Name' (with a person icon), 'Email' (with an envelope icon), and 'Password' (with a lock icon). A pink rounded button labeled 'Sign up' is at the bottom. The left screen is for login, featuring the same two 'PROGIRL' logos. It has two input fields: 'Name' (with a person icon) and 'Password' (with a lock icon). A pink rounded button labeled 'Sign in' is at the bottom. Below the 'Sign in' button is a link: 'Don't have account? Sign up here!'.

Рисунок 1 – Авторизация и регистрация

Пользовательский интерфейс был выполнен в оттенках розового, серого и белого цветов для того, чтобы девушкам было приятно использовать приложение.

После успешной авторизации пользователь попадает на главный экран приложения. На главном экране доступны все функции приложения, а именно: раздел с учебными материалами, раздел с советами, раздел с вопросами и ответами, личные заметки и личный кабинет (рис.2). В меню приложения доступны такие функции, как редактирование профиля, различные настройки, включая смену языка (приложение поддерживает русский и английский), а также выход из приложения. В личном кабинете отображены фотография, имя пользователя, дата рождения, email, мобильный телефон, skype, текущее местоположение.

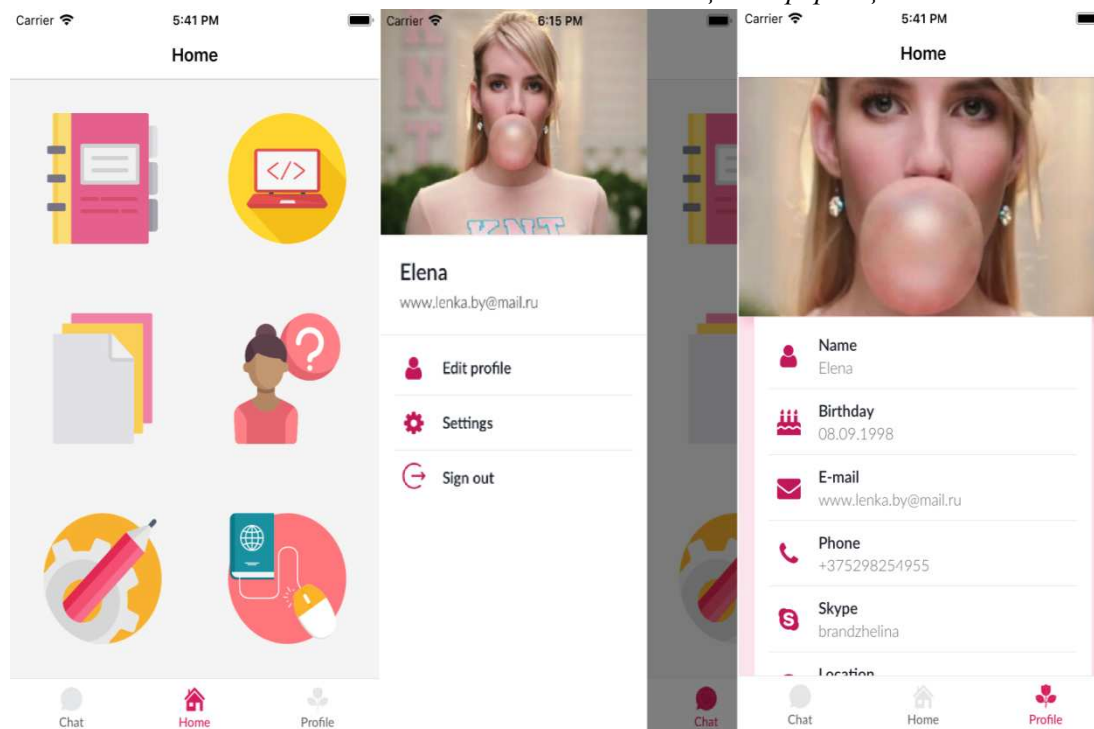


Рисунок 2 – Главный экран, меню, личный кабинет пользователя

Чтобы начать изучение новой технологии, необходимо выбрать один из предлагаемых курсов (рис.3). Каждый курс содержит в себе темы для изучения, которые расположены в порядке возрастания сложности. В процессе обучения предусмотрено тестирование по каждому ключевому разделу выбранного курса.

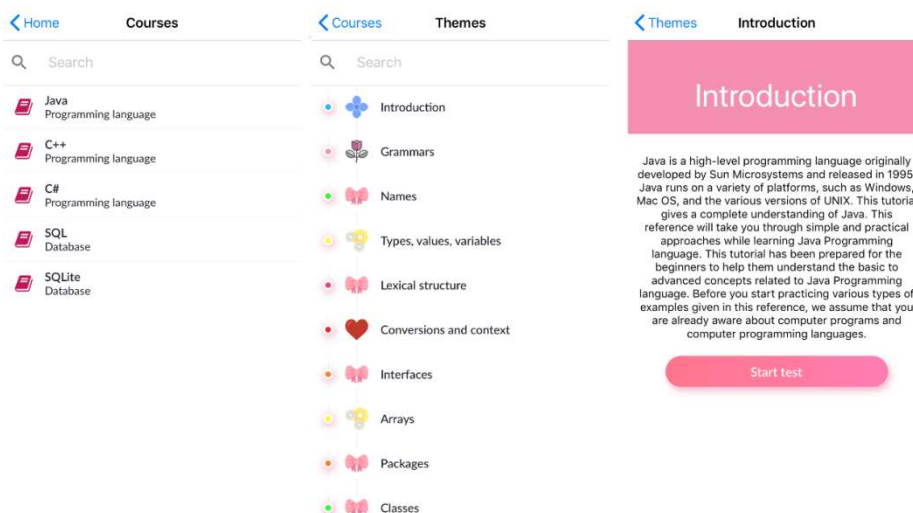


Рисунок 3 – Прохождение курса

В приложении реализован функционал для управления заметками (рис. 4). Управление заметками осуществляется напрямую в локальной базе данных Realm. Доступные операции: создание, редактирование и удаление.

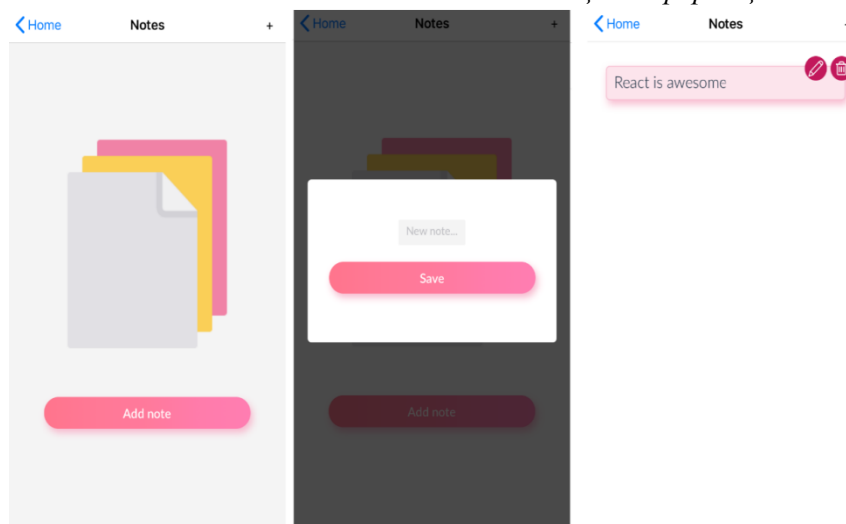


Рисунок 4 – Заметки

Веб-клиент предназначен для совершения операций с правами администратора, т. е. для управления данными. В веб-приложении можно добавлять, удалять и редактировать данные хранящиеся в базе данных сервера. Для разработки веб-сайта были использованы следующие технологии: React, HTML5, CSS, Bootstrap framework [5], Redux, Javascript.

Приложение ProGirl привлекает женскую аудиторию красивым дизайном, удобным интерфейсом, понятным донесением информации. Таким образом, была реализована главная цель проекта, а именно: разработано программное средство для обучения программированию, которое вдохновляет девушек на изучение новых технологий.

ЛИТЕРАТУРА

1. React Native - A framework for building native apps using React [Электронный ресурс]. – Режим доступа: <https://facebook.github.io/react-native/>. – Дата доступа: 13.04.2019.
2. Redux - A Predictable State Container for JS Apps [Электронный ресурс]. – Режим доступа: <https://redux.js.org/>. – Дата доступа: 13.04.2019.
3. Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/>. – Дата доступа: 13.04.2019.
4. Realm: Create reactive mobile apps in a fraction of the time [Электронный ресурс]. – Режим доступа: <https://realm.io/>. – Дата доступа: 13.04.2019.
5. Bootstrap - The most popular HTML, CSS, and JS library in the world. [Электронный ресурс]. – Режим доступа: <https://getbootstrap.com/>. – Дата доступа: 13.04.2019.

Студ. Е.Н. Бондарчик
Науч. рук. ст. преп. Е.А. Блинова
(кафедра информационных систем и технологий, БГТУ)

ХРАНИЛИЩЕ ДАННЫХ КАК ИНСТРУМЕНТ АВТОМАТИЗАЦИИ РАСПРЕДЕЛЕНИЯ НАГРУЗКИ ПРЕПОДАВАТЕЛЕЙ

Хранилища данных спроектированы для принятия бизнес-решений путем сбора, консолидации и организации данных для анализа и отчетности при помощи систем делового анализа (OLAP-систем) и интеллектуального анализа данных (Data mining). Хотя хранилища данных построены на технологии реляционных баз данных, структура базы данных хранилища данных существенно отличается от структуры базы данных системы оперативной обработки транзакций (OLTP-системы). Реляционные базы данных проектируются под определенную цель. Но поскольку назначение хранилища данных отличается от назначения OLTP, архитектурные аспекты реляционной базы данных, что поддерживают OLAP-решения, отличаются от архитектурных характеристик базы данных OLTP [1].

Хранилище данных используется, в первую очередь, аналитиками и менеджерами для принятия решений ведения бизнеса, и должно удовлетворять следующим условиям:

- Обеспечение хранилища согласованных данных;
- Предоставление различных мощных аналитических инструментов, таких как OLAP и интеллектуальный анализ данных;
- Давать быстрый результат выполнения комплексных запросов;
- Работать без вмешательства в OLTP-системы.

Множество хранилищ данных, удовлетворяющих, данным условиям, имеют общие характеристики:

- Хранение истории изменений, т.е. вариантность по времени;
- Включают в себя как разряженные, так и сводные данные;
- Консолидация разрозненных данных из нескольких источников при сохранении непротиворечивости, т.е. интегрированность;
- Фокусировка на основном процессе системы;
- Многомерная модель данных.

Данный подход к проектированию системы автоматизации распределения нагрузки преподавателей был выбран с учетом требований и нужд со стороны данного процесса. Программные средства позволяют оптимизировать работу инженеров кафедры, путем сни-

жения затрат ресурсов на распределение нагрузки преподавателей. В результате исследования рынка был сделан вывод об отсутствии мобильных версий приложений для решения данной задачи.

Разработанное мобильное приложение позволяет решать возникающие вопросы, без привязки к персональному компьютеру и рабочему месту.

В ходе проектирования системы автоматизации распределения нагрузки преподавателей были выбраны следующие компоненты и технологии их реализации:

- Веб-сервер, разработанный на облачной платформе «Microsoft Azure» технологией ASP.NET, позволяющий принимать запросы клиентских приложений, обрабатывать их и отправлять ответы;
- Веб-клиент, разработанный также с помощью технологии ASP.NET, позволяет инженерам управлять базой данных, вносить изменения, управлять процессом распределения нагрузки с помощью задания параметров и фильтрации отдельных ресурсов;
- Мобильное приложение, разработанное на языке Java для платформы Android 4.4 и выше, позволит получать актуальную информацию преподавателям об их текущей нагрузке;
- Сервер базы данных – Microsoft Azure SQL Databases – это облачный сервис от корпорации Microsoft, предоставляющий возможность хранения и обработки реляционных баз данных, а также генерации отчетности, что является конечным результатом работы системы.

В ходе анализа требований были поставлены и выполнены следующие задачи:

- Формирование нагрузки для каждого преподавателя;
- Учет планового числа часов учебной нагрузки на ставку;
- Формирование планового распределения нагрузки по кафедре.

ЛИТЕРАТУРА

1. Кимбэлл, Р., Роз, М. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling / Кимбэлл, Р., Росс, М. – Индианаполис.: Wiley, 2013. – 600 с.

УДК 004.915

Инж.-програм. лаборатории ИТОЭУИ и ДО БГТУ Е. С. Сахонь
Науч. рук. доц. кафедры Н. Н. Гурин
(кафедра информационных систем и технологий, БГТУ)

ИНТЕРФЕЙС ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ ДИСТАНЦИОННОГО ВЫПОЛНЕНИЯ ВИРТУАЛЬНЫХ ЛАБОРАТОРНЫХ РАБОТ

Программный комплекс виртуальных лабораторных работ для дистанционного обучения по дисциплине «Физика» выполнен с помощью межплатформенной среды разработки Unity и языка программирования C#. Каждая виртуальная лабораторная работа комплекса основана на материалах «Лабораторного практикума», изданного в учреждении образования «Белорусский Государственный Технологический Университет».

Экспорт приложения осуществлен под веб-ресурсы, поэтому были поставлены следующие задачи при создании интерфейса для выполнения лабораторной работы:

- проектирование UX интерфейса, то есть пользовательского процесса поведения в ходе выполнения заданий, его последовательность действий;
- разработка UI, то есть дизайна удобного и простого пользовательского интерфейса;
- разработка макета сайта для дальнейшего размещения установки;
- верстка сайта для дистанционного выполнения лабораторных работ.

Выполнение виртуальной лабораторной работы происходит аналогично действиям студента, осуществляемым при выполнении реальной лабораторной работы на занятии с преподавателем.

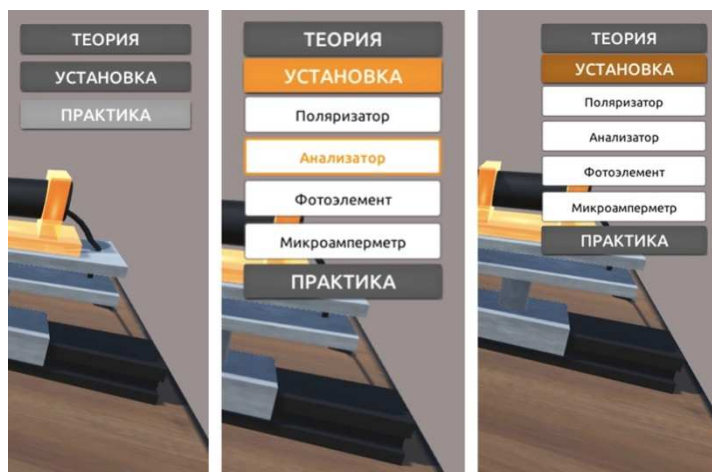


Рисунок 1 – Отображение состояний кнопок

При разработке главного меню выдели три главных раздела: теория, установка и практика, в соответствии с которыми были сделаны необходимые кнопки (рисунок 1). Для того, чтобы кнопки всегда находились в правом верхнем углу экрана вне зависимости от его разрешения, главному контейнеру, содержащему меню, была добавлена соответствующая точка привязки. Данные кнопки имеют четыре разных состояния (покоя, наведения, нажатия и недоступности), в соответствии с которыми и был разработан их дизайн.

При выполнении лабораторной работы первую очередь пользователю предлагается ознакомиться с теорией, для этой цели служит интуитивно понятный интерфейс, на котором пользователь выбирает необходимый ему вопрос и читает краткую теорию к нему. Просмотр полной версии теоретического материала осуществляется на веб-ресурсе.

Вторым пунктом для выполнения лабораторной работы является ознакомление с установкой. Для того, чтобы узнать, как называется какая-либо часть установки, необходимо навести курсор на ее название в описании и тогда название этого элемента подсветится в описании, а на самой установке подсветится соответствующий элемент (рисунок 2).



Рисунок 2 – Ознакомление с установкой

Следующим и основным элементом для выполнения лабораторной работы является прохождение раздела «Практика». В данном разделе в нижней части экрана последовательно выводятся сообщения, которые отображают текущее задание для пользователя, соответствующее этапам хода работы лабораторной работы.

В ходе выполнения снятые измерения с установки записываются в нужные столбцы таблицы автоматически, по нажатию кнопки «Записать значение». После осуществления всех необходимых заданий из раздела «Практика» выводится соответствующее сообщение о завершении лабораторной работы.

Третий блок содержит в себе симулятор, в котором можно выполнить лабораторную работу дистанционно (рисунок 3). В нём полностью аналогичный порядок выполнения работы и обработка результатов измерений книжному варианту.

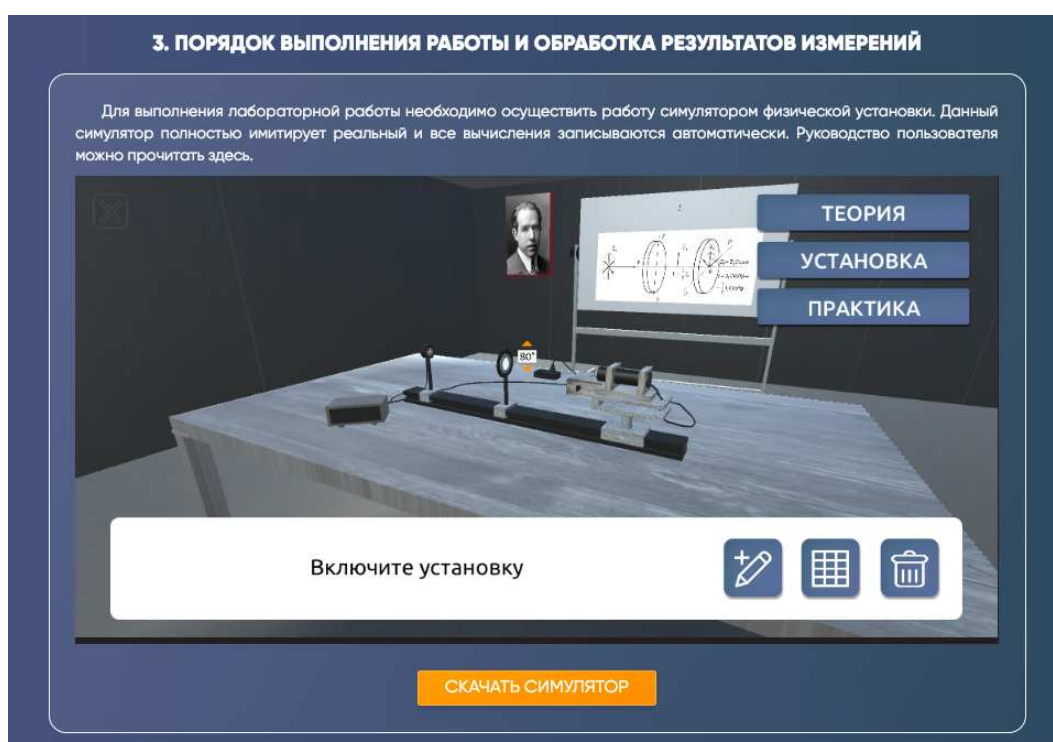


Рисунок 3 – Блок с симулятором на сайте

Для создания макета сайта и обработки изображений, которые находятся на сайте, использовался многофункциональный графический редактор Adobe Photoshop. Сайт написан благодаря технологиям HTML, CSS, Bootstrap и JS.HTML. Для публикации 3D-моделей использовалась программная библиотека WebGL.

Веб-сайт разработан для получения информации о созданном приложении, он функционирует как полноценный веб-ресурс.

Таким образом, разработанный сайт дает возможность быстро и удобно получить всю необходимую информацию о лабораторной работе, провести измерения и закрепить полученные навыки.

Магистр. Д.С. Лосик
Науч. рук. доц. Н.Н. Буснюк
(кафедра информационных систем и технологий, БГТУ)

МОДЕЛЬ ЗАДАЧИ ОПТИМИЗАЦИИ МАРШРУТОВ ПЕРЕВОЗКИ НЕФТЕТОПЛИВА

Одним из способов экономии ресурсов при транспортировке грузов является применение систем поддержки принятия решений в области транспортной логистики. Актуальность подобных задач вызвана тем, что до 50% всех затрат на логистику связано с транспортными издержками. Особенно актуально решение задач маршрутизации в условиях крупных городов и мегаполисов, в виду разветвленной транспортной сети и большого количества точек доставки. Основными задачами планирования доставки являются: задача нахождения оптимальных маршрутов и задача оптимальной загрузки транспортных средств.

Цель исследования: обоснование выбора математической модели по формированию перевозочных схем и критериев для оптимизации, формулировка оптимизационной задачи целочисленного линейного программирования.

Общая постановка рассматриваемой задачи оптимизации выглядит следующим образом:

- Транспортная система компании по доставке нефтепродуктов содержит следующие компоненты:

- 1) автозаправочные станции (АЗС, автоматические АЗС (ААЗС));
- 2) парк бензовозов (свои и наёмные бензовозы);
- 3) нефтебазы (свои и/или чужие, с которыми заключены договора по обеспечению);
- 4) список маршрутов перемещения бензовозов от нефтебаз к АЗС;
- 5) список реализуемых компанией видов топлива.

- Компания по обеспечению нефтепродуктами может иметь в распоряжении свои нефтебазы или пользоваться услугами других, поскольку некоторые виды топлива, реализуемые на АЗС, отсутствуют на своих нефтебазах. В связи с этим компания нуждается в дополнительных источниках топлива для обеспечения поставок.

- Парк бензовозов компании может содержать собственные транспортные средства (ТС) или привлекать наёмные бензовозы. Бензовозы имеют различные емкости автоцистерн и нормы расхода топлива.

- Известно количество ТС каждого типа, все ТС одного типа имеют одинаковые характеристики (интервал работы водителя, количество секций в ТС и их емкости, время обслуживания на нефтебазе, время слива на АЗС для каждой секции, стоимость перемещения 1 т. топлива на 1 км, скорость движения).

В качестве примера, на рисунке 1 представлена схема движения бензовозов и расположение АЗС относительно стоянки, нефтебазы и друг друга. Каждой стрелке присвоен приоритет расстояния перевозки. Чем меньше приоритет, тем меньше расстояние АЗС [1].

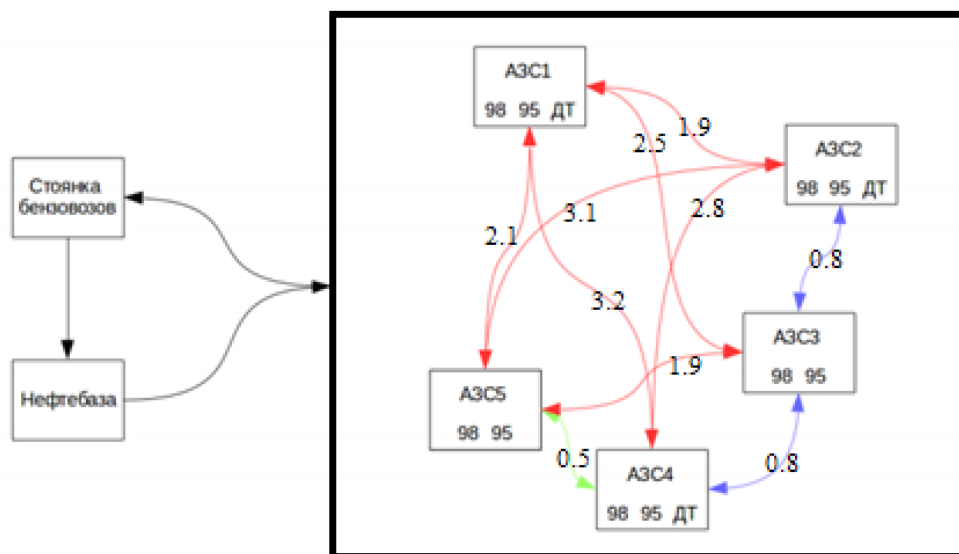


Рисунок 1 - Схема движения бензовозов и расположение АЗС относительно стоянки, нефтебазы и друг друга

Проведем обоснование выбора математической модели по формированию перевозочных схем и критериев по оптимизации.

Под **моделью** понимается любое описание логистической системы, которое может быть использовано вместо нее для исследования ее свойств и прогнозирования возможных вариантов поведения.

Моделирование логистических систем можно проводить различным образом и приходиться в итоге к разным моделям. Для того чтобы построить модель, необходимо следовать следующим принципам [2]:

- модель должна иметь поведение, структуру и функции, такую как у моделируемой логистической системы или ее компонента;
- отклонения параметров модели в процессе ее функционирования от соответствующих параметров моделируемой логистической системы не должны выходить за рамки допустимой точности моделирования;
- на основании исследования модели и ее поведения должно быть возможным обнаружить новые свойства моделируемой логисти-

ческой системы, не содержащиеся в исходном материале, использованном для составления данной модели;

- проводить исследования и эксперименты на модели должно быть более удобно, чем на реальной логистической системе.

Метод линейного программирования наиболее приемлем для оптимизации транспортного потока, поскольку достаточно прост, удобен в применении и удовлетворяет условиям по оптимизации транспортного потока. Данный метод представляет собой направление математики, которое изучает методы решения задач, характеризующиеся линейной зависимостью между переменными и линейными критериями оптимальности.

Математическая модель для решения задачи с помощью линейного программирования включает в себя целевую функцию, где необходимо определить оптимальное значение (максимум или минимум), систему ограничений, а также требование, чтобы переменные были неотрицательными [3].

В общем виде модель записывается следующим образом:

- целевая функция:

$$f(\bar{x}) = c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \max(\min); \quad (1)$$

- ограничения:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \{ \leq = \geq \} b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \{ \leq = \geq \} b_2, \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \{ \leq = \geq \} b_m; \end{cases} \quad (2)$$

- требование неотрицательности переменных:

$$x_j \geq 0, j = \overline{1, n}, \quad (3)$$

где a_{ij}, b_i, c_j ($i = \overline{1, m}, j = \overline{1, n}$) – заданные постоянные величины.

Для решения задачи необходимо правильно найти оптимальное значение функции при соблюдении ограничений.

Используем метод линейного программирования для постановки задачи оптимизации маршрутов грузоперевозки топлива от нефтебаз к АЗС, находящимся в мегаполисе.

Примем множество P маршрутов p , для которых существует ТС типа k . Введем неотрицательные целочисленные переменные $x_{kp} = h$, если h штук ТС типа k назначены на маршрут p .

Сформулируем задачу линейного программирования:

Критерий оптимальности:

$$\min F(x) = \sum_{(k,p) \in P} c_{kp}x_{kp}, \quad (4)$$

где c_{kp} – стоимость маршрута p транспортного средства типа k .

При ограничениях на:

- запасы топлива на нефтебазах:

$$\sum_{(k,p) \in P} a_{kprj} x_{kp} \leq D_{rj}, \forall r, j, \quad (5)$$

где a_{kprj} – объем вывезенного топлива вида r с нефтебазы j ;

- удовлетворение спроса АЗС:

$$d_{ri} \leq \sum_{(k,p) \in P} b_{kpri} x_{kp} \leq d_{ri} + \delta_{ri}, \forall r, i, \quad (6)$$

где b_{kpri} – объем привезенного топлива вида r на АЗС i , δ_{ri} – заданные максимальные величины превышения спроса;

- назначение не более h_k ТС типа k на все маршруты:

$$\sum_{p \in M(P)} x_{kp} \leq h_k, \forall k \in K(P), x_{kp} \in Z, \forall (k, p) \in P, \quad (7)$$

где h_k – количество (идентичных) ТС типа k .

Таким образом, сформулирована задача целочисленного линейного программирования оптимизации маршрутов перевозки нефтетоплива, которая будет решаться в магистерской диссертации на тему «Метод поиска оптимальных маршрутов перевозки нефтетоплива».

ЛИТЕРАТУРА

1. А. Л. Неволлина. Разработка метода планирования для нефтепродуктообеспечения сети автозаправочных станций. / А. Л. Неволлина; ФГАОУ ВО «УФУ» Новосибирск 2017. – 34–35, 60 с.
2. В. В. Михайлович Алгоритмическая реализация численных методов / В. В. Михайлович; БНТУ – Минск, 2018: 32-38, 42-53 с.
3. О. В. Мартинсон. Моделирование автоматизации проектирования логических систем и потоковых процессов в строительстве / О. В. Мартинсон; ФГБОУ ВПО «МГСУ» – Москва, 2011: 1-2 с.

УДК 004.4

Студ. В.А. Баранов

Науч. рук. доц. Н.Н. Пустовалова

(кафедра информационных систем и технологий)

ШИФРОВАНИЕ ИНФОРМАЦИИ МЕТОДОМ СОЗДАНИЯ ДИНАМИЧЕСКОГО КЛЮЧА

Проблема защиты информации от несанкционированного доступа заметно обострилась в настоящее время в связи с широким распространением компьютерных сетей, особенно глобальных. Защита информации необходима для уменьшения вероятности разглашения,

утечки, умышленного искажения, утраты или уничтожения информации, представляющей определенную ценность для ее владельца.

Шифрование – это такое преобразование информации, которое делает исходные данные нечитаемыми и трудно раскрываемыми без знания ключа. Под ключом понимается секретная информация, определяющая, какое преобразование из множества возможных, выполняется в каждом конкретном случае над открытым текстом.

Основной принцип шифрования, предлагаемого в данной статье, заключается в том, что каждый символ получает свой уникальный код, который генерируется динамически по ходу процесса шифрования, т. е. обрабатывается отдельно, независимо от остальных.

Программа, реализующая шифрование информации методом создания динамического ключа, написана на языке программирования C++. Символ строки передается в функцию пользователя вместе с кодом шифрования, который получен с помощью встроенной функции `random()` и функции `clock()` стандартной библиотеки `ctime` [1]. В зависимости от алфавита (кириллица или латиница) и от регистра (верхнего и нижнего) символ будет шифроваться по-своему.

Рассмотрим известный шифр с названием ROT1. Этот шифр основан на принципе смещения символов (англ. rotate). Цифра после записи значит, насколько будет смещен символ по алфавиту вправо. Существуют и другие версии этого шифра (ROT2, ROT13 и т. д.).

В разработанной программе шифрование происходит по принципу шифра Цезаря, который является разновидностью шифра ROTn: исходный символ заменяется символом, стоящим в алфавитном порядке справа от исходного на количество, указанное заранее [2]. Поскольку символов в тексте может быть больше, чем в алфавите, то данная процедура замены продолжается нужное количество раз, используя символы алфавита заново.

Полученный результат записывается в динамический массив, который имеет размер такой же, как и исходная строка (на каждый символ по шифру). Ключ шифрования получают все символы в строке, однако шифруются только алфавитные символы. Формула шифрования имеет вид: $(\text{clock}() + \text{rand}() \% 10000) \% (\text{rand}() \% 1000)$.

Дешифрование – обратный шифрованию процесс. На основе ключа зашифрованный текст преобразуется в исходный открытый.

В разработанной программе может осуществляться и дешифровка с использованием сгенерированного шифра, однако этот шифр пропадает при закрытии программы, или при шифровании другого сообщения.

Использованный метод дешифровки делает невозможным чтение сообщения на другом девайсе. Чтобы дешифровать сообщение, следует ввести зашифрованную строку и ее шифр.

Таким образом, разработанный алгоритм, реализованный в программе, обеспечивает высокий уровень защиты данных против дешифрования и возможной модификации, защищенность информации основывается только на знании ключа, малое изменение исходного текста или ключа приводит к значительному изменению зашифрованного текста, экономичность реализации алгоритма при достаточном быстройдействии.

ЛИТЕРАТУРА

1. Н. В. Пацей. Основы алгоритмизации и программирования: учеб.-метод. пособие для студентов специальности «Информационные системы и технологии» / Н. В. Пацей. – Минск: БГТУ, 2010. – 289 с.
2. Ф. Циммерман. Введение в криптографию / Ф. Циммерман. – PGP Corporation, 2004. – 18 с.

УДК 004.4

Магистрант А.Г. Оганесян
Науч. рук. доц. Н.Н. Пустовалова
(кафедра информационных систем и технологий, БГТУ)

АНАЛИЗ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ PANDAS НА ЯЗЫКЕ PYTHON

Учет личных расходов является одним из инструментов оптимизации личных финансов. Система, которая смогла бы вести учет личных расходов и на основании собранных данных выдавать пользователю информацию о его тратах является актуальной и будет всегда востребована. Для обработки и анализа данных при решении данной задачи можно воспользоваться библиотекой pandas.

Pandas – это высокоуровневая Python библиотека для анализа данных. Она построена поверх низкоуровневой библиотеки NumPy, которая написана на языке Си, что является большим плюсом для производительности. С помощью pandas можно как анализировать данные, так и в связке с matplotlib визуализировать их.

Чтобы эффективно работать с pandas, необходимо освоить самые главные структуры данных библиотеки: DataFrame и Series[1].

Объект Series представляет собой объект, похожий на одномерный массив, но отличительной его чертой является наличие ассоциированных меток (индексов), вдоль каждого элемента из списка. Если индекс явно не задан, то pandas автоматически создаёт RangeIndex от

0 до N-1, где N – общее количество элементов. Доступ к элементам объекта Series возможен по их индексу.

Объект DataFrame лучше всего представить в виде обычной таблицы, поскольку он является табличной структурой данных. В любой таблице всегда присутствуют строки и столбцы. Столбцами в объекте DataFrame выступают объекты Series, строки которых являются их непосредственными элементами.

В задаче анализа расходов с помощью pandas можно получить такие востребованные данные, как: расходы пользователя за некоторое время, предварительно сгруппированные по категориям; расходы пользователя по определенной категории за каждый месяц; все расходы пользователя ежемесячно, без разделения на категории.

Для выполнения первой задачи (отображения сгруппированных по категориям расходов) необходимо считать данные из файла с расходами в DataFrame с помощью функции read_csv(), далее с помощью функции groupby() необходимо сгруппировать расходы по атрибуту «категория». Теперь все что осталось, это получить Series для колонки «amount», вызвать функцию sum(), после чего в нашем распоряжении будет новый объект типа DataFrame (рис. 1). Останется только визуализировать его.

```
import pandas

df = pandas.read_csv('expenses.csv', parse_dates=True, index_col='date')

new_df = df.groupby('category')['amount'].sum()
plot = new_df.plot(kind='barh', grid=True, legend=True)
plot.get_figure().savefig("all_by_cat.png")
```

Рисунок 1– Исходный код визуализации расходов по категориям

На рис. 2 представлена визуализация расходов по категориям.

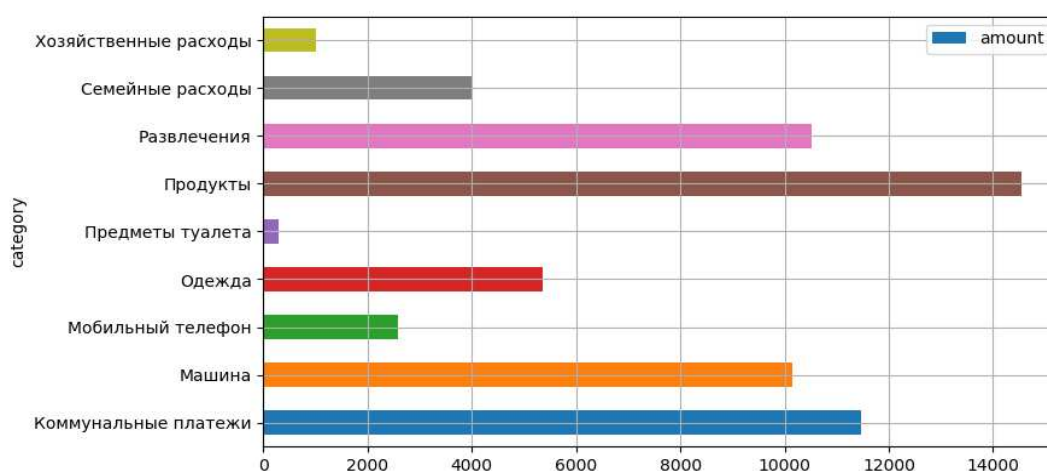


Рисунок 2– Визуализация расходов по категориям

Чтобы отобразить расходы по определенной категории (например, для категории «Развлечения») ежемесячно для начала необходимо отфильтровать DataFrame по нужному значению нужной колонки, далее с помощью функции `resample()` переформировать DataFrame ежемесячно. Далее все, как и в предыдущем примере.

Таким образом, библиотека `pandas` поддерживает все самые популярные форматы хранения данных, позволяет их анализировать по различным параметрам и визуализировать полученные результаты. Использование данного программного продукта способствует эффективному решению задачи учета личных расходов пользователя.

ЛИТЕРАТУРА

1. Документация `pandas` [Электронный ресурс]. – Режим доступа: <https://pandas.pydata.org/pandas-docs/stable/index.html> – (Дата обращения: 15.04.2019).

УДК 004.056.55

Студ. К.А. Ермаков

Науч. рук. проф., д.т.н. П.П. Урбанович

(кафедра информационных систем и технологий, БГТУ)

ОСОБЕННОСТИ ПРАКТИЧЕСКОЙ РЕАЛИЗАЦИИ КВАНТОВОЙ КРИПТОГРАФИИ НА ОСНОВЕ ПРОТОКОЛА БЕННЕТА

Основной задачей криптографии является зашифрование/расшифрование передаваемых данных. К другим задачам относятся: генерация и распределение ключей, аутентификации сторон и т. п. [1].

Задача согласования ключевой информация, предложенная Диффи и Хеллманом, относится к классическим методам асимметричной криптографии. Согласование ключей может основываться и на технологиях искусственных нейронных сетей [2].

В работе рассматриваются основные идеи относительно нового направления – квантовой криптографии (КК). Задача КК – секретное распределение симметричных (одинаковых) ключей между легитимными пользователями. В отличие от традиционной криптографии, опирающейся на математические методы, квантовая криптография основывается на фундаментальных законах квантовой механики [3, 4]. При попытке «подслушать»/измерить квантовое состояние оно неизбежно меняется, а, значит, если кто-то попытается скопировать ключ во время его распределения, об этом станет известно легитимным пользователям.

Простейший алгоритм генерации секретного ключа – протокол BB84 – первый протокол квантового распределения ключа, который был предложен в 1984 году Ч. Беннетом и Ж. Brassаром [4]. Протокол использует для кодирования информации четыре квантовых состояния двухуровневой системы, формирующие два сопряжённых базиса. Носителями информации являются 2-х уровневые системы, называемые *кубитами* (квантовыми битами), например поляризационные состояния света. Состояния внутри одного базиса ортогональны, но состояния из разных базисов – попарно неортогональны. Эта особенность протокола позволяет отследить попытки «подслушивания» перехватчиком.

Протокол использует четыре квантовых состояния, образующие два базиса, например поляризационные состояния света (рис. 1). Внутри обоих базисов состояния ортогональны, но состояния из разных базисов являются попарно неортогональными, что необходимо для определения возможных попыток нелегитимного съема информации. Таким образом, носителями информации в протоколе BB84 являются фотоны, поляризованные под углами 0 , 45 , 90 , 135° . С помощью некоторого измерения можно различить только два ортогональных состояния [3]:

- фотон поляризован вертикально или горизонтально (базис плюс);
- фотон поляризован под углами 45° или 135° (базис крест).

Отличить горизонтальный фотон от фотона, поляризованного под углом 135° , невозможно.

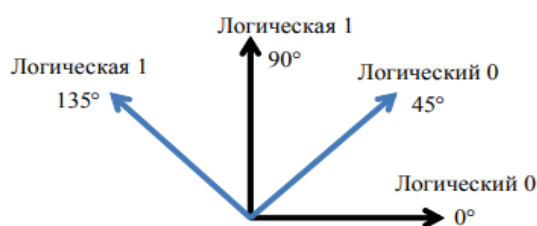


Рисунок 1 – Возможные состояния фотонов

Формирование ключей в протоколе BB84 между сторонами А и В состоит из следующих шагов:

1. А случайным образом выбирает один из базисов (на рис. 1 обозначены в различных оттенках). Затем внутри базиса случайно выбирает одно из состояний, соответствующее 0 или 1, и посылает фотоны (рис. 2). Они могут посылааться все вместе или один за другим, но главное, чтобы А и В смогли установить взаимно однозначное соответствие между посланным и принятым фотоном.

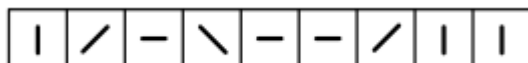


Рисунок 2 – Фотоны с различной поляризацией

2. В случайно и независимо от А выбирает для каждого поступающего фотона (рис. 3): прямолинейный (+) или диагональный (×) базис.

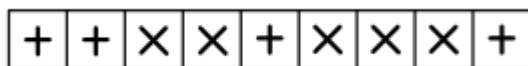


Рисунок 3 – Выбранный тип измерений

Затем В сохраняет результаты измерений (рис. 4).

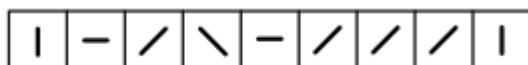


Рисунок 4 – Результаты измерений

3. В по открытому общедоступному каналу связи сообщает, какой тип измерений был использован для каждого фотона, то есть какой был выбран базис, но результаты измерений остаются в секрете.

4. А сообщает В по открытому общедоступному каналу связи, какие измерения были выбраны в соответствии с исходным базисом А (рис. 5).



Рисунок 5 – Случаи правильных замеров

5. Далее пользователи оставляют только те случаи, в которых выбранные базисы совпали. Эти случаи переводят в биты (0 и 1), и получают, таким образом, ключ (рис. 6).

			\	-		/		
1			1	0		0		1

Рисунок 6 – Получение ключевой последовательности по результатам правильных замеров

В таком случае примерно 50% данных выбрасывается. В случае отсутствия подслушивания и шумов в канале связи А и В будут теперь иметь полностью коррелированную строку случайных битов. Если же подслушивание имело место, то А и В должны будут отбросить все свои данные и начать повторное выполнение первичной квантовой передачи. В противном случае они оставляют прежнюю поляризацию. Согласно принципу неопределённости, криптоаналитик (Е) не может измерить как диагональную, так и прямоугольную поляризацию одного и того же фотона. Даже если им будет произведено измерение для

какого-либо фотона и затем этот же фотон будет переслан В, то в итоге количество ошибок намного увеличится, и это станет заметно Алисе. Это приведет к тому, что А и В будут полностью уверены в состоявшемся перехвате фотонов. Если расхождений нет, то биты, использованные для сравнения, отбрасываются, ключ принимается. С вероятностью $1-2^{-k}$ (где k — число сравненных битов) канал не прослушивался. Существует оценка, что если ошибка в канале меньше приблизительно 11%, то секретная передача данных возможна.

Квантовая криптография позволяют создавать системы, обеспечивающие практически 100%-ю защиту ключевой информации.

ЛИТЕРАТУРА

1. Урбанович, П. П. Защита информации методами криптографии, стеганографии и обфускации : учеб.-метод. пособие для студ. – Минск: БГТУ, 2016. – 220 с.

2. Плонковски, М. Криптографическое преобразование информации на основе нейросетевых технологий / М. Плонковски, П. П. Урбанович // Труды БГТУ. Серия VI. Физико-математические науки и информатика. – Минск: БГТУ. – 2005. – Вып. XIII. – С. 161–164.

3. Лесовик Г. Б. Фундаментальные проблемы физики квантовых технологий. [Электронный ресурс]. – Режим доступа:

<https://mipt.ru/education/chairs/fpftk/downloads/edumaterials/presentations/presentation.pdf>. – Дата доступа: 15.04.2019.

4. Wikipedia. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/BB8> - Дата доступа: 15.04.2019.

УДК 004.056

Студ. М. Н. Карпович, Е. В. Карпович

Науч. рук. проф. П. П. Урбанович

(кафедра информационных систем и технологий, БГТУ)

ШИФРОВАНИЕ ИНФОРМАЦИИ В БЕСПРОВОДНЫХ СЕТЯХ. УЯЗВИМОСТИ АЛГОРИТМА «РУКОПОЖАТИЯ»

Технология WPA является развитием технологий WEP [1]. Описать эту технологию можно с помощью формальной записи: WPA = 802.1X + EAP + TKIP + MIC, где 802.1X – набор стандартов связи для коммуникации в беспроводной локальной сетевой зоне, EAP – расширяемый протокол аутентификации, TKIP – протокол целостности временного ключа, MIC – проверка целостности сообщений.

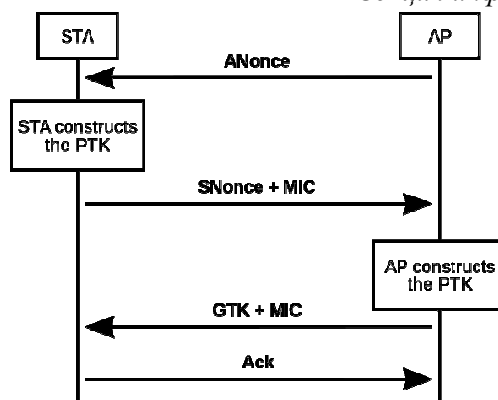


Рисунок 1 – Схема 4-х этапного «рукопожатия»

4-х этапное «рукопожатия» (рис. 1) работает следующим образом. Точка доступа (AP) отправляет клиенту (STA) временное значение PTK, в свою очередь использует Pre-SharedKey и пять других параметров — SS-ID, AuthenticatorNonce (ANonce или временное значение), SNonce, MAC-адрес точки доступа и MAC-адрес wifi-клиента. Этот ключ в дальнейшем использует шифрование между точкой доступа (AP) и wifi-клиентом.

Удалось на практике доказать, что злоумышленник способен заставить жертву сбрасывать счётчики путём повторной отправки сообщения 3-го этапа во время 4-х этапного «рукопожатия».

В обычной WPA/WPA2 PSK атаке по словарю, злоумышленник будет использовать словарь с программой. Программа будет выводить 256-bit Pre-SharedKey для каждой парольной фразы и использовать ее с другими параметрами, которые были описаны в создании PTK. PTK будет использован для проверки MessageIntegrityCheck (MIC) в одном из пакетов 4-х этапного «рукопожатия». Если они совпадут, то парольная фраза в словаре будет верной, в противном случае неверной. Именно так работает WPA/WPA2 PSK. Нашей главной задачей будет выбор цели для получения доступа и перехват EAPOL пакетов, из которых впоследствии мы получим нужную нам информацию [2].

На рис.1-8 последовательно представлены скриншоты и скрипты, соответствующие отдельным шагам анализируемой технологии.

MAC	SSID	AP	Channel	Auth	Encryption	Signal	Mode
3C:95:09:1B:1C...		STA					
8C:6D:77:DF:B1:01	byfly WIFI	AP	802.11n	WPA-CCMP	-13/-14/-...	12.2	1
58:FB:84:35:E0...		STA					
72:87:AA:57:E9:D7	mixershpixer	AP	802.11n	WPA-CCMP	-71/-45/-...	72.2	1
8C:6D:77:DF:B1:00	Alesha	AP	802.11n	WPA-CCMP,W...	-83/-49/-...	144.0	2
EC:D0:9F:16:57:...		STA					
F8:28:19:46:EC:67		STA					
A6:8E:0C:60:DF:C5	iPhone (PiPiCsB)	AP	802.11n	WPA-CCMP	-82/-58/-...	72.2	1
74:C6:3B:5A:96:...		STA					
74:C1:4F:70:F2:85	Honor 9 Lite	AP	802.11n	WPA-CCMP	-81/-60/-...	72.2	1
18:F0:E4:2F:FB:08	AndroidAP	AP	802.11n	WPA-CCMP	-85/-61/-...	72.2	1
B8:81:98:4D:D5:...		STA					
D8:63:75:2E:FD:09	Redmi	AP	802.11n	WPA-CCMP	-68/-62/-...	72.2	1
SamsungE:13:7...		STA					
8C:6D:77:FA:6A:F0	HUAWAI-3Z5G	AP	802.11n	WPA-CCMP,W...	-72/-63/-...	144.0	2
16:33:65:22:99:C3	PSP5509DUO	AP	802.11n	WPA-CCMP	-88/-63/-...	72.2	1
Comcast:5C:8...		STA					

Рисунок 2 – Этап перевода адаптера в режим прослушивания, выбор нужного объекта

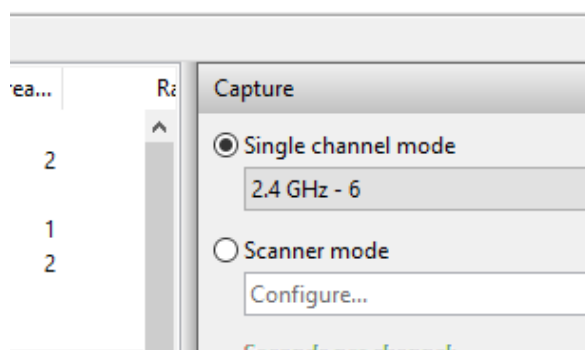


Рисунок 3 – Адаптер на прослушивание канала на котором находится объект

Программа предназначена для работы с дампами, содержащими пакеты для последующего взлома. Также она используется для:

- мониторинга: захват пакетов и экспорт данных в текстовые файлы для дальнейшей обработки сторонними инструментами,
- проведения атак: повторные атаки, деаутентификация, фальшивые точки доступа и другие через внедрение пакетов,
- тестирования: проверка WiFi-карт и возможностей драйвера (захват и внедрение),
- взлома: WEP и WPA PSK (WPA 1 и 2).

Для решения этой задачи были выпущены специальные «патчи» для операционных систем, а так же анонсированный в 2018 году WPA3. Он будет дополнен встроенной защитой от брутфорс-атак [2], индивидуальным шифрованием данных для усиления конфиденциальности пользователей в открытых Wi-Fi сетях, упрощенная настройка IoT-устройств, а так же усовершенствованный криптографический стандарт для сетей Wi-Fi, который будет иметь «192-разрядный пакет безопасности» [3].

No	Protocol	Src MAC	Dest MAC	Src IP	Dest IP	Src Port	Dest Port	Time	Signal	Rate	More details
17543	MNGT/B...	A6:8E:0C:60...	Broadcast	?/N/A	?/N/A	N/A	N/A	23:42:...	-47	1	SSID=iPhone (P+PiPsCб), (Infra...
17544	MNGT/P...	A6:8E:0C:60...	DA:A1:19:42:E7:B4	?/N/A	?/N/A	N/A	N/A	23:42:...	-46	1	SSID=iPhone (P+PiPsCб), (Infra...
17545	MNGT/P...	A6:8E:0C:60...	DA:A1:19:42:E7:B4	?/N/A	?/N/A	N/A	N/A	23:42:...	-48	1	SSID=iPhone (P+PiPsCб), (Infra...
17546	MNGT/P...	DA:A1:19:4...	Broadcast	?/N/A	?/N/A	N/A	N/A	23:42:...	-55	1	SSID=any, Seq=658
17547	MNGT/P...	A6:8E:0C:60...	DA:A1:19:42:E7:B4	?/N/A	?/N/A	N/A	N/A	23:42:...	-49	1	SSID=iPhone (P+PiPsCб), (Infra...
17548	MNGT/P...	DA:A1:19:4...	Broadcast	?/N/A	?/N/A	N/A	N/A	23:42:...	-60	1	SSID=any, Seq=659
17549	MNGT/P...	A6:8E:0C:60...	DA:A1:19:42:E7:B4	?/N/A	?/N/A	N/A	N/A	23:42:...	-46	1	SSID=iPhone (P+PiPsCб), (Infra...
17550	MNGT/P...	A6:8E:0C:60...	DA:A1:19:42:E7:B4	?/N/A	?/N/A	N/A	N/A	23:42:...	-45	1	SSID=iPhone (P+PiPsCб), (Infra...
17551	MNGT/P...	18:F0:E4:2F...	DA:A1:19:42:E7:B4	?/N/A	?/N/A	N/A	N/A	23:42:...	-53	1	SSID=AndroidAP, (Infra.), Ch.#6...
17552	MNGT/B...	A6:8E:0C:60...	Broadcast	?/N/A	?/N/A	N/A	N/A	23:42:...	-50	1	SSID=iPhone (P+PiPsCб), (Infra...
17553	MNGT/B...	16:33:65:22...	Broadcast	?/N/A	?/N/A	N/A	N/A	23:42:...	-52	1	SSID=PSP5509DUO, (Infra.), Ch...
17554	MNGT/B...	EE:D0:9F:CA...	Broadcast	?/N/A	?/N/A	N/A	N/A	23:42:...	-71	1	SSID=Mi Phone, (Infra.), Ch.#6, ...
17555	MNGT/B...	A6:8E:0C:60...	Broadcast	?/N/A	?/N/A	N/A	N/A	23:42:...	-48	1	SSID=iPhone (P+PiPsCб), (Infra...
17556	MNGT/P...	A6:8E:0C:60...	DA:A1:19:42:E7:B4	?/N/A	?/N/A	N/A	N/A	23:42:...	-44	1	SSID=iPhone (P+PiPsCб), (Infra...
17557	MNGT/P...	A6:8E:0C:60...	DA:A1:19:42:E7:B4	?/N/A	?/N/A	N/A	N/A	23:42:...	-46	1	SSID=iPhone (P+PiPsCб), (Infra...
17558	MNGT/P...	A6:8E:0C:60...	DA:A1:19:42:E7:B4	?/N/A	?/N/A	N/A	N/A	23:42:...	-44	1	SSID=iPhone (P+PiPsCб), (Infra...
17559	MNGT/B...	A6:8E:0C:60...	Broadcast	?/N/A	?/N/A	N/A	N/A	23:42:...	-44	1	SSID=iPhone (P+PiPsCб), (Infra...
17560	MNGT/B...	16:33:65:22...	Broadcast	?/N/A	?/N/A	N/A	N/A	23:42:...	-53	1	SSID=PSP5509DUO, (Infra.), Ch...
17561	MNGT/P...	A6:8E:0C:60...	DA:A1:19:42:E7:B4	?/N/A	?/N/A	N/A	N/A	23:42:...	-53	1	SSID=iPhone (P+PiPsCб), (Infra...
17562	MNGT/P...	A6:8E:0C:60...	DA:A1:19:42:E7:B4	?/N/A	?/N/A	N/A	N/A	23:42:...	-53	1	SSID=iPhone (P+PiPsCб), (Infra...

Рисунок 4 – Таблица полученных пакетов

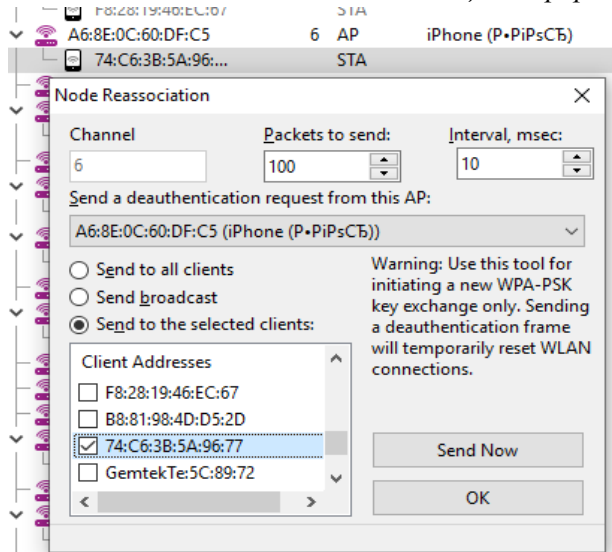


Рисунок 5 – Реассоциация узлов

- 1 SSID=FUAWVEI-4V8, (Infra.), Ch.#3, Seq=2272, BI=100
- 1 SSID=Lenovo A6010_A, (Infra.), Ch.#6, Seq=3173, BI=100
- 1 EAPOL-Key (4-Way Handshake Message 1), Length = 95
- 1 EAPOL-Key (4-Way Handshake Message 1), Length = 95
- 1 SSID=Mason, (Infra.), Ch.#6, Seq=188, BI=100
- 72 WPA: Can't decrypt
- 1 SSID=Lenovo A6010_A, (Infra.), Ch.#6, Seq=3174, BI=100
- 1 SSID=PSP5509DUO, (Infra.), Ch.#5, Seq=1089, BI=100
- 1 EAPOL-Key (4-Way Handshake Message 1), Length = 95
- 1 EAPOL-Key (4-Way Handshake Message 1), Length = 95
- 1 SSID=Mason, (Infra.), Ch.#6, Seq=189, BI=100

Рисунок 6 – Таблица EAPOL пакетов

Далее используется программа aircrack-ng GUI.

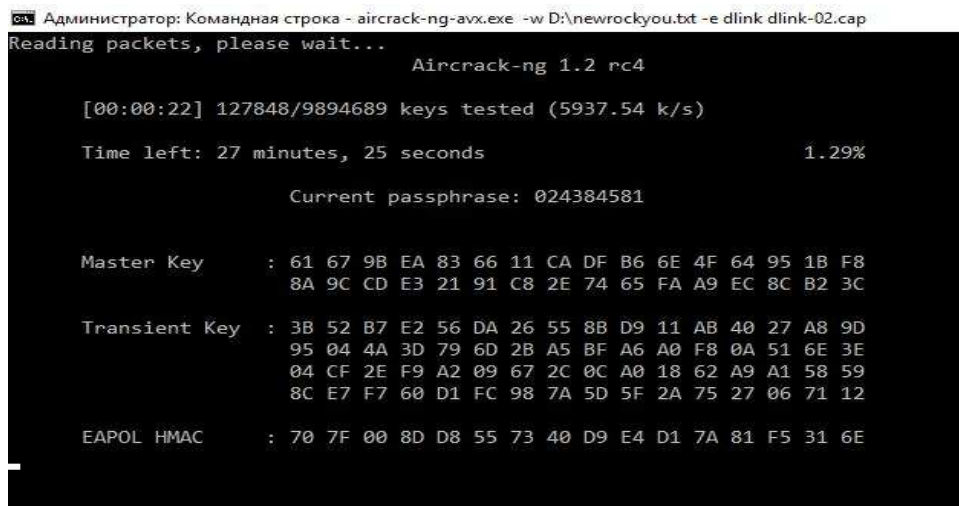


Рисунок 7 – Окно программы aircrack-ng

```

KEY FOUND! [ qwerty123 ]

Master Key      : ED FF 23 E4 68 76 CE C7 BA A3 49 9F 8E 3C 32 CD
                  44 96 43 9E DC 30 C7 5E 46 73 F0 40 CE 2B BE C8

Transcient Key  : 7F 5E AD AE 34 94 39 AD AC 4E C5 64 B2 7B FE CF
                  28 CF F5 1D 64 AC 5B BA B5 CB F6 25 11 0B 3C F4
                  C1 A6 8B 85 E3 68 C1 97 27 4C 03 65 D1 67 E3 1E
                  67 22 1E 51 9A 6D 49 67 F6 5F BE 47 4A 7A D6 51

EAPOE HMAC     : 36 33 8C 56 61 C7 9C 93 9D 47 A4 B5 A4 1F 05 F4

```

Рисунок 8 – Результат успешного взлома

ЛИТЕРАТУРА

1. Урбанович, П. П. Компьютерные сети: учебное пособие для студентов высших учебных заведений по техническим специальностям / П. П. Урбанович, Д. М. Романенко, Е. В. Кабак. – Минск: БГТУ, 2011. – 399 с.
2. Урбанович, П. П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие для студ./ П.П. Урбанович. – Минск: БГТУ, 2016. – 220 с.
3. Нефёдова, М. Анонсирован стандарт WPA3, Wi-Fi обещают сделать безопаснее / М. Нефёдова. [Электронный ресурс]. – Режим доступа: <https://хакер.ru/2018/01/09/wpa3/>. – Дата доступа: 10.04.2019.

УДК004.056

Магистрант А. А. Сущеня

Науч. рук. проф. П. П. Урбанович

(кафедра информационных систем и технологий, БГТУ)

СТЕГАНОГРАФИЧЕСКИЙ МЕТОД ВНЕДРЕНИЯ ТЕКСТОВОЙ ИНФОРМАЦИИ В КОНТЕЙНЕР ФОРМАТА EPUB

Актуальность исследования стеганографии обострена проблемами защиты цифрового контента от несанкционированного использования. Ввиду этого существует необходимость расширения и углубления теоретической базы стеганографии как платформы для тайной передачи и хранения информации [1].

С развитием информационных технологий стали появляться цифровые объекты, имеющие в своей структуре избыточность, которую можно использовать для реализации стеганографических методов передачи данных. Одними из таких объектов являются электронные книги. На сегодняшний день, наиболее популярным форматом электронных книг являются EPUB. Рассмотрим структуру данного формата. EPUB — это ZIP-файл, сжатый особым образом. В этом можно

убедиться, изменив расширение файла .epub на .zip и разархивировав его любым архиватором.

Минимальный набор файлов, который входит в EPUB должен быть следующим:

- файл `mimetype`, сострочкой «`application/epub+zip`»;
- папка `META-INF` с файлом `container.xml`, который указывает, где хранится содержимое книги;
- папка `OPS`, в которой должны быть: файл с метаданными книги, списком всех файлов, которые нужны для ее содержимого, описанием последовательности чтения файлов и путеводителем по ключевым файлам; файл `toc.ncx`, содержащий оглавление книги в том виде, в котором оно будет прочитано программой для чтения; файл `stylesheet.css` с описанием стилей оформления текста; файлы содержимого в формате XHTML; папки с иллюстрациями и шрифтами.

XHTML — это основанный на XML язык разметки гипертекста, максимально приближенный к стандартам HTML. XHTML отличается от HTML строгостью написания кода. Если HTML позволяет писать практически любые конструкции и браузер их корректно распознавал, то, с появлением XHTML это стало невозможным. Последний требует строгого соблюдения всех правил, предъявляемых W3C. Строгие требования к оформлению XHTML-кода позволяют избежать многих ошибок ещё на стадии написания и отладки.

Для защиты авторского права на электронную книгу формата EPUB предлагается внедрять метку, содержащую уникальный идентификатор пользователя в системе распространения, а также международный стандартный книжный номер. Значение метки предварительно шифруется и кодируется в бинарную последовательность.

Исходя из того, что книга состоит из определенного количества глав, закодированная последовательность последовательно внедряется в эти главы (рис. 1). Для внедрения информации в XHTML файлы выбран метод кавычек [2-6]. Данный метод позволяет размещать двоичную информацию в файлах языков разметки. Метод основан на замене одинарных кавычек на двойные или наоборот в файлах с разметкой, т. е. на нечувствительности языка разметки к типу кавычек. Предварительно осаждаемая бинарная последовательность шифруется. Для увеличения стеганографической стойкости системы вводится дополнительный ключ, которым является название файла главы.

При наличии дополнительного числа после знака «-» в наименовании главы, осаждение информации производится, начиная с конца файла.

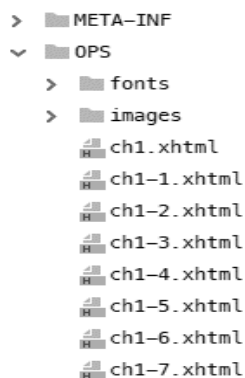


Рисунок 1 — Структура файлов электронной книги формата EPUB

Для контроля достоверности внедренной метки вводится дополнительный метод осаждения информации в электронные книги на основе использования каскадных таблиц стилей. Стил — это совокупность правил, применяемых к элементу гипертекста и определяющих способ его отображения. Стил включает все типы элементов дизайна: шрифт, фон, текст, цвета ссылок, поля и расположение объектов. Таблица стилей — это совокупность стилей, применимых к гипертекстовому документу. Каскадирование — это порядок применения различных стилей.

При создании CSS файлов существует возможность хранения изображений в формате base64. Base64 — способ кодирования произвольных двоичных данных в ASCII текст. Кодирование Base64 занимает три байта, каждый из которых состоит из восьми битов, и представляет их в виде четырех печатных символов в стандарте ASCII. Хранение картинки небольшого размера, в data:image base64 в CSS — существенно экономит количество запросов к серверу.

Для проверки осажденной метки необходимо закодировать изображение в формат base64. Далее разместить получившуюся строку в CSS файл, заменяя «ТИП» на MIME-тип изображения — JPEG/PNG/GIF или BMP и «КОД» на нужную строку в base64.

Предварительно во внедряемое изображение встраивается контрольная сумма метки, используя стеганографический метод LSB [1, 7,8]. Суть метода LSB заключается в следующем: заменяются младшие биты в байтах, отвечающих за кодирование цвета изображения. Допустим, если очередной байт секретного сообщения – 11001011, а байты в изображении –...11101100 01001110 01111100 0101100111..., то кодирование будет выглядеть так: байт секретного сообщения разбивается на 4 двухбитовые части: 11, 00, 10, 11. После чего заменяем полученными фрагментами младшие биты изображения: ...11101111 01001100 01111110 0101100111.... Такая замена в общем случае не заметна человеческому глазу. Можно менять не только два младших

бита, но и любое их количество. Однако есть следующая закономерность: при замене большего количества бит, теряется больший объем информации, следовательно, факт наличия закодированного сообщения обнаружить проще.

ЛИТЕРАТУРА

1. Урбанович П.П. Защита информации методами криптографии, стеганографии и обфускации/ П.П. Урбанович. – Минск: БГТУ, 2016. – 220 с.
2. Сушня, А. А. Стеганографическое преобразование текстов-контейнеров на основе языков разметки / А. А. Сушня // 68-я научно-техническая конференция учащихся, студентов и магистрантов, 17-22 апреля, Минск: сборник научных работ: в 4 ч., Ч. 4 / Белорусский государственный технологический университет. – Минск: БГТУ, 2017.– С. 145-149.
3. Сушня, А. А. Способ стеганографического осаждения информации в документ с расширением .DOCX / А. А. Сушня // XXI Республиканская научная конференция студентов и аспирантов, 19–21 марта, Гомель: сборник научных работ / Гомельский государственный университет имени Ф. Скорины. – С. 303-304.
4. Сушня, А.А. Идея и архитектура веб-приложения, использующего в качестве стеганографического контейнера документы формата DOCX / А. А. Сушня // Международная научно-практическая конференция, 14–18 мая, Минск: сборник научных работ / Белорусский государственный университет. – С. 170.
5. Сушня, А.А. Модификация стеганографического метода изменения междустрочного расстояния электронного документа/ А.А. Сушня, Е.А. Блинова, П.П. Урбанович// Технические средства защиты информации: Тезисы докладов XVI Белорусско-российской научно-технической конференции, 5 июня 2018 г., Минск. – Минск: БГУИР, 2018. – С 90-91.
6. Сушня, А. А. Программное средство стеганографического преобразования текстов-контейнеров на основе языка разметки XML / А. А. Сушня // 69-я научно-техническая конференция учащихся, студентов и магистрантов, 2-13 апреля, Минск: сборник научных работ: в 4 ч., Ч. 4 / Белорусский государственный технологический университет. – Минск: БГТУ, 2018. –С. 81-84.
7. Урбанович, П. П. Стеганография в графических объектах / П. П. Урбанович, Т. В. Коваленок, Н. П. Урбанович // Международная научно-техническая конференция "Автоматический контроль и авто-

матризация производственных процессов": материалы конференции, Минск, 28-29 октября 2009 г. – Минск, 2009. – С. 69-70.

8. Шутько, Н. П. Защита авторских прав на текстовые документы на основе стеганографической модификации цвета символов текста / Н. П. Шутько, П. П. Урбанович // Информационные технологии: материалы 83-й научно-технической конференции профессорско-преподавательского состава, научных сотрудников и аспирантов (с международным участием), Минск, 4-15 февраля 2019 г. / отв. за изд. И. В. Войтов; УО БГТУ. – Минск: БГТУ, 2019. – С. 41-43.

УДК 004.056+003.26

Студ. М. А. Тихонович
Науч. рук. проф. П. П. Урбанович
(кафедра информационных систем и технологий, БГТУ)

КРИПТОСТОЙКОСТЬ ГОМОМОРФНЫХ ШИФРОВ

Гомоморфное шифрование – форма шифрования, позволяющая производить определённые математические действия с зашифрованным текстом и получать зашифрованный результат, который соответствует результату операций, выполняемых с открытым текстом.

Задачи гомоморфной криптографии:

– необходимо производить вычисления над зашифрованными данными так, чтобы результат после расшифрования соответствовал результату тех же вычислений над открытыми данными (m_1 и m_2). Сформулирована в 1978 году Ривестом, Адлеманом и Дертусосом. Является основой для систем полностью гомоморфного шифрования (FHE);

– имеется 2 или более стороны. Необходимо вычислить произвольную функцию, аргументы которой являются секретными данными этих сторон, таким образом, чтобы ни одна из сторон не узнала вход другой стороны, а результат вычисления функции был корректен. Сформулирована Яо в 1982 году. Является основной задачей протоколов секретных распределённых вычислений (SMC).

Риверст и Адлеман впервые обосновали, что методом, позволяющим успешно проводить операции над зашифрованными данными, не искажая и не расшифровывая их, является так называемое гомоморфное шифрование [1].

Полностью гомоморфные схемы шифрования позволяют выполнять арифметические операции над зашифрованными данными без их предварительного расшифровывания. Частично гомоморфные

криптосистемы обладают свойством гомоморфности относительно только одной операции, например, сложения или умножения

Система шифрования является гомоморфной относительно операции умножения, если

$$D(E(m_1) \otimes E(m_2)) = m_1 * m_2.$$

Система шифрования является гомоморфной относительно операции сложения, если

$$D(E(m_1) \oplus E(m_2)) = m_1 + m_2.$$

Система шифрования является гомоморфной относительно операций умножения и сложения, т.е. полностью гомоморфной, если

$$D(E(m_1) \otimes E(m_2)) = m_1 * m_2 \text{ и } D(E(m_1) \oplus E(m_2)) = m_1 + m_2,$$

где \otimes , \oplus – операции умножения и сложения над шифрованными текстами, соответствующие операциям умножения и сложения над открытыми текстами; D – функция расшифрования; E – функция зашифрования.

Если криптосистема с такими свойствами сможет зашифровать два бита, то, поскольку операции сложения и умножения формируют над битами полный по Тьюрингу базис, становится возможным вычислить любую булеву функцию, а, следовательно, и любую другую вычислимую функцию.

Любую стандартную систему шифрования можно описать в виде трех операций: генерации ключей, зашифрования и расшифрования [2, 3]. Гомоморфная система шифрования, кроме трех перечисленных выше операций, включает в себя операцию вычислений. Таким образом, в каждой гомоморфной криптосистеме можно выделить некоторые общие алгоритмы:

- генерация секретного ключа sk и открытого ключа pk ;
- функция зашифрования Enc ;
- функция расшифрования $Decrypt$;
- гомоморфное вычисление $Eval$;

Функция зашифрования Enc представляется в следующем виде:

$$C = Enc(pk, O),$$

где O – принимаемый на вход алгоритма открытый текст; C – полученный шифротекст.

Функция расшифрования $Decrypt$ в математическом виде представляется следующим образом:

$$O = Decrypt(sk, C).$$

Гомоморфное вычисление $Eval$ оперирует некоторой математической функцией $f()$ и парой шифротекстов C_1, C_2 . Результатом вычисления будет некий шифротекст C , причем такой, что:

$$C = Eval(f(), C_1, C_2), \quad O = Decrypt(sk, C) = f(O_1, O_2).$$

Схематично алгоритм $Eval$ можно представить следующим образом [4].

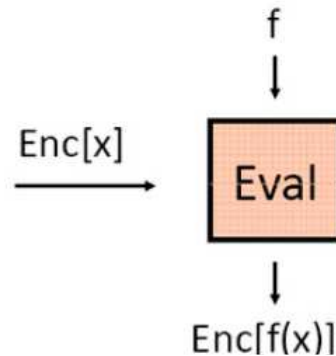


Рисунок 1 – Алгоритм Eval

На вход алгоритму подается зашифрованное сообщение $Enc(M)$ и некая математическая функция $f()$. Результатом работы алгоритма является другое зашифрованное сообщение $Enc(M_2)$, причем $M_2=f(M)$.

Операции гомоморфного шифрования:

- генерация ключей: клиент генерирует открытый ключ pk и секретный ключ sk для шифрования открытого текста;
- шифрование: используя секретный ключ sk , клиент шифрует открытый текст O , и вместе с открытым ключом pk отправляет зашифрованный текст C на сервер;
- вычисление: сервер получает функцию $f()$ для проведения вычислений над зашифрованным текстом C и выполняет их в соответствии с требованиями данной функции, используя pk ;
- расшифрование: для получения искомого результата значение $Eval(f(O))$, полученное в ходе вычислений, расшифровывается клиентом с использованием своего секретного ключа sk .

Рассмотрим частично гомоморфную схему, а именно схему гомоморфную относительно операции сложения [5]. Данная схема построена на группе вычетов целого числа N . Таким образом, закрытый ключ в данной схеме представляет собой целое число, на которое накладываются следующие условия:

- число должно быть много больше N ;
- число должно быть взаимно простым с N ;

Этапы шифрования и некоторые подробности реализации частично гомоморфной схемы относительно операции сложения:

1. **Выбор числа N .** Группа целых чисел, с которыми способна работать криптосистема генерируется как, группа вычетов по модулю N . Таким образом, в качестве открытого текста могут выступать числа от 0 до N .

2. **Генерация ключей.** Генерируется случайное число sk , такое что $sk \gg N$ и $\text{НОД}(sk, N) = 1$. В данной реализации накладывається еще одно условие: $sk \leq 2^{31}$, т.к. при $sk > 2^{31}$ число выходит за границу диапазона структуры данных `Int32`.

Открытым ключом pk является набор больших чисел $\{a_1, a_2, \dots, a_n\}$ таких что $a_i \bmod N = 2e_i$, где $e_i \ll N$. В данной реализации N является константой.

3. **Зашифрование.** Функция зашифрования принимает на вход открытый текст O и открытый ключ pk . Из набора pk случайным образом выбирается 50 элементов a_i и суммируются с O .

4. **Дешифрование.** Алгоритм дешифрования принимает на вход шифротекст C и секретный ключ sk . Открытый текст $O = (C \bmod sk) \bmod N$.

5. **Операции над шифротекстом.** Как было сказано ранее, описанная схема является гомоморфной только относительно операции сложения. Поэтому, для того чтобы произвести операцию над зашифрованными данными, достаточно просто сложить или вычесть два шифротекста.

Плюсом данной гомоморфной системы относительно других гомоморфных систем является малая потребность в вычислительных ресурсах.

Минусы данной гомоморфной системы относительно других гомоморфных систем.

1. Данная схема оперирует с числами из группы вычетов некоторого числа N , что означает, что множество открытых текстов – конечное.

2. Описанная схема – частично гомоморфная, поэтому в ней доступна лишь операция сложения.

Свыше 30 лет оставалась нерешенной задача полностью гомоморфного шифрования – создания системы, гомоморфной относительно операций сложения и умножения одновременно. Только в 2009 г. аспирант Стэнфордского университета и стажер IBM Крейг Джентри теоретически обосновал принципиальную возможность создания такой системы шифрования. В схеме Джентри [6] выполняются свойства гомоморфизма как относительно умножения, так и сложения, т. е. она является алгебраической гомоморфной системой.

Следует учитывать, что некоторые гомоморфные криптосистемы могут поддаваться преднамеренным внешним воздействиям (например, принципиально уязвимы к атаке с адаптивно подобранным шифрованным текстом) и поэтому не всегда подходят для безопасной передачи данных в компьютерных системах и сетях [7].

ЛИТЕРАТУРА

1. Rivest, R.L. On data banks and privacy homomorphisms / R.L. Rivest, L. Adleman, M.L. Dertouzos // Foundations of secure computation. – 1978. – Vol. 32, no. 4. – P. 169–178.
2. Урбанович, П. П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие для студ./ П.П. Урбанович. – Минск: БГТУ, 2016. – 220 с.
3. Урбанович, П. П. Информационная безопасность и надежность систем: учебно-методическое пособие по одноименному курсу для студентов специальности 1-40 01 02-03 "Информационные системы и технологии" / П. П. Урбанович, Д. М. Романенко, Е. В. Романцевич. – Минск : БГТУ, 2007. – 87 с.
4. Гомоморфное шифрование своими руками. [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/150067/>. – Дата доступа: 12.04.2019.
5. Gentry G.A Fully Homomorphic Encryption Scheme, 2009 – P. 71-80.
6. Gentry, C. A Fully homomorphic encryption using ideal lattices / C. Gentry // Symposium on the Theory of Computing (STOC). – Bethesda, USA, 2009. – P. 169–178.
7. Урбанович, П. П. Компьютерные сети: учебное пособие для студентов высших учебных заведений по техническим специальностям / П. П. Урбанович, Д. М. Романенко, Е. В. Кабак. – Минск: БГТУ, 2011. – 399 с.

УДК 004.056+003.26

Студ. А. Г. Савенко, А. Д. Черняк
Науч. рук. проф. П. П. Урбанович
(кафедра информационных систем и технологий, БГТУ)

РЕАЛИЗАЦИЯ «ЛЕГКОВЕСНЫХ» КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ ДЛЯ УСТРОЙСТВ С НЕБОЛЬШОЙ ВЫЧИСЛИТЕЛЬНОЙ МОЩНОСТЬЮ

Главной особенностью современного этапа развития интернета является всевозрастающее количество самых различных вычислительных устройств, имеющих доступ в интернет. В силу условий их функционирования, а также жестких ценовых ограничений, свойственных массовому производству, эти устройства характеризуются значительными ограничениями на используемые ресурсы памяти, вычислительную мощность, источники питания и так далее. Отсюда следуют ограничения на используемые технологии и технологические решения, предъявляемые к средствам низкоресурсной криптографии. Так, например, жесткие ограничения накладываются на энергозатратность реализации криптографических алгоритмов для пассивных интеллектуальных устройств таких, как радиочастотные метки или бесконтактные смарт-карты.

Типичными ограничениями, встречающимися в низкоресурсной криптографии, являются:

- размер микросхемы, потребляемая энергия, время, затраченное на исполнение программы (для аппаратной реализации);
- размер программного кода, размер оперативной памяти, время, затраченное на исполнение программы (для программной реализации).

Могут появляться и другие ограничения. Так, в зависимости от конкретных условий применения разрабатываемого средства важной может оказаться такая характеристика, как ширина полосы рабочих частот канала связи [1].

Для понятия назначения и области применения легковесных (малоресурсных) криптографических алгоритмов, необходимо остановиться на понятии «Интернет вещей». Итак, «Интернет вещей» – это беспроводная самоконфигурирующаяся сеть между объектами типа бытовых приборов, транспортных средств, различных сенсоров и датчиков, а также меток радиочастотной идентификации (RFID).

В связи с этим, особенно актуальной становится задача эффективной реализации алгоритмов защиты информации, обеспечивающих конфиденциальность и целостность данных. Очевидно, что основу такой безопасности должны образовывать криптографические методы

защиты информации. Массовый характер применения и небольшие потоки передаваемых данных привели к необходимости использования алгоритмов «легковесной» криптографии, для реализации в устройствах, имеющих ограниченные вычислительные возможности.

К реализации легковесной криптографии предъявляются определенные требования: низкая потребляемая энергия; реализация должного уровня безопасности; малые размеры микросхемы; обработка небольших потоков информации с приемлемым быстродействием; низкая цена устройства. При этом криптостойкость алгоритмов не должна значительно снижаться.

Целью для разработчика является нахождение баланса между производительностью, безопасностью и ценой.

Криптографические примитивы являются основным криптографическим инструментарием, который обеспечивает выполнение тех или иных криптографических сервисов. Обычно их подразделяют на примитивы с секретным ключом (симметричные примитивы), примитивы с открытым ключом (асимметричные примитивы) и бесключевые примитивы [2].

Как упоминалось ранее, одной из технологий, использующей легковесную криптографию, является RFID. RFID – это метод автоматической идентификации объектов, в котором посредством радиосигналов считываются или записываются данные, хранящиеся в так называемых RFID-метках.

Любая RFID-система состоит из считывающего устройства и RFID-метки [3]. Большинство RFID-меток состоит из двух частей: интегральной схемы для хранения и обработки информации и антенны для приёма и передачи сигнала. По типу источника питания RFID-метки делятся на следующие категории:

1. Пассивные;
2. Активные;
3. Полупассивные.

Пассивные RFID-метки не имеют встроенного источника энергии. Электрический ток, индуцированный в антенне электромагнитным сигналом от считывателя, обеспечивает достаточную мощность для функционирования кремниевого чипа, размещённого в метке, и передачи ответного сигнала.

Активные RFID-метки обладают собственным источником питания и не зависят от энергии считывателя, вследствие чего они читаются на дальнем расстоянии, имеют большие размеры и могут быть оснащены дополнительной электроникой. Однако, такие метки наиболее дороги, а у батарей ограничено время работы.

Полупассивные RFID-метки, также называемые полуактивными, очень похожи на пассивные метки, но оснащены батареей, которая обеспечивает чип энергоснабжением.

Существуют как блочные, так и потоковые легковесные алгоритмы. В активных RFID-системах активно используются потоковые алгоритмы шифрования [2]. На данный момент, известно четыре описанных поточных LW-шифра, имеющих относительно приемлемые характеристики. Это алгоритмы Mickey, Bean, Trivium и Grainv1. На алгоритме Grain v1 мы остановимся подробнее.

Grain v1 – это симметричный алгоритм синхронного потокового шифрования, ориентированный, в первую очередь, на аппаратную реализацию. Схема шифрования состоит из трех блоков:

1. Регистр сдвига с линейной обратной связью (LFSR).
2. Регистр сдвига с нелинейной обратной связью (NFSR).
3. Выходной функции $h(x)$.

Следует пояснить, что такое регистр сдвига с обратной связью – это последовательность бит, которая на каждом такте шифрования сдвигается вправо на один разряд, при этом выход из крайнего правого бита является выходом генератора, а на вход крайнего левого бита подается значение, вычисляемое как некоторая функция от отдельных битов регистра. Ключ шифрования поточного шифра заносится в регистр перед началом генерации потока бит.

Изначально происходит загрузка 80 битов ключа в NFSR, 64-битного вектора инициализации, который дополняется 16 единицами, в LFSR. После 160 холостых тактов, во время которых будет происходить выполнение функций $f(x)$ на LFSR и $g(x)$ на NFSR, регистры полностью обновятся:

$$f(x) = 1 + x^{18} + x^{29} + x^{42} + x^{57} + x^{67} + x^{80}, \quad (1)$$

$$g(x) = 1 + x^{18} + x^{20} + x^{28} + x^{35} + x^{43} + x^{47} + x^{52} + x^{59} + x^{66} + x^{71} + (x^{80} + x^{17}x^{20}) + \\ + (x^{43}x^{47} + x^{65}x^{71}) + (x^{20}x^{28}x^{35}) + (x^{47}x^{52}x^{59}) + (x^{17}x^{35}x^{52}x^{71}) + (x^{20}x^{28}x^{43}x^{47}) + \\ + (x^{17}x^{20}x^{59}x^{65}) + (x^{17}x^{20}x^{28}x^{35}x^{43}) + (x^{47}x^{52}x^{59}x^{65}x^{71}) + (x^{28}x^{35}x^{43}x^{47}x^{52}x^{59}). \quad (2)$$

Далее аргументами функции $h(x)$ будут выступать 4 бита из LFSR и 1 из NFSR. По итогу на выход поступает бит, в процессе вычисления которого участвует результат функции $h(x)$ и сумма битов из NFSR.

Структура шифра Grainv1 представлена на рисунке 1. Шифр разработан таким образом, что невозможна атака быстрее, чем полный перебор ключей. Таким образом, лучшая атака должна иметь сложность порядка 2^{80} .

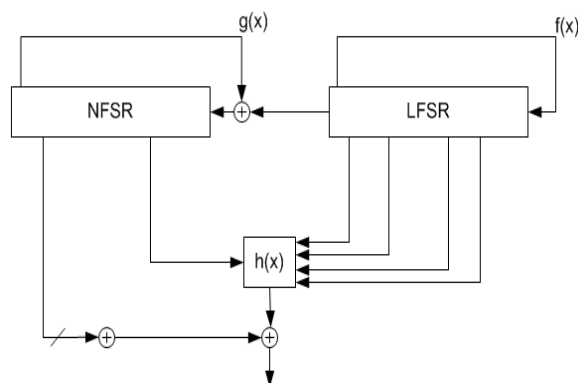


Рисунок 1 – Структура шифра Grainv1

Пропускная способность Grainv1, в сравнении с другими алгоритмами, довольно большая. Результаты аппаратной реализации различных потоковых шифров представлены в таблице 1.

Таблица 1 – Сравнение результатов аппаратной реализации различных потоковых шифров*

Алгоритм	Nk	Тактов на инициализацию	GE	Cl/byte
Enocoro-128v2		4869.5	4100	46.3
Grain v1	80	321	1294 2599	31.2
Mickey-128 v2		56592.1	5000	1231.4
Trivium	80	1333	749 2599 3488 5504 1294	8.0 0.125
AES-CTR		469.6		17.8

* – Nk – длина ключа (в битах), GE – условные логические элементы (Gate Equivalent), Cl/byte – число тактов работы алгоритма на байт зашифрованной информации – мера скорости работы алгоритма, пропускная способность.

Рассмотренные криптоалгоритмы и их реализация строятся и применяются для обеспечения информационной безопасности соответствующих систем с учетом их взаимодействия с ИНТЕРНЕТ и сетевыми технологиями [4].

ЛИТЕРАТУРА

1. Жукова, А.Е. Легковесная криптография. Части 1 и 2/ А.Е. Жукова. [Электронный ресурс]. – Режим доступа: www.cyberrus.com. – Дата доступа: 28.03.2019.

2. Урбанович, П. П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие для студ./ П.П. Урбанович. – Минск: БГТУ, 2016. – 220 с.

3. LW-криптография: шифры для RFID-систем. [Электронный ресурс]. – Режим доступа: www.habr.com. – Дата доступа: 03.04.2019.

4. Урбанович, П. П. Компьютерные сети: учебное пособие для студентов высших учебных заведений по техническим специальностям / П. П. Урбанович, Д. М. Романенко, Е. В. Кабак. – Минск: БГТУ, 2011. – 399 с.

УДК 004.056+003.26

Студ. А.А. Чопик

Науч. рук. проф. П.П. Урбанович

(кафедра информационных систем и технологий, БГТУ)

РЕАЛИЗАЦИЯ СЕРВЕРА ДЛЯ ПРЕДОСТАВЛЕНИЯ УДОБНОГО ИНТЕРФЕЙСА К ДАННЫМ БЛОКЧЕЙНА ЕСНО И ВОЗМОЖНОСТИ ПОДПИСАТЬСЯ НА ИХ ОБНОВЛЕНИЕ

В современном мире широкое распространение получили криптовалюты и технология блокчейн (Blockchain), которая лежит в основе большинства из них. По своей сути блокчейн – это реестр, представляющий собой неразрывную цепочку из блоков, содержащих транзакции с информацией и расположенную на множестве узлов независимо друг от друга [1]. Под транзакциями понимаются любые действия, совершаемые участниками сети (отправка денежных средств, покупка какого-либо контента, установление прав собственности и др.). Кроме новой информации в блоке хранится зашифрованная информация о предшествующих блоках. Реестр автоматически обновляется на всех устройствах системы, после чего начинается генерация следующего блока. Шифрование основывается на определенных криптоалгоритмах [2, 3].

Основными причинами популярности технологии блокчейн в настоящее время являются проблемы обеспечения безопасности цифровой инфраструктуры и в частности распределенных реестров, а также проблемы доверия к проведению денежных транзакций.

Распространены блокчейны трех видов: блокчейн без необходимости в разрешениях, открытый блокчейн с разным уровнем разрешений, закрытый блокчейн с разным уровнем разрешений [4].

Под блокчейном без необходимости в разрешениях понимаются блокчейны без управляющего органа, подтверждающего транзакции. При отправке транзакции происходит публичное анонсирование и об этом узнает вся сеть. После этого начинается процесс подтверждения транзакции, причем не известно, кто именно подтвердит транзакцию. Это пример по-настоящему демократичной системы, в которой поль-

зователи обладают некоторой степенью анонимности. Данный вид блокчейна хорошо подходит для защиты особо важной информации.

В открытом блокчейне с разным уровнем разрешений транзакции подтверждают определенные люди. Просматривать информацию может любой пользователь.

Закрытый блокчейн с разным уровнем разрешений похож на открытый, с единственным отличием – данные в нем не открыты для всех. Данный вид хорошо подходит в ситуациях, когда узлам нужно хранить в блокчейне приватную информацию.

Для исследования был использован блокчейн Echo, который в качестве основы использует Graphene и встроенную виртуальную машину Ethereum. Echo – это блокчейн-платформа, которая может использоваться в качестве концептуальной основы и среды для широкого спектра приложений и сервисов на основе смарт контрактов. Особенностями являются: новая модель организации и функционирования сети, инновационная виртуальная машина, поддержка интеграции с другими блокчейнами посредством технологии SideChain, усовершенствованная система управления аккаунтом, поддержка смарт контрактов с использованием не только языка solidity, но и C, и C++.

В данный момент в разработке находится сервер для предоставления удобного интерфейса к данным блокчейна Echo (рабочее название «EchoDB»). При разработке сервера используются программная платформа Node.js, язык программирования TypeScript и документо-ориентированная СУБД MongoDB. Сервер состоит из двух модулей - парсер блокчейна Echo, который заносит информацию из блокчейна в базу данных, и публичного API (GraphQL + WS) для получения этой информации сторонними приложениями.

Парсинг происходит следующим образом – в момент первого запуска парсера из блокчейна получается и создается в базе данных документ, содержащий информацию о так называемом core ассете. Ассет – форма валюты в сети Echo, которая может быть создана и выпущена пользователями. После этого начинается парсинг блоков – по порядку, начиная с первого, из блокчейна запрашивается блок, при помощи библиотеки echojs-lib, затем из него получают транзакции, которые в свою очередь содержат различные операции. На данный момент существует около 50 операций с блокчейном, например, создание нового аккаунта, создание нового ассета, перевод ассетов на другой аккаунт и другие. В зависимости от типа операции и полученной транзакции в базу данных сохраняются различные документы, а также срабатывают различные события, на которые пользователь может подписываться, чтобы получить новую информацию. После того

как все операции в блоке будут обработаны парсером начинается парсинг следующего блока.

Одновременно с парсером работает модуль публичного API. Он использует технологию GraphQL – язык запросов, основанный на графах. При помощи которого можно гибко составлять запросы к базе данных и указывать какую конкретно информацию сервер должен вернуть в ответе. Кроме того, на сервере используется библиотека `apollo-server-express`, поэтому существует возможность подписаться на какие-либо события и если в процессе парсинга будет вызвано данное событие, то сервер вернет информацию о событии. Также эта библиотека предоставляет стандартный пользовательский интерфейс, позволяющий ознакомиться с существующими методами и опробовать API. На рисунке 1 приведен вид стандартного пользовательского интерфейса.

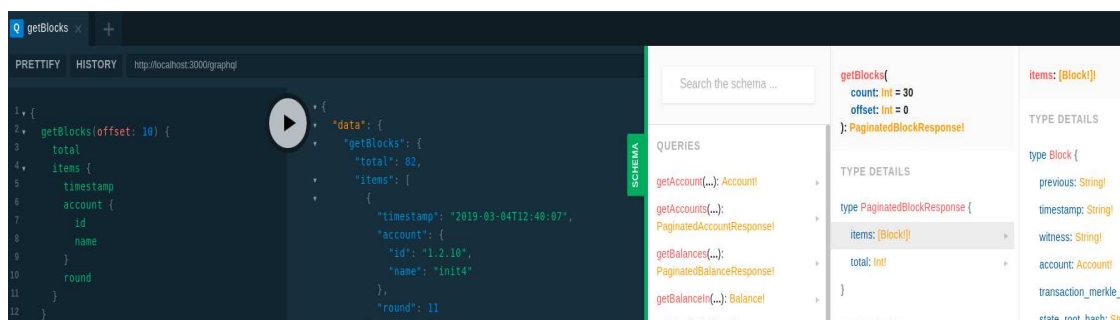


Рисунок 1 – Стандартный пользовательский интерфейс

ЛИТЕРАТУРА

1. Ethclassic – Все о BTC: Новости, события, анализ. Чем различаются блокчейн и распределенный реестр. [Электронный ресурс]. – Режим доступа: <https://ethclassic.ru/2018/04/11/chem-otlichayutsya-blokchejn-i-raspredeleenny-reestr/>. – Дата доступа: 12.04.2019.

2. Урбанович, П. П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие для студ./ Урбанович П.П. – Минск: БГТУ, 2016. – 220 с.

3. Урбанович, П. П. Информационная безопасность и надежность систем: учебно-методическое пособие по одноименному курсу для студентов специальности 1-40 01 02-03 "Информационные системы и технологии" / П. П. Урбанович, Д. М. Романенко, Е. В. Романцевич. – Минск: БГТУ, 2007. – 87 с.

4. BitNovosti – Все о мире Bitcoin: новости, события, факты, курс, анализ. Три распространенных вида блокчейна. [Электронный ресурс]. – Режим доступа: <https://bitnovosti.com/2018/04/25/3-popular-types-of-blockchains/>. – Дата доступа: 12.04.2019.

Студ. Е.А. Шпаковский
Науч. рук. проф. П.П. Урбанович
(кафедра информационных систем и технологий, БГТУ)

СЕТЕВАЯ БЕЗОПАСНОСТЬ: СНИФФИНГ И СПУФИНГ

В настоящее время люди все чаще сталкиваются с терминами «Информационной безопасности и защиты данных в компьютерных сетях» [1,2]. В мире происходит много хакерских атак, давайте рассмотрим такие, как сниффинг и спуфинг [3,4].

Сниффер может анализировать только то, что проходит через «его» сетевую карту. Внутри одного сегмента сети Ethernet все пакеты рассылаются всем машинам, из-за этого возможно перехватывать чужую информацию. Использование коммутаторов (switch, switch-hub) и их грамотная конфигурация уже является защитой от прослушивания. Между сегментами информация передаётся через коммутаторы. Коммутация пакетов — форма передачи, при которой данные, разбитые на отдельные пакеты, могут пересылаться из исходного пункта в пункт назначения разными маршрутами. Так что если кто-то в другом сегменте посылает внутри него какие-либо пакеты, то в ваш сегмент коммутатор эти данные не отправит.

Перехват трафика может осуществляться:

- обычным «прослушиванием» сетевого интерфейса (метод эффективен при использовании в сегменте концентраторов (хабов) вместо коммутаторов (свитчей), в противном случае метод малоэффективен, поскольку на сниффер попадают лишь отдельные фреймы);
- подключением сниффера в разрыв канала;
- ответвлением (программным или аппаратным) трафика и направлением его копии на сниффер (Network tap);
- через анализ побочных электромагнитных излучений и восстановление, таким образом, прослушиваемого трафика;
- через атаку на канальном (MAC-spoofing) или сетевом уровне (IP-spoofing), приводящую к перенаправлению трафика жертвы или всего трафика сегмента на сниффер с последующим возвращением трафика в надлежащий адрес.

Снифферы не всегда применяются в плохих целях, они также могут использоваться в благих целях. Анализ прошедшего через сниффер трафика позволяет:

- обнаружить паразитный и закольцованный трафик, наличие которого увеличивает загрузку сетевого оборудования и каналов связи (снифферы здесь малоэффективны; как правило, для этих целей ис-

пользуют сбор разнообразной статистики серверами и активным сетевым оборудованием и её последующий анализ);

- выявить в сети вредоносное и несанкционированное ПО, например, сетевые сканеры, флудеры, троянские программы, клиенты пиринговых сетей и другие (это обычно делают при помощи специализированных снифферов — мониторов сетевой активности);

- перехватить любой незашифрованный (а порой и зашифрованный) пользовательский трафик с целью получения паролей и другой информации;

- локализовать неисправность сети или ошибку конфигурации сетевых агентов (для этой цели снифферы часто применяются системными администраторами).

Спуфинг (англ. spoofing — подмена) — вид сетевой атаки, при которой злоумышленник стремится выдать себя за другое лицо. Мошенник стремится обмануть сеть или конкретного пользователя, чтобы убедить его в надежности информации.

ARP-спуфинг осуществляет перехват трафика за счет уязвимости арп-протокола из-за отсутствия проверок подлинности запросов и ответов. Протоколы пропускают исходящий трафик на сервер злоумышленника, в результате хакер получает секретные сведения логин, пароль и т. д.

Рассмотрим пример, приведенный на рисунке 1.

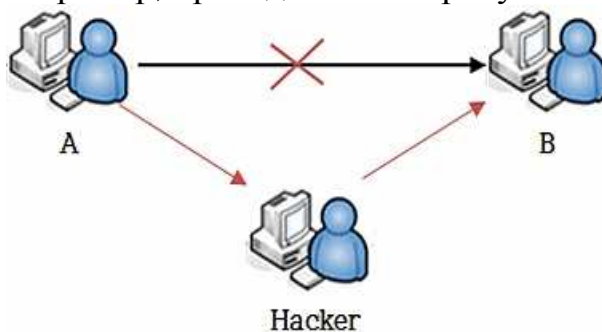


Рисунок 1 – Принцип реализации ARP-атаки

Связь между А и В до применения ARP-spoofing производилась напрямую между двумя компьютерами. В случае, когда произошла ARP-spoofing атака на связь между А и В, все пакеты проходят сперва через Н-хакера.

1. Два компьютера (узла) А и В в локальной сети Ethernet обмениваются сообщениями. Злоумышленник Н, находящийся в этой же сети, хочет перехватывать сообщения между этими узлами. До применения атаки ARP-spoofing на сетевом интерфейсе узла А ARP-таблица содержит IP- и MAC-адреса [2] узла В. Также на сетевом интерфейсе узла В ARP-таблица содержит IP- и MAC-адреса узла А.

2. Во время атаки ARP-spoofing узел Н (злоумышленник) отправляет два ARP-ответа (без запроса) — узлу А и узлу В. ARP-ответ узлу А содержит IP-адрес В и MAC-адрес Н. ARP-ответ узлу В содержит IP-адрес А и MAC-адрес В.

3. Так как компьютеры А и В поддерживают самопроизвольный ARP, то, после получения ARP-ответа, они изменяют свои ARP-таблицы, и теперь ARP-таблица А содержит MAC адрес Н, привязанный к IP-адресу В, а ARP-таблица В содержит MAC адрес Н, привязанный к IP-адресу А.

4. Тем самым атака ARP-spoofing выполнена, и теперь все пакеты (трафик) между А и В проходят через Н. К примеру, если А хочет передать пакет компьютеру В, то А «смотрит» в свою ARP-таблицу, находит запись с IP-адресом узла В, выбирает оттуда MAC-адрес (а там уже MAC-адрес узла Н) и передает пакет. Пакет поступает на интерфейс Н, анализируется им, после чего перенаправляется узлу В.

Чтобы предотвратить ARP-атаку можно использовать несколько методов, каждый из которых имеет свои плюсы и минусы. К ним относятся статические ARP записи, шифрование, VPN и анализ пакетов.

Статические ARP записи – это решение связано с большими административными затратами и рекомендуется только для небольших сетей. Он включает добавление ARP записи для каждого компьютера в сети на каждый отдельный компьютер.

Сопоставление компьютеров с наборами статических IP-адресов и MAC-адресов помогает предотвратить спуфинговые атаки (spoofing attacks), поскольку компьютеры могут игнорировать ответы ARP. К сожалению, это решение может защитить вас только от простых атак.

Шифрование[1, 5]: протоколы, такие как HTTPS и SSH, также могут помочь уменьшить вероятность успешной атаки отравления ARP. Когда трафик зашифрован, злоумышленнику придется предпринять дополнительный шаг, чтобы обмануть браузер цели и заставить его принять незаконный сертификат. Однако любые данные, передаваемые за пределы этих протоколов, будут по-прежнему уязвимы.

VPN может быть разумной защитой для частных лиц, но данный вариант обычно не подходит для крупных организаций. Если только один человек устанавливает потенциально опасное соединение, например, использует публичный Wi-Fi в аэропорту, то VPN зашифрует все данные, которые передаются между клиентом и сервером выхода. Это помогает обеспечить их безопасность, потому что злоумышленник сможет увидеть только зашифрованный текст.

Тем не менее, это менее осуществимое решение на организационном уровне, поскольку между каждым компьютером и каждым сервером должны быть установлены VPN-соединения. Это будет не только сложно настроить и поддерживать, но шифрование и дешифрование в таких масштабах также будет влиять на производительность сети.

Фильтры пакетов. Эти фильтры анализируют каждый пакет, отправляемый по сети. Они могут отфильтровывать и блокировать вредоносные пакеты, а также те, чьи IP-адреса являются подозрительными. Фильтры пакетов также могут сообщать о том, поступает ли пакет из внутренней сети, когда он фактически исходит извне, что, в свою очередь помогает снизить вероятность успеха атаки.

Если вы хотите, чтобы ваша сеть была защищена от угрозы «отравления» ARP, лучший вариант для вас – это комбинация вышеупомянутых инструментов предотвращения и обнаружения. Методы предотвращения, как правило, имеют недостатки в определенных ситуациях, поэтому даже самая безопасная среда может оказаться под угрозой.

ЛИТЕРАТУРА

1. Урбанович, П.П. Защита информации методами криптографии, стеганографии и обфускации/ П.П. Урбанович. – Минск : БГТУ, 2016, – 220 с.
2. Урбанович, П. П. Компьютерные сети : учебное пособие для студентов высших учебных заведений по техническим специальностям / П. П. Урбанович, Д. М. Романенко, Е. В. Кабак. - Минск: БГТУ, 2011. - 399с.
3. IT-безопасность. [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/ARP-spoofing>. – Дата доступа: 17.04.2019.
4. Сниферы. [Электронный ресурс]. – Режим доступа: <https://www.avast.ru/c-sniffer>. – Дата доступа: 17.04.2019.
5. Урбанович, П. П. Информационная безопасность и надежность систем: учебно-методическое пособие по одноименному курсу для студентов специальности 1-40 01 02-03 "Информационные системы и технологии" / П. П. Урбанович, Д. М. Романенко, Е. В. Романцевич. – Минск: БГТУ, 2007. – 87 с.

КОЛЛИЗИИ И НЕОБРАТИМОСТЬ ХЕШ-ПРЕОБРАЗОВАНИЙ В АЛГОРИТМЕ SHA-256

Хэш-функция – функция, преобразовывающая входную последовательность данных произвольного размера в выходную последовательность фиксированного размера. Процесс преобразования данных называется хеширование [1-4].

Использование хеш-преобразований:

- проверка целостности данных (обнаружение изменений),
- системы аутентификации,
- создание и проверка ЭЦП.

Основное свойство всех хеш-функций – их необратимость, то есть, по хэшу невозможно восстановить первоначальные данные, по которым он вычислен. Это свойство позволяет применять хеширование в таких важных областях человеческой деятельности, как компьютерная безопасность и криптография. При этом хеш-функции, применяемые в криптографии, должны отвечать требованиям криптографической стойкости: должно быть практически невозможно подобрать для заданных данных другую последовательность данных с совпадающей хеш-функцией, и, кроме того, должно быть практически невозможно подобрать наугад две последовательности данных с совпадающим хэшем. Слово "практически" подразумевает разумную сложность подбора, определяемую математически.

Виды хеш-функций

1. Хеш-функции, основанные на делении. Пусть x – хешируемое сообщение, а N – максимально возможное число хеш-кодов. Тогда метод хеширования посредством деления будет заключаться в вычислении остатка от деления k на N : $h(x) = x \bmod N$.

2. Хеш-функции, основанные на умножении. Получить из исходной последовательности последовательность хеш-кодов, используя метод умножения (мультипликативный метод), значит воспользоваться хеш-функцией: $h(x) = \lfloor N * (x * A) \rfloor$. Здесь A – рациональное число, по модулю меньшее единицы ($0 < A < 1$).

Также правая часть функции содержит три пары скобок, означающих следующее:

- () – скобки приоритета;
- [] – скобки вычисления целой части;
- { } – скобки определения дробной части.

Хеш-функции семейства SHA-2 построены на основе структуры Меркла–Дамгора: – метода построения криптографических хеш-

функций, предусматривающего разбиение входных сообщений произвольной длины на блоки фиксированной длины и работающего с ними по очереди с помощью функции сжатия, каждый раз принимающего входной блок с выходным от предыдущего прохода. Популярность структуры Меркла–Дамгора обусловлена следующим результатом: если односторонняя функция сжатия $\{ \displaystyle f \}$ устойчива к коллизиям, то и хеш-функция, построенная на её основе, будет также устойчива к коллизиям.

Схематично алгоритм *Sha-256* можно представить следующим образом (см. рис.1) [2].

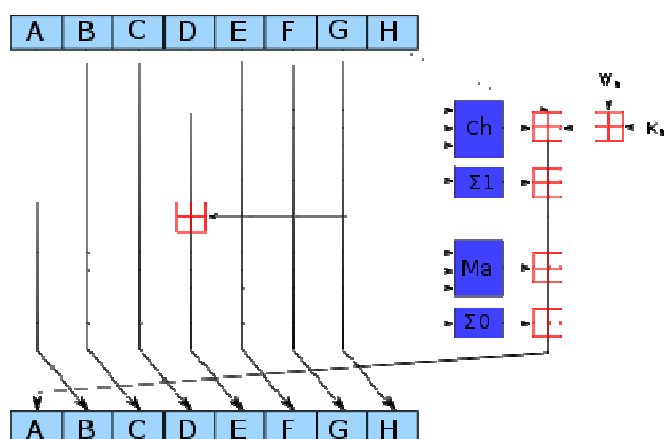


Рисунок 1 – Схема одной итерации алгоритмов SHA-256

Исходное сообщение после дополнения разбивается на блоки, каждый блок – на 16 слов. Алгоритм пропускает каждый блок сообщения через цикл с 64 или 80 итерациями (раундами). На каждой итерации 2 слова преобразуются, функцию преобразования задают остальные слова. Результаты обработки каждого блока складываются, сумма является значением хеш-функции. Тем не менее, инициализация внутреннего состояния производится результатом обработки предыдущего блока. Поэтому независимо обрабатывать блоки и складывать результаты нельзя.

Криптоанализ хеш-функции подразумевает исследование устойчивости алгоритма по отношению, по меньшей мере, к следующим видам атак:

- нахождение коллизий, то есть разных сообщений с одинаковым хешем,
- нахождение прообраза, то есть неизвестного сообщения по его хешу.

От устойчивости хеш-функции к коллизиям зависит безопасность электронной цифровой подписи с использованием данного хеш-алгоритма. От устойчивости к нахождению прообраза зависит, например, безопасность хранения хешей паролей для целей аутентифи-

кации, генерации ключей: клиент генерирует открытый ключ pk и секретный ключ sk для шифрования открытого текста. В большинстве случаев речь идет о безопасности сетевых технологий [1, 6].

Одной из причин необратимости является операция логического сдвига. Потеря некоторой части информации, совершить которую легко, а восстановить, что там было, можно только перебором. Схематично алгоритм сдвига можно представить в виде рис.2 [5].

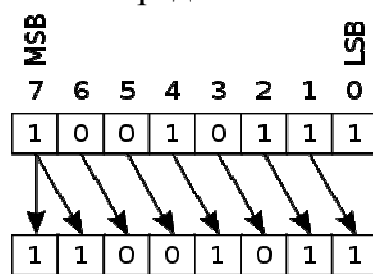


Рисунок 2 – Схема сдвига

Существуют и иные методы хеширования [7-8].

ЛИТЕРАТУРА

1. Урбанович, П. П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие для студ./ Урбанович П.П. – Минск: БГТУ, 2016. – 220 с.
2. Функции хэширования [Электронный ресурс]. – Режим доступа: http://mf.grsu.by/UchProc/livak/b_protect/zok_7.htm. Дата обращения: 10.04.2019.
3. Сингх, С. Книга шифров. Тайная история шифров и их расшифровки/ С. Сингх. – М.: Астрель, 2006. – Р. 150–215.
4. Мао, В. Современная криптография. Теория и практика / В. Мао. – М.: Вильямс, 2005. – Р. 16-80.
5. Алгоритмы / Хэш-функция SHA-256. [Электронный ресурс]. – Режим доступа: <https://medium.com/dtechlog/алгоритмы-хэш-фнкция-sha-256-9862302f942f> –Дата обращения: 09.04.2019.
6. Урбанович, П. П. Компьютерные сети: учебное пособие для студентов высших учебных заведений по техническим специальностям / П. П. Урбанович, Д. М. Романенко, Е. В. Кабак. – Минск: БГТУ, 2011. – 399 с.
7. Urbanovich, P. The appearance of conflict when using the chaos function for calculating the hash code / P. Urbanovich, M. Plonkowski , K. Churikov // Przegląd elektrotechniczny. – 2012. – R. 88, № 11b. – P. 346-347.
8. Urbanovich, P. The appearance of conflict by using the chaos function to calculate the hash code /P. Urbanovich, M. Plonkowski, K. Churikau // 7th International Conference NEET'2011, Zakopane, Poland, June 28–July 1, 2011. – P. 151.

УДК 004.415.2:336

Студ. Д.С.Ющик

Науч. рук. ст. преп. Е.А. Блинова

(кафедра информационных систем и технологий, БГТУ)

РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ ПОВЫШЕНИЯ ФИНАНСОВОЙ ГРАМОТНОСТИ ШКОЛЬНИКОВ

Финансовая грамотность и проблемы ее повышения среди населения в нашей стране находятся в зоне особого внимания. Исследования показывают, что в целом для Беларуси характерен относительно невысокий уровень финансовой грамотности. Типичный портрет белорусского пользователя финансовых услуг таков: «он плохо ориентируется в финансовых продуктах, пользуется только несколькими основными услугами и слабо информирован о механизмах защиты своих прав в финансовой сфере».

Перед началом разработки был проведен аналитический обзор существующих на рынке программных продуктов. На основе данных, полученных при качественной оценке аналогов, формируются основные аспекты деятельности, выделяются проблемные направления, требующие особого внимания. Анализ позволяет выявить и сильные позиции, что в свою очередь может повлиять на уже существующую идею в пользу более грамотного решения.

Проведя аналитический обзор, выяснилось, что аналогов данному программному продукту в Республике Беларусь нет. Схожие программные продукты есть в Российской Федерации.

В Республике Беларусь есть курсы по финансовой грамотности при университетах. Курсы очные, что в реалиях современного мира не всегда удобно. Участие в очных курсах неудобно для школьников из небольших городов и деревень.

В Республике Беларусь существует «Единый интернет-портал финансовой грамотности населения» [1]. Данный портал содержит актуальную официальную информацию, однако, такая информация трудная для понимания школьниками.

В Российской Федерации есть большой выбор онлайн-курсов по финансовой грамотности. Курсы рассчитаны на разный возраст и уровень финансовой грамотности. Для аналитического обзора были рассмотрены курсы именно для школьников.

Одним из аналогов является онлайн-курс «Основы финансов для молодежи» [2]. Проект разработан Центром дополнительного образования Новосибирского Государственного Университета. Курс бесплатный и не ограничен во времени, после прохождения курсов

пользователь получает сертификат. Курс состоит из более чем 100 мини-заданий, разбитых по уровням. В каждом задании есть видео-лекция или статья, а также задачи по теме задания.

Курс размещен на собственном сайте. Дизайн сайта интуитивно понятен для пользователя с любым уровнем владения компьютером. Цветовая схема не отвлекает от информации. Однако меню, расположенное с левой стороны, нельзя скрыть, из-за чего съедается часть экрана и отвлекает пользователя.

Следующим анализируемым курсом является «Основы финансовой грамотности» от Московской академии предпринимательства при Правительстве Москвы [3]. Курс не имеет собственного сайта, а располагается на платформе Zillion.net.

Курс бесплатный и рассчитан на учащихся старших классов. Курс включает в себя 13 глав. В каждой главе есть видео-лекция, лекция в текстовом формате, а также тест по данной главе. По результатам изучения дисциплины предусмотрен итоговый контроль в форме тестирования. По окончании курса пользователю выдается сертификат.

У пользователя так же есть возможность оставлять заметки при просмотре материалов и прохождения тестов. А так же можно скачать презентацию по теме раздела.

Дизайн платформы интуитивно понятен. На сайте присутствуют яркие цвета, но они используются только для привлечения внимания к главным деталям и не отвлекают пользователя.

Информация на странице располагается неудобно. Текст лекции занимает 1/3 экрана, что является неудобным для изучения. Справа находится блок с информацией об активности пользователя, который нельзя скрыть и который отвлекает. Заголовок с названием темы лекции занимает целый блок, который так же нельзя скрыть.

По результатам аналитического сбора можно сделать вывод, что данная тема актуальна, т.к. не имеет аналогов на территории Республики Беларусь.

Проанализировав аналоги на территории Российской Федерации, были выявлены сильные и слабые стороны. Так же можно заметить, что ни у одной из рассмотренных систем нет мобильного приложения, что в современных реалиях значительно снижает популярность продукта.

Для разработки системы была выбрана программная платформа .Net, для БД был выбран MicrosoftSQLServer 2017, для мобильного приложения была выбрана ОС Android и язык программирования JavaAndroid. Схема БД представлена на рисунке 1.

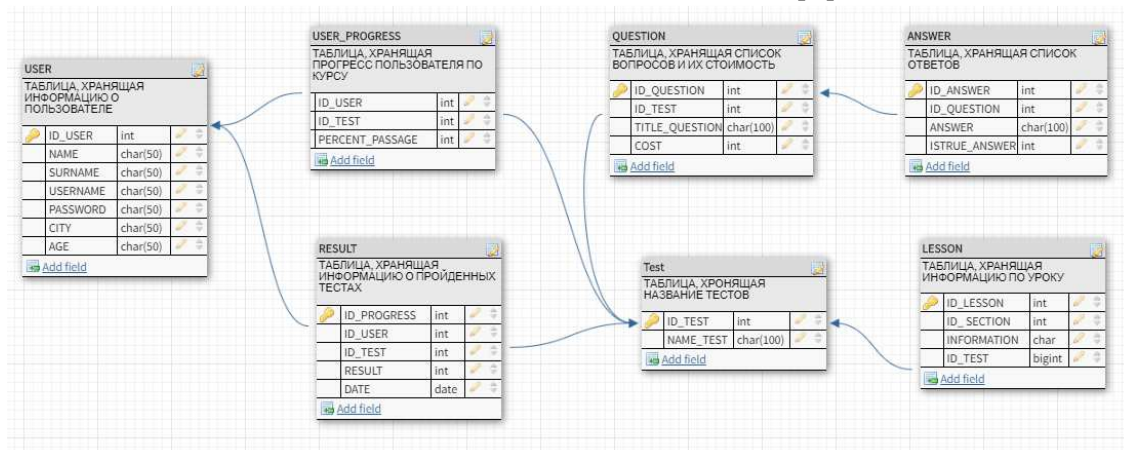


Рисунок 1 – Схема базы данных

Так же, на основании использования приложений-аналогов были выработаны основные критерии для построения UX дизайна приложения, позволяющего осуществлять наиболее простую и интуитивную навигацию по приложению.

Курс состоит из трех модулей, в каждом по несколько уроков. Каждый следующий модуль является логическим продолжением предыдущего и развивает тему от простого к сложному.

Модуль 1 «Личные финансы. Планирование» содержит уроки, изучив которые слушатели смогут научиться основам личного финансового планирования, узнают основные правила финансового поведения, поймут, с чего начать свою правильную финансовую жизнь.

Модуль 2 «Управление финансами» посвящен изучению инструментов для сохранения и приумножения денежных средств, правилам общения в банковской сфере.

Модуль 3 «Развитие финансов». На уроках этого модуля рассматриваются инструменты для развития и приумножения финансов: инвестиции и страхование. Серьезные вопросы рассматриваются на доступном уровне.

Все уроки курса снабжены видеолекциями, сопровождаются текстовым материалом.

Каждый урок – это набор шагов, позволяющих поэтапно изучить и закрепить материал. Разработаны комплексы практических заданий для каждой темы, включающие тесты. Материал курса подобран так, чтобы он был доступен и учащимся 7-9 классов, но подойдет и более старшим слушателям. В конце курса слушатели проходят итоговый тест. При успешном прохождении курса выдается сертификат.

ЛИТЕРАТУРА

1. <http://fingramota.by/ru>[Электронный ресурс]. 2018. – Режим доступа: <http://fingramota.by/ru>. – Дата обращения: 10.02.2019.

2. <http://osnovi-finansov.ru/course/about/>[Электронный ресурс]. 2018. – Режим доступа: <http://osnovi-finansov.ru/course/unit.>– Дата обращения: 20.02.2019.

3. <https://zillion.net/courses/show/6926/osnovy-finansovoi-ghramotnosti/>[Электронный ресурс]. 2018. – Режим доступа: <https://zillion.net/ru/>. – Дата обращения: 21.02.2019.

УДК 557.114:616-006

Студ. Н.Д. Куницкий

Науч. рук. ассист. К.И. Павловская
(кафедра информационных систем и технологий, БГТУ)

РАСПРЕДЕЛЕННАЯ СИСТЕМА МЕНЕДЖМЕНТА ДОПОЛНИТЕЛЬНЫХ ЗАНЯТИЙ

Переход современного общества от индустриального к информационному обуславливает возрастание роли и социальной значимости образования. Для человека становится все более необходимым приобретение знаний, умений, навыков, чтобы обеспечить устойчивость собственного положения в обществе. Это касается не только его сегодняшней практической деятельности, но и будущего. Образование является основным источником повышения интеллектуального, культурного, профессионального и кадрового потенциала.

Цель работы: провести поиск аналогов, написать распределенную систему, позволяющую выполнять менеджмент дополнительных учебных занятий, сделать мобильную версию.

Распределённое приложение – это программа, состоящая из нескольких взаимодействующих частей, каждая из которых, как правило, выполняется на отдельном компьютере (или другом устройстве) сети, на рисунке 1 представлена схема.



Рисунок 1 – Схематичное представление распределенной системы.

К основным задачам распределенной системы относятся:

1. Безопасность - большая проблема в распределенной среде, особенно при использовании общедоступных сетей.
2. Отказоустойчивость - может быть жесткой, когда модель построена на основе ненадежных компонентов.

3. Координация и распределение ресурсов - могут быть трудными, если нет надлежащих протоколов или требуемой политики.

Один из главных вопросов — как точно разделить клиента и сервера. Анализ приложений типа клиент-сервер, предназначенных для организации доступа к базам данных, привел к их логическому разбиению на три уровня:

- уровень пользовательского интерфейса;
- уровень обработки;
- уровень данных.

Суть распределенных систем — связь между процессами, реализующими не только взаимодействие компьютеров, но и частей (уровней) приложений. Взаимодействие частей приложений реализуется с помощью протоколов, описывающих состав и формат данных, пересылаемых соответствующими частями клиентских и серверных приложений друг другу для решения поставленной задачи [1].

Создание приложений учебного и образовательного назначения очень трудоемкий процесс, а установление их в сетях влечет дополнительные проблемы, связанные с условиями их распространения. В настоящее время формируются основы методик разработки, распространения и пользования такими приложениями. Однако это лишь небольшая часть от множества других проблем, сопутствующих разработке и распространению образовательных приложений. В мировом сообществе обсуждению подходов к решению этих проблем уделяется большое внимание.

Одним из подходов к решению создания и распространения образовательных приложений в телекоммуникационных сетях Беларуси является адаптация уже существующего в системе общего и профессионального образования программного продукта учебного назначения к особенностям телематических систем. Адаптация ПО заключается в первую очередь в том, что оно должно быть открытым для модификации и сопровождения, восприниматься и поддерживаться соответствующими браузерами. Что касается методических аспектов, то ПО должно быть тщательно отредактировано, информационно выверено и поддерживать принцип персонификации и дозирования учебного материала. Наряду с этими вопросами необходимо решить проблемы авторского права, права на пользование приложением образовательными учреждениями, а также реализовать различные способы доступа и распространения его в сетях.

В рамках исследования были сформулированы следующие задачи:

- разработка распределённого программного средства для управления дополнительными занятиями;
- реализация хранения данных на удалённой базе данных MS SQL Server;
- реализация клиентской части, в виде web-приложения;
- реализация клиентской части в виде мобильного приложения;
- реализация серверной части приложения.

Результатом работы является готовый программный продукт. Данная система состоит из:

- сервера – принимает запросы клиентских приложений, обрабатывает их и отправляет ответы;
- web-сайта – предоставляет функционал для администратора, преподавателя, и студентов. Администратор может добавить пользователя как преподавателя, добавить или удалить пользователя;
- мобильного android-приложение выполняет следующие функции: авторизация пользователя как студента, как преподавателя. Преподаватель имеет права на добавление дополнительного занятия, определяет количество человек, которое может прийти, аудиторию, дату, время, предмет. Студент может записаться на данное занятие или отписаться от него.

Серверная часть приложения, реализована с помощью технологии ASP.NETCoreMVC [2]. Клиентское, мобильное приложение, было реализовано на языке программирования Java.

Для хранения данных на удаленном сервере, была выбрана свободная объектно-реляционная система управления базами данных – MSSQLServer [3]. Для хранения данных на мобильном устройстве использовалась локальная база данных – SQLite [4].

ЛИТЕРАТУРА

1. Распределенные системы и приложения [Электронный ресурс]. – 2019. – Режим доступа: <https://helpiks.org/9-49793.html>. – Дата доступа: 02.02.2019.
2. ASP.NET Core MVC [Электронный ресурс]. – 2019. – Режим доступа: <https://metanit.com/sharp/aspnet5/3.1.php>. – Дата доступа: 02.02.2019.
3. MySQL [Электронный ресурс]. – 2019. – Режим доступа: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2016>– Дата доступа: 02.02.2019.
4. SQLite [Электронный ресурс]. – 2019. – Режим доступа: <https://www.sqlite.org/index.html>. – Дата доступа: 02.02.2019.

УДК 557.114:616-006

Студ. А.А. Ивлиева
 Науч. рук. ассист. К. И. Павловская
 (кафедра информационных систем и технологи БГТУ)

РАСПРЕДЕЛЕННОЕ ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ ПОКУПКОЙ И ДОСТАВКОЙ ПОДАРКОВ

В условиях современного мира высокий темп роста количества пользователей сети Internet обусловлен возрастающим уровнем информатизации общества. На сегодняшний день миллионы людей ежедневно, не выходя из дома, совершают покупки в интернет-магазинах.

Сегодня сфера интернет-торговли – один из самых популярных способов продажи товаров и услуг. Люди все чаще совершают покупки, сидя за компьютером или используя свой телефон.

Преимуществами интернет-магазина являются:

- Экономия времени. Благодаря навигации и поисковым системам можно быстро найти нужный товар;
- Доставка товаров на дом. Пользователям Интернет-магазинов больше нет необходимости магазин лично, в поисках необходимого товара, а также способа доставки приобретенного товара. Все это быстро и легко делают сотрудники службы доставки;
- Возможность избежать очередей и суеты традиционных магазинов – проблема, которая вызывала больше всего возмущения у предыдущих поколений.

Общее количество интернет-пользователей в возрасте 15-74 лет, которые хотя бы раз в месяц посещали сайты из категорий интернет-магазины, маркетплейсы, прайс-агрегаторы, каталоги товаров/услуг и т.п. составляет 2 069 793 человек (с учетом пересечения аудитории), или 40,7% всех интернет-пользователей Беларуси [1] (рисунок 1).

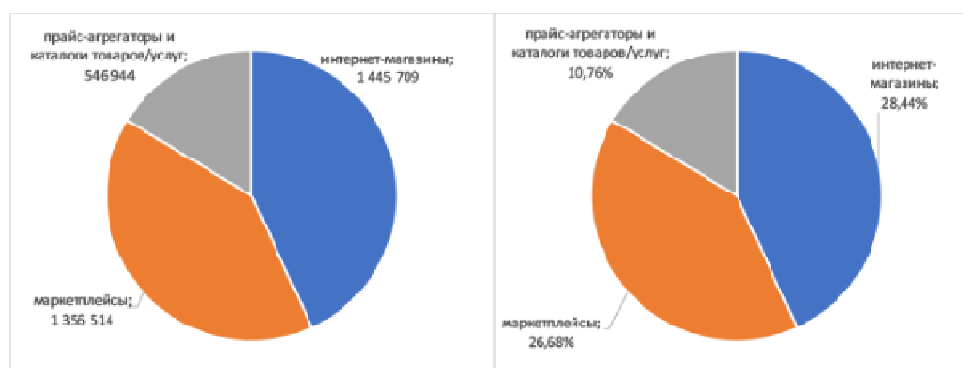


Рисунок 1 – Количество и доли интернет-пользователей сайтов основных сегментов e-commerce в Беларуси

Цель моей работы является разработка распределенного клиент-серверного приложения, компонентами которого являются: веб-приложение, размещенное на сервере, которое будет получать запро-

сы со двух других компонентов разработанной мной системы – мобильные android приложения. Взаимодействие между приложениями осуществляется по протоколу http, модель взаимодействия компонентов системы представлена на рисунке 2.

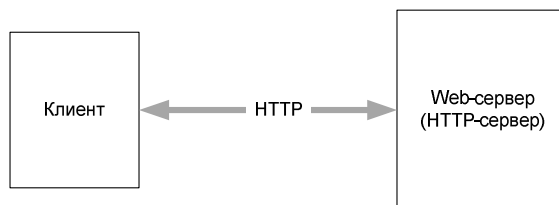


Рисунок 2 – Модель взаимодействия компонентов в системе

Серверная часть выполнена в виде веб-приложения, реализованного с использованием инфраструктуры ASP.NETCoreMVC от компании Microsoft [2]. Мобильные клиенты представлены отдельными приложениями для операционной системы Android на языке программирования Java.

Первое из них предоставляет интерфейс для работы с каталогом товаров, а именно, подарков, а также взаимодействие с пользовательской корзиной. Второе мобильное android приложение необходимо для работы курьера, которое отображает текущие заказы, позволяет изменять статус заказа, а также данное приложение позволяет отслеживать местоположение курьера.

Серверное веб-приложение использует базу данных MySQL [3]. Для хранения данных на мобильном устройстве использовалась локальная база данных – SQLite [4].

В результате проделанной работы, было разработано распределенное клиент-серверное приложения для заказа подарков и получение информации о местоположении курьеров, осуществляющих доставку, заказанного товара, а также решены поставленные цели.

ЛИТЕРАТУРА

1. Исследование: Сколько интернет-пользователей Беларуси пользуется услугами e-commerce площадок? [электронный ресурс] – 2019. – Режим доступа: <http://marketing.by/analitika/issledovanie-skolko-internet-polzovateley-belarusi-polzuetsya-uslugami-e-commerce-ploshchadok/> – Дата обращения: 11.03.2019.
2. ASP.NET Core MVC [Электронный ресурс]. – 2019. – Режим доступа: <https://metanit.com/sharp/aspnet5/3.1.php> – Дата обращения: 11.03.2019.
3. MySQL [Электронный ресурс]. – 2019. – Режим доступа: <https://www.mysql.com/> – Дата обращения: 12.03.2019.
4. SQLite [Электронный ресурс]. – 2019. – Режим доступа: <https://www.sqlite.org/index.html> – Дата обращения: 12.03.2019.

Студ. К.А. Ахраменок
Науч. рук. доц., к.т.н. Н.А. Жилияк
(кафедра информационных технологий, БГТУ)

ПРИМЕНЕНИЕ СКОРИНГА В РАБОТЕ С BIGDATA

BigData — термин, появившийся совсем недавно. Большими данными принято называть огромные массивы информации со сложной неоднородной и\или неопределенной структурой [3]. Отсюда возникает проблема обработки, структурирования и извлечения полезной информации из больших наборов данных.

Цель нашей работы: проанализировать литературы по вопросу применения скоринга в работе с BigData, выявить виды скоринга, которые могут быть взяты из банковского опыта и применены к обработке больших объемов данных.

Сегодня термин BigData, как правило, используется для обозначения не только самих массивов данных, но также инструментов для их обработки. Главные характеристики, отличающие BigData от другого рода данных – три V: volume, velocity, variety [3]. Первая – большие объемы, вторая – необходимость в быстрой обработке и высокая скорость накопления этих данных, третья – разнообразие. Исходя из характеристик BigData, было принято решение проанализировать возможность использования скоринга в обработке больших объемов данных [3].

Скоринг (от англ. Scoring – подсчет очков в игре) – это модель классификации клиентской базы на различные группы, если неизвестна характеристика, разделяющая эти группы, но известны другие факторы, связанные с интересующей нас характеристикой [1]. В основе скоринговой система лежит предположение, что люди со схожими социальными показателями ведут себя одинаково. Таким образом, мы можем попытаться применить это предположение в работе с большими данными информации.

Посредством различных видов скоринга можно как классифицировать данные так и давать вероятностные прогнозы на происхождение того или иного события [2].

После анализа литературы было выяснено, что из всех существующих видов скоринга, для работы с большими объемами данных подходят только два: Application-scoring и Behavioral-scoring. Их описание применения в банковской сфере и возможность применения в работе с BigData приведены в таблице [1].

Таблица 1

Вид скоринга	Применение в банковской сфере	Возможное применение в работе с BigData
application-scoring	Это самый распространенный и известный клиентам вид скоринга. В его основе лежат первичный сбор анкетных данных заемщика, их обработка компьютером и вывод результата: предоставлять заем или нет [2].	Каждому свойству исследуемого объекта присваивается вес. После этого вес всех свойств подсчитывается и на его основе выставляется оценка. Исходя из этой оценки, отнести объект к какой-либо группе.
behavioral-scoring	«скоринг поведения» - оценка наиболее вероятных финансовых действий заемщика. Такая система дает возможность прогнозировать изменение платежеспособности заемщика, корректировать установленные для него лимиты [3].	Может быть использован для решения задач прогнозирования на основе имеющихся данных за период времени. Данные будут проходить обработку посредством скоринговой модели, а на выходе мы будем получать предполагаемое поведение объекта.

На основе выше сказанного были выделены следующие этапы обработки больших объемов данных посредством скоринга:

1. определить интересующую характеристику;
2. провести сбор второстепенных сведений о объекте исследования;
3. разработать скоринговую модель на основе имеющихся данных;
4. провести обработку данных посредством скоринговой модели

Актуальность работы заключается в увеличении объемов информации, требующей обработки, недостатке качественных методов обработки больших объемов информации.

Научная новизна заключается в том, что до текущего момента не было предложено вариантов использования скоринга в работе с большими объемами данных.

В результате работы была проанализирована литература по теме исследования, были выявлены методы скоринга, которые можно применить в работе с BigData, были выделены этапы обработки больших объемов данных посредством скоринга.

ЛИТЕРАТУРА

1. The credit scoring toolkit: Theory and Practice for retail Credit Risk Management and Decision Automation/ Anderson R. [et al] // England: Oxford University Press. – 2007. – P. 8–13, 16–17.
2. Data Smart: Using Data Science to Transform Information into Insight / John W. Foreman [et al] // Wiley, UK. – 2013. – P. 2–5.
3. Bigdata простым языком / А.П. Благирев; Издательство АСТ – Москва, 2019. №1:27–32 с.

Студ. А.В. Тарасенко

Науч. рук. доц., к.т.н., Н.А. Жилияк

(кафедра информационных систем и технологий, БГТУ)

СЕРВИС ДЛЯ ПОДБОРА НОВОСТНЫХ СТАТЕЙ НА ОСНОВЕ ПРЕДПОЧТЕНИЙ ПОЛЬЗОВАТЕЛЯ

Переход общества в цифровую эру обусловил увеличение спроса на потребление информации в электронном виде. Как результат на данный момент необходимы быстрые и эффективные способы обработки большого количества информации и её структурирование для дальнейшего подбора для конечного пользователя. Всё это привело к увеличению спроса на различные алгоритмы анализа текстовой информации. Идет переход от ручного анализа текста к машинному. В связи с этим нужно создать эффективный алгоритм для анализа текста и выделения ключевых слов, для последующего применения данного алгоритма новостными порталами и агрегаторами.

Для решения поставленной задачи необходимо очистить входные данные [1]. Данный алгоритм разработан и представлен на рисунке 1. После очистки данных необходимо выделить в тексте ключевые слова т.к с их помощью, на основе алгоритма TextRankи будет происходить подбор новостного контента [2].

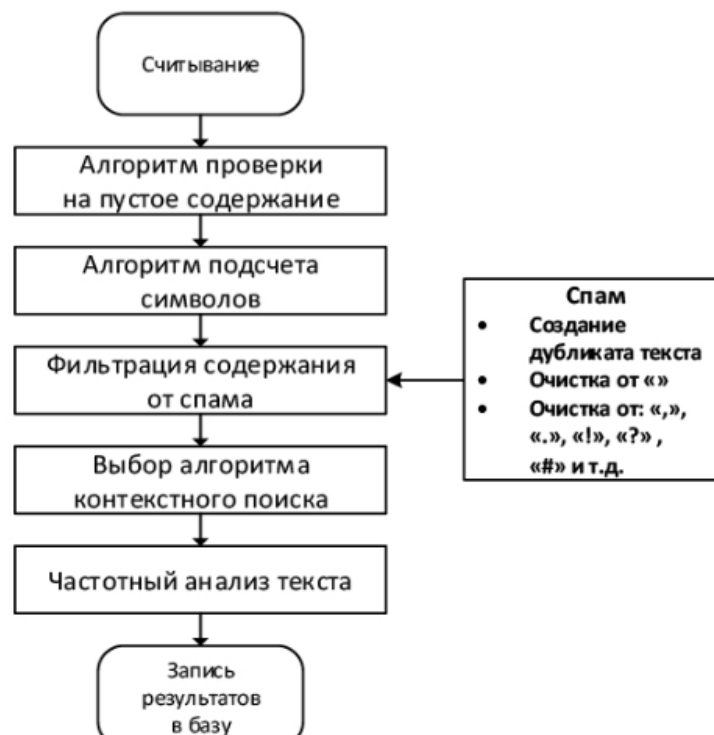


Рисунок 1 – Алгоритм анализа текста

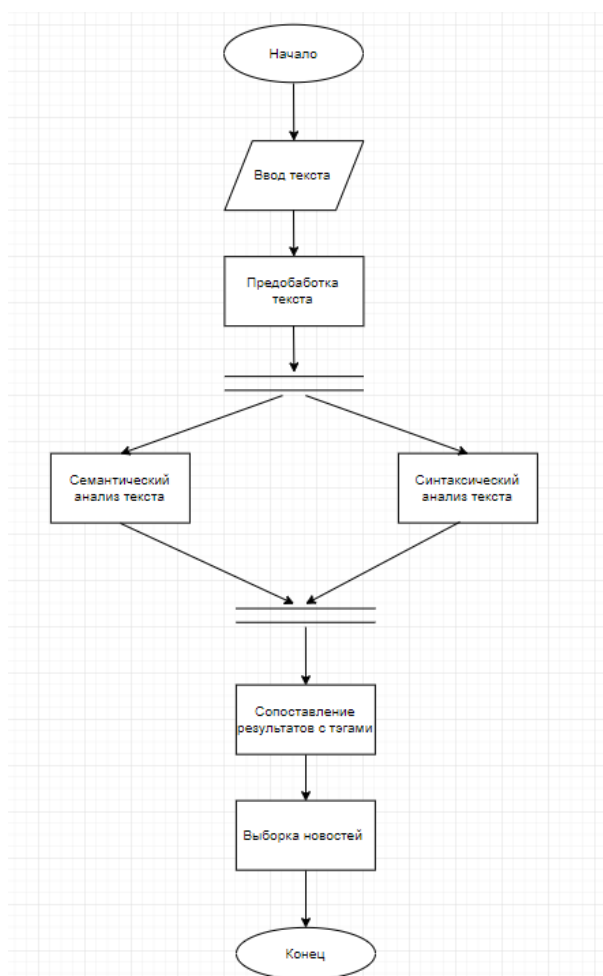


Рисунок 2 – Алгоритм подбора новостного контента

Сервис, созданный на основе алгоритма, разработанного и представленного на рисунке 2, позволяет анализировать входной текст с высокой скоростью и на основе его выдавать схожие статьи. Данный сервис может быть использован новостными порталами и агрегаторами не только для подбора новостей, но и для контекстной рекламы.

ЛИТЕРАТУРА

1. Web Rank/ Shaoping N. [et al] // USA: Google Ltd. – 2018. – P. 36–37.
2. Синтаксический разбор / Буркнцев В.А. // Министерство образования и науки Российской Федерации. Федеральное агентство по образованию Российской Федерации. Еленцкий Государственный университет им И.А.Бунина. – P. 10–15.

Студ. А.Н. Зайцев

Науч. рук. доц. А.А. Дятко

(кафедра информатики и веб-дизайна, БГТУ)

СТАТИЧЕСКИЙ АНАЛИЗ ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ КОНТЕЙНЕРОВ ИЗ БИБЛИОТЕКИ STL В ПРОГРАММАХ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ C++

В современном мире наблюдается тенденция смещения акцента при выборе средств разработки к более высокоуровневым, которые сокращают время, затрачиваемое на разработку программного комплекса. Также развиваются различного вида «помощники» для программистов, которые облегчают различные стороны разработки: мощные интегрированные среды разработки, автоматическое форматирование кода, средства виртуализации и контейнеризации и т.д.

Одним из самых популярных методов автоматической проверки программ является статический анализ. Статический анализ — анализ программного обеспечения, производимый (в отличие от динамического анализа) без реального выполнения исследуемых программ [1]. Данный вид анализа позволяет предупредить множество потенциальных проблем с программой: использование освобождённой памяти, неоптимальное использование языковых средств и множество других проблем. В ходе данной работы мною был реализован статический анализатор, который позволяет определить, оптимален ли контейнер для работы с данными, которые выбирает программист в тех или иных ситуациях в программах на языке программирования C++.

Для реализации данного проекта мною был использован набор LLVM (Low-Level Virtual Machine) — набор библиотек для разработки компиляторов и утилит для работы с программами [2]. Из LLVM для реализации проекта я использую компилятор Clang [3] и статические анализаторы Clang Static Analyzer[4] и Clang Tidy[5].

В данной работе анализируется использование следующих контейнеров и Стандартной Библиотеки Шаблонов (Standard Template Library, STL): `std::vector`, `std::list`, `std::forward_list`, `std::array`, `std::stack`, `std::queue`, `std::deque`.

В связи со сложностью задачи введены следующие ограничения:

- анализ проводится только на уровне функций. Не допускается анализ между функциями и между разными единицами трансляции;
- анализ ограничен только подмножеством STL из-за сложности поддержки большого множества контейнеров;

- не поддерживается автоматическая замена контейнеров в исходном коде программы ввиду ограничений выбранного для данной работы статического анализатора и сложности сохранения корректности при такой замене.

В данном статическом анализаторе используется анализ абстрактного синтаксического дерева (Abstract Syntax Tree, AST). Если говорить более конкретно, то используется сопоставление с паттерном по AST.

В ходе анализа нас интересуют следующие метрики для каждого контейнера:

- количество операций поиска в контейнере;
- количество операций удаления элемента из контейнера;
- количество операций добавления элемента в контейнер;
- известен ли начальный и/или конечный размер контейнера при его создании.

Ниже на рис. 1 представлена схема работы данного статического анализатора.



Рисунок 1 – Схема работы данного статического анализатора

Для сбора метрик для каждого контейнера были написаны специальные поисковые алгоритмы на специальном встраиваемом языке из CSA (embedded Domain Specific Language, eDSL).

Затем на AST каждой функции запускается механизм, который сопоставляет различные части AST с паттернами, которые были описаны выше. Если было обнаружено какое-то совпадение, то AST кода сохраняется в кеш. После того начинается стадия обработки собранных данных. Оно производится с помощью обыкновенного дерева различных вариантов. В конце работы данного алгоритма исходя из полученных метрик выносится решение о целесообразности применения в данном месте того или иного контейнера. Данный анализатор обладает следующими недостатками:

- недостаток информации о паттерне исполнения программы, что может крайне негативно сказываться на качестве проводимого анализатора;
- не используются анализ потока графа управления и метод символического исполнения.

Возможные пути улучшения качества анализа:

- использование динамического анализа (например, PGO [6]) для получения более точных метрик о паттерне поведения программы.

Это в том числе поможет более точно определять «горячие» и «холодные» пути выполнения программы;

- проведение анализа не только для контейнеров из библиотеки STL. Это сделать достаточно сложно, так как нужна информация о том, как контейнер себя ведёт. Этого можно добиться с помощью ручного аннотирования контейнеров и проведения различных бенчмарков с сохранением результата в базе данных статического анализатора.

Похожие исследования в данной области проводили следующие два проекта:

- Cozy [7]. Проект по генерации контейнеров из высокоуровневого описания требований к контейнеру. Открытая реализация;

- Chameleon [8]. Проект для языка программирования Java по выбору оптимального контейнера в зависимости от текущей нагрузки и замена «на лету» контейнера при помощи модификации JVM.

С предварительной реализацией данного статического анализатора можно ознакомиться по следующей ссылке: <https://github.com/ZaMaZaN4iK/llvm-project>

ЛИТЕРАТУРА

1. Wikipedia : Staticprogramanalysis [Электронный ресурс] / https://en.wikipedia.org/wiki/Static_program_analysis

2. TheLLVMCompilerInfrastructure [Электронный ресурс] / <https://llvm.org>

3. Clang: a C language family frontend for LLVM [Электронный ресурс] / <https://clang.llvm.org>

4. ClangStaticAnalyzer [Электронный ресурс] / <https://clang-analyzer.llvm.org/>

5. ClangTidy [Электронный ресурс] / <https://clang.llvm.org/extra/clang-tidy/>

6. Wikipedia : Profile-guided optimization [Электронный ресурс] / https://en.wikipedia.org/wiki/Profile-guided_optimization

7. Cozy, the collection synthesizer [Электронный ресурс] / <https://cozy.uwplse.org/>

8. Research Gate : Chameleon: Adaptive Selection of Collections [Электронный ресурс] / https://www.researchgate.net/publication/220751882_Chameleon_Adaptive_Selection_of_Collections

УДК 004. 934.2

Магистрант А. С. Демещик
Науч. рук. проф. И. Г. Сухорукова
(кафедра программной инженерии, БГТУ)

ПОДХОДЫ К ПРОВЕДЕНИЮ СЕНТИМЕНТ-АНАЛИЗА

Сентимент-анализ – класс методов контент-анализа в компьютерной лингвистике, предназначенный для автоматизированного выявления в текстах эмоционально окрашенной лексики и эмоциональной оценки авторов (мнений) по отношению к объектам, о которых говорится в тексте.

Тональность – это эмоциональное отношение автора высказывания к некоторому объекту (объекту реального мира, событию, процессу или их свойствам/атрибутам), выраженное в тексте. Эмоциональная составляющая, выраженная на уровне лексемы или коммуникативного фрагмента, называется лексической тональностью (или лексическим сентиментом). Тональность всего текста в целом можно определить как функцию (в простейшем случае сумму) лексических тональностей составляющих его единиц (предложений) и правил их сочетания.

Для цельного понимания сентимент-анализа необходимо сформулировать его главную задачу, которая вкратце звучит следующим образом: нахождение в текстовой информации мнений и выявление их свойств. Мнение же, в свою очередь, может быть определено кортежем из 5 элементов (e, f, op, h, t):

- e – entity – объект мнения;
- f – feature – свойств(о/а) объекта;
- op – orientation/polarity – тональная оценка мнения;
- h – holder – субъект(владелец) мнения;
- t – time – момент времени, в который было выражено мнение.

Приведем пример. Ниже представлен комментарий, размещенный на форуме:

Джон Х., 05.02.2019: «Я вчера купил новую фотокамеру и она превосходна!»

Выделим элементы мнения:

- e – entity – фотокамера;
- f – feature – обобщение;
- op – orientation/polarity – положительная;
- h – holder – Джон Х.;
- t – time – 05.02.2019.

Полученный кортеж полностью характеризует ситуацию, описанную в комментарии. Мы знаем объект, относительно которого выражается мнение, его эмоциональную окраску и человека, который выразил мнение.

Само определение сентимент-анализа говорит об областях, в которых он находит применение, а определение мнения дополняет эту картину. Рассмотрим основные области.

Маркетинг. Сентимент-анализ позволяет компаниям получать статистику по продажам и отзывам пользователей о своем продукте. Эта информация может быть использована для индивидуализации продуктов или способов продвижения продуктов, либо для изменения самих продуктов. Например, пользователь в целом положительно отзывается о фотокамере, но отмечает недостаток в надежности. С помощью глубокого сентимент-анализа, компания может выделить отрицательную составляющую отзыва и поработать над надежностью продукта.

Политика. Мнения всегда составляли важнейшую часть политических влияний и систем. Можно сказать, политическая система базируется на основе системы мнений людей абсолютно разных классов и сфер. Сентимент-анализ, главным образом, используется в политике с целью анализа мнений народных масс с целью влияния.

Мониторинг мнений миллионов пользователей Сети. Развитие информационных технологий и, в частности, Интернета, позволяет упрощать процесс контроля и анализа народных мнений. Существует достаточное разнообразие различных инструментов, позволяющих автоматизированно извлекать необходимую информацию с различных порталов, предоставляя ее в удобном для анализа виде.

Развитие Интернета не только дает более простой доступ к мнениям пользователей, но и значительно увеличивает объем информации. Обработать данные вручную – трудоемкий и ресурсозатратный процесс. Поэтому сентимент-анализ, в большинстве своем, проводится автоматизированно, с использованием следующих методов:

- методы, использующие правила либо шаблоны. Один из простейших видов анализа. Он заключается в применении к информации неких шаблонов и анализе полученных результатов. Сложность подобного анализа заключается лишь в составлении правил либо шаблонов, которые будут применяться в дальнейшем. Главным минусом подхода является его негибкость;
- методы машинного обучения (с учителем и без). Один из наиболее популярных видов сентимент-анализа. Эти методы более за-

тратны, чем шаблонные, поскольку требуют этапа обучения модели, однако и наиболее гибкие, ибо не привязаны к определенным языковым паттернам и доменам;

- методы, основанные на теоретико-графовых моделях. Суть этих методов заключается в предположении, что не все слова в документе равнозначно важны. Минусом методов является то, что слова, особенно в русском языке, зависимы от домена и могут иметь кардинально разные значения в разных доменах.

В своей работе я использовал три метода машинного обучения – наивный Байес (Bayes), метод опорных векторов (SVM) и рекуррентную нейронную сеть (RNN). Для RNN функцией активации использовалась ReLU (rectifier) – выпрямитель. Это наиболее простая функция, которая имеет следующий вид:

$$f(x) = \begin{cases} 0.01x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Ниже представлены полученные результаты:

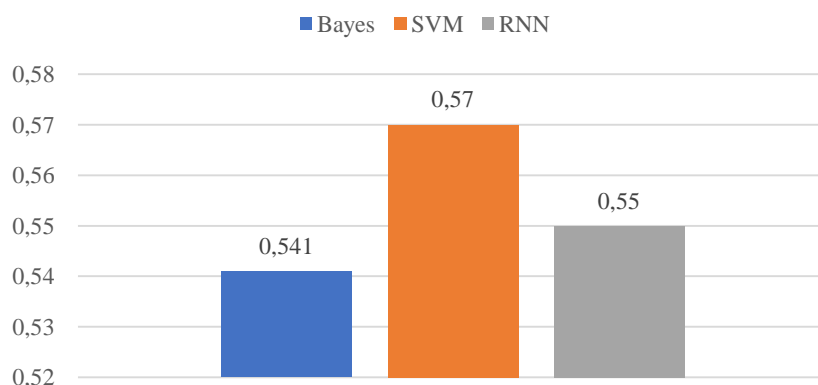


Рисунок 1 – Точность определения при 1000 записях

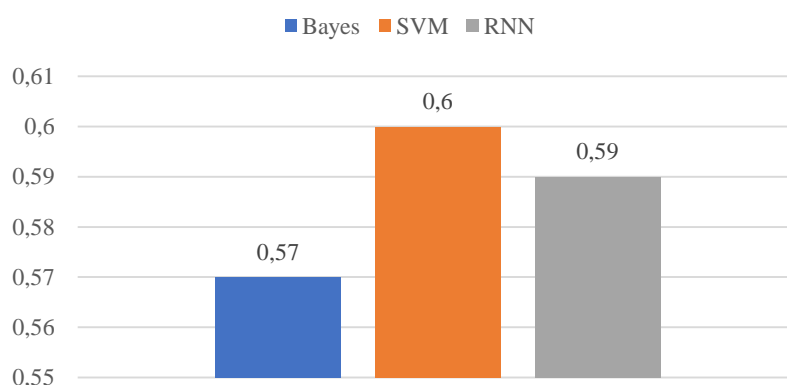


Рисунок 2 – Точность определения при 5000 записях

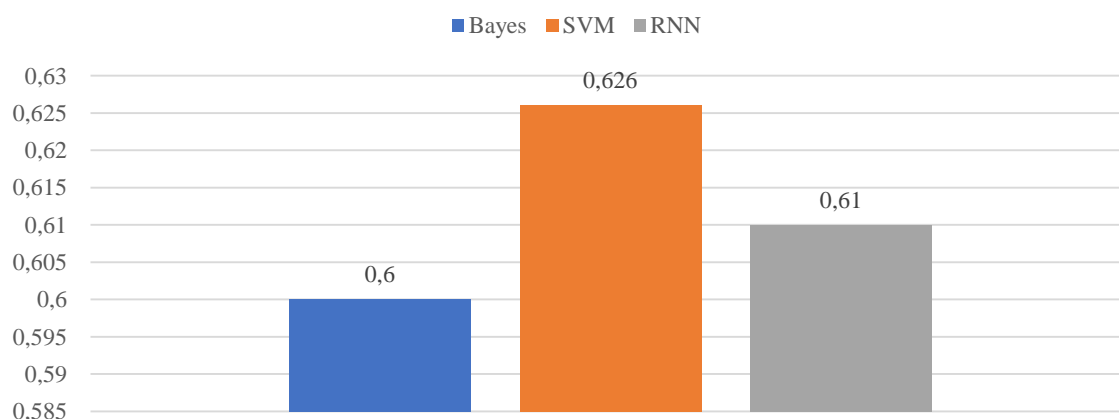


Рисунок 3 – Точность определения при 5000 записях с предобработкой

Для получения результатов на Рис.3 была проведена предобработка данных: удалены общие слова (меньше 4 букв), знаки пунктуации, использовались биграммы вместо униграмм.

По графикам можно сделать вывод, что лучший результат показывает SVM и RNN. Причем, качество RNN растет с увеличением обучающих данных. При достаточном количестве данных и увеличении классов с двух («положительный», «отрицательный») до трех («нейтральный») или переходе на оценочную систему, RNN выглядит лучшим вариантом.

Вместе с этим, можно выделить направления для улучшения результатов: комбинация n-грамм, применение TF-IDF, возможную комбинацию SVM и RNN.

ЛИТЕРАТУРА

1. Bing Liu. Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, 2012. – 141 с.
2. Пазельская А., Соловьев А. Метод определения эмоций в текстах на русском языке, Москва, 2011. – 510-222 с.
3. Recurrent Neural Network Tutorial, Part 4 – Implementing GRU/LSTM RNN with Python and Theano [электронный ресурс] / WildML – Режим доступа: <http://www.wildml.com/2015/10/recurrent-neural-network-tutorial-part-4-implementing-a-grulstm-rnn-with-python-and-theano/> - Дата доступа 09.04.2019.

Студ. А. В. Мартынюк
Науч. рук. зав. кафедрой Н. В. Пацей
(кафедра программной инженерии, БГТУ)

ПОМЕХОУСТОЙЧИВЫЕ КОДЕКИ НА ОСНОВЕ КАСКАДНЫХ СХЕМ КОДИРОВАНИЯ/ДЕКОДИРОВАНИЯ ДЛЯ СИСТЕМ ПЕРЕДАЧИ ИНФОРМАЦИИ С ОДИНОЧНЫМИ И МОДУЛЬНЫМИ ОШИБКАМИ

На текущий момент существуют тенденции построения и эксплуатации распределённых информационных вычислительных систем для решения разного класса задач [1]. Для корректного функционирования распределённых приложений необходимы устойчивые и быстрые каналы связи. Современные каналы передачи данных достаточно быстры и защищены от помех, а дополнительное применение современных помехоустойчивых кодов позволяет свести на нет вероятность неправильной передачи информации. Помехоустойчивое кодирование решает две основные задачи: 1) повышение эффективности передачи данных, за счёт устранения избыточности данных в каналах, не содержащих помехи, 2) повышение вероятности правильного декодирования пакета в канале с помехами с минимальной дополнительной избыточностью. В контексте работы рассмотрим вторую задачу применения помехоустойчивых кодов. Целью научной работы является построение имитационной модели передачи информации по каналу с шумами с применением каскадного способа построения кодеков и выявление общих закономерностей и характеристик каскадных кодеков [2-4].

В ходе работы было разработано программное средство помехоустойчивого кодера на основе последовательной каскадной схемы. Для разработки использовалась платформа python 3. Программное средство поддерживает и позволяет применить в каскаде четыре вида кодеров (Хемминга, Циклический, Свёрточный, Фонтанный), а также задать их параметры. Общая схема эмуляции дискретного канала Мелентьев связи представлена на рисунке 1. Разработанная система поддерживает использование перемежителей после каждого составного кодера.

Программное обеспечение предоставляет четыре режима тестирования кодера:

- режим одиночного кодера, при одном уровне шума;
- режим одиночного кодера с изменяющимся спектром шумов;
- режим каскадного кодера, при одном уровне шума;
- режим каскадного кодера с изменяющимся спектром шумов.

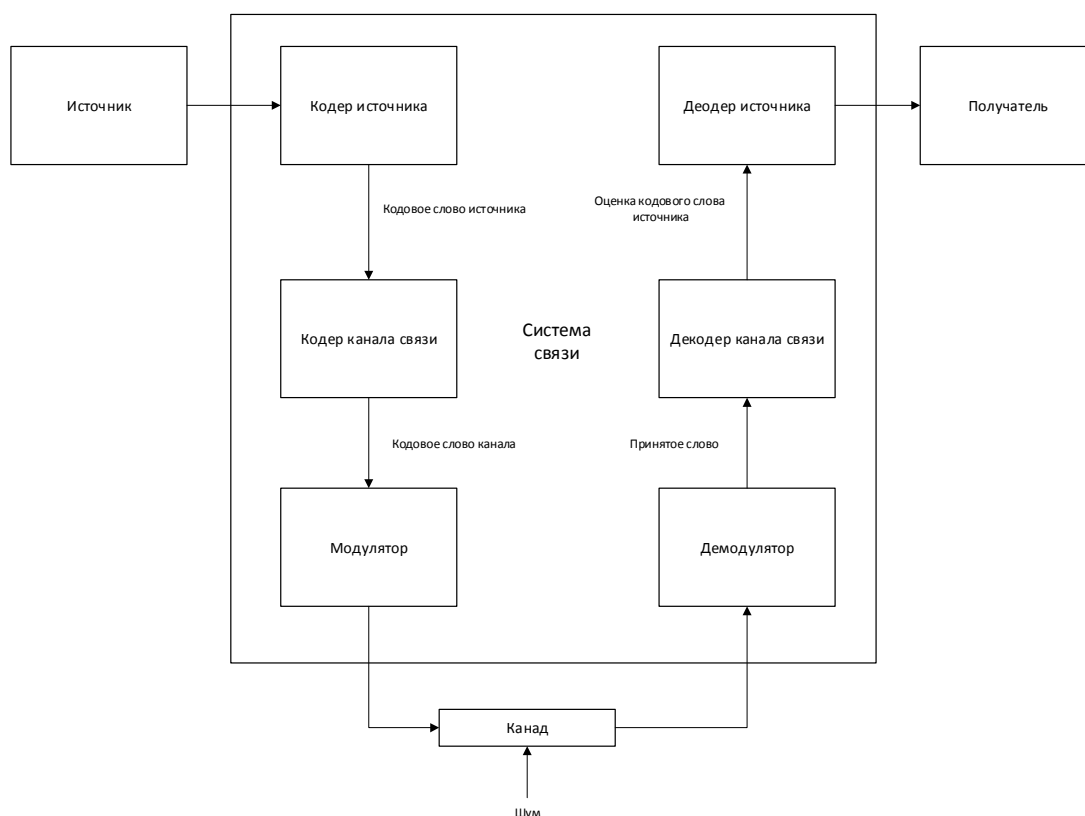


Рисунок 1 – Структурная схема дискретной системы связи

Результатом работы программы является статистика, собранная в ходе передачи данных по каналу связи и исправления ошибок. Фиксируется количество неповреждённых пакетов/бит информации, количество исправленных пакетов/бит информации, количество ошибочных и неисправленных пакетов/бит информации, а также качественные характеристики кодеров и всего кодека. После завершения цикла моделирования результаты работы записываются в базу данных PostgreSQL 5.3. Программное средство визуализирует полученные результаты в виде графика (рисунок 2). На графике отображаются отношение количества некорректных пакетов к их общему количеству, отношение количества неисправленных битов к их общему количеству, отношение количества повреждённых битов к их общему количеству (ось y), при заданном уровне шума (ось x).

В ходе работе было спроектировано и разработано программное средство, предоставляющее возможность эмуляции передачи данных по каналу с шумами при применении каскадного способа построение кодеков и выявления общих закономерностей и характеристик каскадных кодеков.

В результате проведения исследований выполнена модификация программной модели каскадной схемы. Разработаны скрипты прове-

дения автоматизированного тестирования. Выполнено тестирование модели для различных комбинаций составных кодов с перемежителями, длины пакетов, уровня шума в канале, типов ошибок (единичные, модульные и их комбинации).

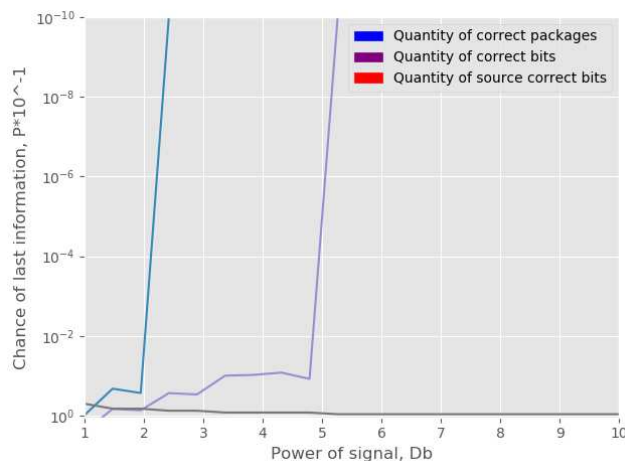


Рисунок 2 – Частота скорректированных пакетов/ошибок для разного уровня отношения сигнал/шум

Дана оценка эффективности исправления модульных и единичных ошибок в каналах связи с использованием разработанной схемы адаптивного каскадного кодирования/декодирования. Разработаны рекомендации по практическому использованию схемы адаптивного каскадного кодирования/декодирования избыточных кодов.

ЛИТЕРАТУРА

1. Блеймут Р. Б68 Теория и практика кодов, контролирующих ошибки. Пер. с англ. — М.: Мир, 1986. — 567 с., ил. / Москва «МИР» 1986.
2. Мартынюк, А.В. Сверточные коды для исправления пакетов ошибок при помехоустойчивой передаче информации // 68-я НТК студентов и магистрантов: сб. науч. работ: в 4-х ч. 17-22 апреля 2017 г. — Минск: БГТУ, 2017. — Ч. 4. — С.169-171
3. Пацей, Н.В., Мартынюк А.В. Адаптивные помехоустойчивые кодеки на основе каскадных схем кодирования/ декодирования для беспроводных систем передачи информации // материалы XIV Междунар. науч.-практ. конф «Управление информационными ресурсами», Минск, 20 декабря. 2017 г. / Акад. упр. при Президенте Республики Беларусь — Минск : Акад. упр. при Президенте Респ. Беларусь, 2017 — с.186-188.
4. Мартынюк, А.В. Реализация программной модели каскадного кодека, ориентированного на исправление многократных модульных ошибок // 69-я НТК студентов и магистрантов: сб. науч. работ: в 4-х ч. 2-13 апреля 2018 г. — Минск: БГТУ, 2018. — Ч. 4. — С.358-359.

Маг. И. В. Капелько
Науч. рук. доц. Д. В. Шиман
(кафедра программной инженерии, БГТУ)

АНАЛИЗ РАБОТОСПОСОБНОСТИ АЛГОРИТМА ТРЕХУРОВНЕВОЙ ТОКЕНИЗАЦИИ ДЛЯ АВТОМАТИЧЕСКОГО РЕФЕРИРОВАНИЯ ТЕКСТА

Автоматическое реферирование – это составление коротких изложений материалов, дайджестов, т.е. извлечение наиболее важных сведений из одного или нескольких документов и генерация на их основе лаконичных отчетов.

Для точного понимания термина реферата его можно сравнить с аннотацией. Реферат знакомит читателя с сутью содержания оригинала, используются формулировки и обобщения, заимствованные из текста оригинала. Аннотация даёт общее представление об оригинале. В аннотации отсутствует цитирование текста-источника. Основное содержание передаётся «своими словами», которые представляют собой высокую степень обобщения материала. Аннотация меньше информативного реферата в 3-4 раза. По объёму реферат всегда пространнее аннотации. Но рекомендации, касающиеся объёма обоих документов, в значительной степени варьируются. Так, для реферата разные авторы считают приемлемыми размеры от 200 до 1200 слов и сокращение текста в 3 или 8 и даже 10 раз. Реферату, состоящему из 10 – 120 слов (7 – 9 предложений), соответствуют оптимальные размеры аннотации от 40 до 60 слов (3 – 4 предложения).

По реферату можно составить мнение о содержании, сути излагаемого в оригинале содержания, аннотация даёт представление только о главной теме и о перечне вопросов, затрагиваемых в нём.

Описание работы алгоритма трёхуровневой токенизации текста. В основе алгоритма лежит метод TF-IDF (от англ. TF — termfrequency, IDF — inversedocumentfrequency) – статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса.

Перед применением каких-либо методов реферирования необходимо использовать предварительную обработку текста: удаление стоп-слов, исправление грамматических ошибок.

Под трёхуровневой токенизацией подразумевается, что токеном будет являться не только какое-то отдельное слово, но и предложение и абзац. Токеном первого уровня будет абзац текста, который включает в себя коллекцию токенов второго уровня – отдельные предложения этого абзаца. Токеном третьего уровня является слово. Для каж-

дого слова будет выделена его лемма – неизменная, исходная форма слова, а для нее рассчитана TF-IDF. Где TF – отношение числа вхождений некоторой леммы к общему числу слов документа. Таким образом, оценивается важность леммы в пределах отдельного документа. IDF – это обратная частотность документов. Она измеряет непосредственно важность термина. В моем алгоритме ее можно выразить формулой:

$$IDF = \log_{10}\left(\frac{n}{w}\right),$$

где n – количество абзацев, w – количество абзацев в которых встречается токен.

После расчёта TF-IDF у каждого слова есть свое числовое значение, выражающее его вес в тексте. Для расчёта веса предложений необходимо сложить значения токенов (слов), которые входят в состав этого предложения и разделить на их количество. Далее ту же операцию необходимо применить и для абзацев: сложить значение весов предложений и разделить на количество этих предложений в абзаце.

Таким образом, можно выделить наиболее значимые слова, а на их основе выделить наиболее важные части текста.

Анализ результатов использования алгоритма трехуровневой токенизации. В ходе написания магистерской работы было создано программное средство для использования алгоритма трехуровневой токенизации текста. Приложение создано на языке C# при помощи технологии WinForms. Так же были подготовлены различные научные и художественные тексты. Все научные тексты имели отношение к IT сфере для удобства оценки качества реферата. Для проверки работоспособности алгоритма необходимо загрузить файл в формате txt в разработанное приложение. После чего начинается предварительная обработка текста и его разделение на токены, а также расчет их весов. В предварительную обработку текста также входит процесс удаление заголовков разделов, т.к. они в большинстве случаев содержат слова значение весов, которых достаточно велики. В результате на экран пользователю выводится таблица с результатами (рисунок 1).

На данном этапе можно посмотреть отсортированные списки токенов всех уровней. Далее необходимо выбрать процент реферата от основного текста и нажать кнопку показать результат. После этого происходит расчет необходимого числа параграфов, сортировка их по весам, а затем сортировка по номеру абзаца в тексте. Реферат выводится в отдельном окне.

Текстовое значение	Вес
Тип задачи определяет метод, наиболее подходящий для ее решения. Задачи, которые сводятся к процедуральному анализу, вообще говоря, лучше всего решаются на компьютере. Учетные и аналитические задачи служат примера...	0.003316589948...
Тип задачи определяет метод, наиболее подходящий для ее решения. Задачи, которые сводятся к процедуральному анализу, вообще говоря, лучше всего решаются на компьютере. Учетные и аналитические задачи служат примерами...	0.003316589948...
газунно сочетая такие критерии (например, в виде линейной комбинации с экспериментально подобранными коэффициентами или более сложным образом); можно для оценки очередного хода машины получить некоторый число...	0.003293017011...
Второй подход в качестве объекта исследования рассматривает искусственный интеллект. Здесь речь идет о моделировании интеллектуальной деятельности с помощью вычислительных машин. Целью работ в этом направлении...	0.003309067296...
Перцептрон или любая программа, имитирующая процесс распознавания, работают в двух режимах: в режиме обучения и в режиме распознавания. В режиме обучения некто (человек, машина, робот или природа), играющий роль у...	0.003240810314...
Наконец, третий подход ориентирован на создание смешанных человеко-машинных, или, как еще говорят, интерактивных интеллектуальных систем, на симбиоз возможностей естественного и искусственного интеллекта. Важней...	0.003229308180...
Создание общей теории или метода представления значений является стратегической проблемой. Такая теория открыла бы возможность накопления знаний, которые нужны ежедневно для решения все новых и новых задач. Однак...	0.003057446559...
Эти работы положили начало исследованиям в области ИИ, связанному с разработкой программы, решающих задачи на основе применения разнообразных эвристических методов и правил. Эвристика – совокупность логических пр...	0.002974656494...
Каким образом машине удалось достичь столь высокого класса игры?	0.002973863825...
Решение задач сводится к поиску пути из некоторой исходной точки в целевую. Человек делает это весьма эффективно с помощью дедуктивного логического вывода (рассуждения), процедурного анализа, аналогии и индукции...	0.002891064414...
Ярким примером сложной интеллектуальной игры до недавнего времени являлись шахматы. В 1974 г. состоялся международный шахматный турнир машин, снабженных соответствующими программами. Как известно, победу на э...	0.002827017875...
Представление значений – наиболее важная область исследований по искусственному интеллекту, основа всех остальных дисциплин. Знания имеют форму описаний объектов, взаимосвязей и процедур. Наличие адекватных знаний ...	0.002815471081...
В настоящее время существуют и успешно применяются программы, позволяющие машинам играть в деловые или военные игры, имеющие большое прикладное значение. Здесь также чрезвычайно важно придать программам пр...	0.002753856578...
Данный метод решения задачи при этом рассматривался как свойственный человеческому мышлению «вообще», для которого характерно возникновение «догадок» о пути решения с последующей проверкой их. Эвристическому ...	0.0026872071...
Самыми первыми интеллектуальными задачами, которые стали решаться при помощи ЭВМ были логические игры (шахки, шахматы), доказательство теорем. Хотя, правда здесь надо отметить еще кибернетические игрушки типа "...	0.00261121195...
Можно сказать, что все эти элементы интеллекта, продемонстрированные машиной в процессе игры в шахки, сообщены ей автором программы. Странно это так. Но не следует забывать, что программа эта не является "жесткой" ...	0.002578815017...
Тем не менее, в последнее десятилетие это направление возродилось в виде исследований и разработок, направленных на создание экспертных систем с базой знаний. Их используют в управленческой деятельности и многих от...	0.002482027919...
Исторически сложились три основных направления в моделировании искусственного интеллекта.	0.002445914064...
Почему здесь употреблено "до недавнего времени"? Дело в том, что недавние события показали, что, несмотря на довольно большую сложность шахмат, и невозможность, в связи с этим произвести полный перебор ходов, возмо...	0.002369614652...
Американский кибернетик А. Самуэль составил для вычислительной машины программу, которая позволяет ей играть в шахки, причем в ходе игры машина обучается или, по крайней мере, создает впечатление, что обучается, улу...	0.002354650816...
Проблематика ИИ в настоящее время довольно обширна. Список Дисциплин по искусственному интеллекту постоянно увеличивается. Сегодня в него входят представление значений, решение задач, экспертные системы, средства о...	0.002288720714...
В 1957 г. американский физиолог Ф. Розенблат предложил модель зрительного восприятия и распознавания – перцептрон. Появление машины, способной обучаться понятиям и распознавать предъявляемые объекты, оказалось...	0.002240726298...
Начало современного этапа развития систем искусственного интеллекта (ИИ) может быть отнесено к середине 50-х гг. Этому способствовала программа, разработанная А. Ньюэллом, предназначенная для доказательства теорем...	0.002165280717...
В 70–80 гг. исследования в области ИИ характеризовались перемещением внимания специалистов от проблем создания автономно функционирующих систем к созданию человеко-машинных систем, интегрированных в единое цело...	0.002137412300...
В рамках первого подхода объектом исследований являются структура и механизмы работы мозга человека, а конечная цель заключается в раскрытии тайн мышления. Необходимыми этапами исследований в этом направлении ...	0.001966202237...
По мнению А. Самуэля, машина, использующая этот вид обучения, может научиться играть лучше, чем средний игрок, за относительно короткий период времени.	

Рисунок 1 – Таблица весов токенов

В процессе проверки работоспособности алгоритма было выявлено, что наиболее информативный реферат получается только для научных текстов, т.к. у художественных текстов невозможно выделить только несколько основных абзацев. Оптимальный объем реферата от 15 до 20% от основного текста.

ЛИТЕРАТУРА

1. Miner. Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications. Academic Press, 2012 стр. 71–138.
2. TF-IDF с примерами кода: просто и понятно [Электронный ресурс] / NLPx – Режим доступа: <http://nlp.net/archives/57>. – Дата доступа 10.04.2019.
3. Fayyad U., Piatetsky-Shapiro G., Smyth P. From Data Mining to Knowledge Discovery: an Overview // Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996. – стр. 1-34.

УДК 004.41

Магистрант О. Л. Панченко
Науч. рук. зав. кафедрой Н. В. Пацей
(кафедра программной инженерии, БГТУ)

МЕТОДЫ ОПЕРАТИВНОЙ КЛАССИФИКАЦИИ ТЕКСТОВЫХ ДОКУМЕНТОВ

Классификация текстов – одна из важных задач информационного поиска заключающаяся в отнесении документа к одной или нескольким категориям из заранее определенного набора на основе анализа содержания этого документа.

Для решения задач текстовой классификации используются методы машинного обучения. Они отличается такими особенностями как:

- высокая скорость классификации;
- снижение роли человека в процессе получения решения.

В работе был выбран именно этот подход. Поставлена цель: экспериментально исследовать методы автоматизированной классификации текстовых документов.

А также ряд задач: проанализировать и обобщить существующие методы автоматизированной классификации текстовых документов (КТД); оптимизировать один из существующих методов КТД; реализовать программное средство для исследования характеристик существующих методов КТД; провести исследование по быстродействию, точности классификации и функциональных возможностей в сравнении с своим доработанным методом.

Для реализации программного средства был выбран язык Python. Так как он уже имеет ряд вспомогательных библиотек таких как:

- IPython – это интерактивная оболочка с широким набором возможностей и ядро для Jupyter. Jupyter notebook является графической веб-оболочкой для IPython;
- Scikit-learn для работы с классическими алгоритмами машинного обучения;
- Pandas для извлечения и подготовки данных;
- Matplotlib для визуализации данных.

В качестве исходных данных были взяты тематические новости которые были собраны в наборы данных в файлы формата csv. Таких набора два: один для обучения модели (train.csv), а второй для проведения тестов (test.csv)

Данные файле размешены следующим образом: первым идет номер класса к которому принадлежит текст от 1 до 4 (так как у нас 4

класса World, Sports, Business, Sci/Tech). далее идет заголовок новости и потом уже сам текст статьи (рисунок 1).

```
"3", "Building code back in hot seat", "With insurance claim projections fr
"2", "UPDATE 1-Gibernau cruises to victory at Czech GP", "Spaniard Sete Gib
"1", "Singapore #39;s new PM delivers national day rally speech", "Singapor
"3", "Marks and Spencer loses crown as Britain's top clothing retailer (AF
"2", "Gymnast Khorkina Says 'Judges Robbed Me'", " MOSCOW (Reuters) - Russi
"3", "Stocks May Rally if Oil Eases", " NEW YORK (Reuters) - Investors will
"1", "COTE D IVOIRE: All sides pledge commitment to peace process again &l
"1", "Palestinians Chide U.S. Over Settlements", "JERUSALEM - Palestinian l
"2", "Greek Weightlifter Stripped of Olympic Medal, Ejected From Athens &l
"2", "England humble Windies", "ONCE the tailenders of world cricket, Engla
"1", "Iraq clashes kill 40, handover talks stall", "US tanks rumbled to witi
"1", "Train set ablaze as violence spreads in Bangladesh", "Dhaka, Aug 22.
"1", "North Korea Denounces Mass Defection", "North Korea has denounced as
"3", "Judge gives United temporary reprieve", "A federal bankruptcy judge h
"2", "UPDATE 1-Gibernau storms to pole for Czech GP", "Spaniard Sete Gibern
"2", "Swiss pair eliminates Holdren, Metzger", "Stein Metzger screamed, as
"1", "Mob sets fire to train in protest at attack", "An angry mob set fire
"3", "BA prepares new sick leave deal", "British Airways says it will intro
"3", "Asda clothing overtakes M amp;S", "Marks amp; Spencer is no longer tl
"3", "India News gt; Trucker #39;s strike enters second day:", "The Delhi
"4", "Amazon.com to Acquire Retailer Jyo.com", "Internet retail giant Amaz
```

Рисунок 1 - Файл с входными данными

Для классификации были применены следующие методы: Naive Bayes: Bernoulli Naive Bayes, Multinomial NB; SVM (support vector machine): Linear SVM.

На рисунке 2 представлен процесс работы программы.



Рисунок 2 - Процесс работы ПС

На рисунке 3 представлены результаты работы методов, а также коэффициент точности. Можно увидеть, что наилучший результат показал метод SVM.

```
Показатель точности Бернулли
0.8902631578947369
Результат классификатора
['Business' 'Science and Tech' 'Science and Tech
'Science and Tech']
print (accuracy_score(test_lbl, ypredMnb))
Multinomial accuracy score
0.8935526315789474

print (accuracy_score(test_lbl, ypredLsvm))
Linear Svm accuracy score
0.91
```

10

Рисунок 3 - Показатели точности для каждого из методов

По итогам полученных точностей была построена диаграмма, представленная на рисунке 4. На ней отображены показатель точности по каждому методу в соответствии с количеством классов. Далее были построены матрицы путаницы. На рисунке 4 представлена для метода SVM. Числа по диагонали, они еще выделены красным показывают количество корректно классификации для каждого класса. А те что не по диагонали показывают ошибочные классификации (например, 140 кл. Sport было неправильно клас-но как кл. World).

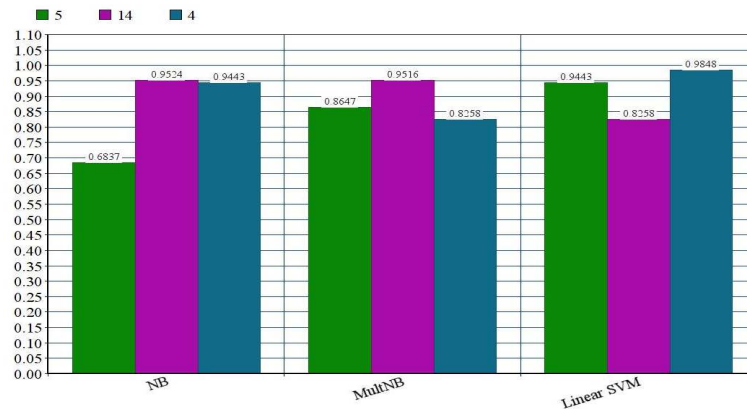


Рисунок 4 – Диаграмма показателей точностей по классам

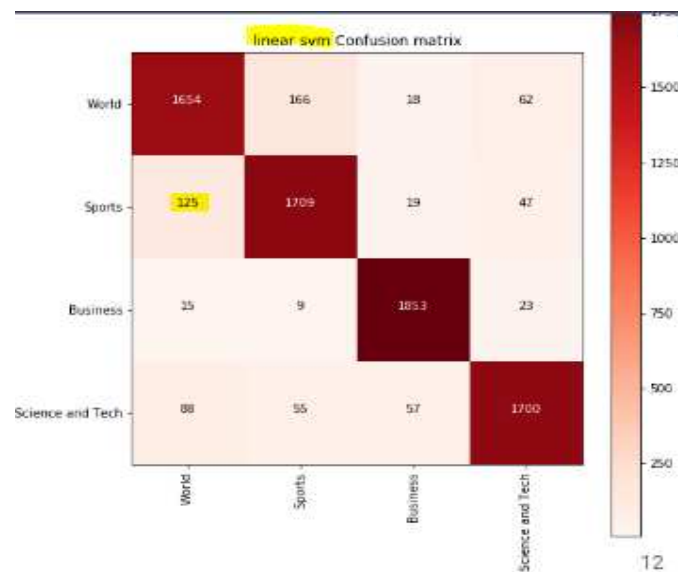


Рисунок 4 - Матрица путаницы

Из матрицы так же было видно, что у метода SVM было меньше всего ошибочных классификаций.

Подводя итог, на данном наборе данных и данными набором классов наилучший результат в текстовой классификации показал метод SVM.

В дальнейшем планируется доработать метод SVM. Результаты, которые будут получены в ходе его работы сравнить с уже имеющимися. Добавить больше алгоритмов. Прodelать тесты на других наборах данных с другим количеством классов.

ЛИТЕРАТУРА

1. Методы автоматической классификации текстов
https://www.researchgate.net/publication/315328102_Metody_avtomaticheskoy_klassifikatsii_tekstov.

Студ. Д. С. Ничипорчик
Науч. рук. доц. Д. В. Шиман
(кафедра информационных систем и технологий, БГТУ)

ОСАЖДЕНИЕ ИНФОРМАЦИИ В ИЗОБРАЖЕНИЯХ

Одной из важнейших задач современности является обеспечение надежности хранения и передачи данных. С каждым днем ответственность за передачу некорректной информации растет, при этом увеличивается количество способов несанкционированного доступа к информации. Данная работа ставит своей целью изучение современных методов шифрования данных.

Основная часть. Шифрование данных широко применяется для различных целей, главные из которых – обеспечение конфиденциальности, защищенности данных, передаваемых компаниями и частными пользователями. Данная научная работа является демонстрационной смесью шифрования и стеганографии. Для шифрования использовался RSA. [1] RSA (аббревиатура от фамилий Rivest, Shamir и Adleman) — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел. Весь алгоритм можно разбить на два блока действий: преобразования над текстом и преобразования над изображением.

В данном примере взято изображение размера 27 x 12 пикселей. Для большей наглядности будет использоваться строка “дом” и однотонное изображение (рисунок 1).

Преобразования над текстом: преобразования над текстом условно можно разбить на несколько последовательных действий: шифрование с использованием RSA и преобразование с использованием таблицы ASCII.

Шифрование с использованием RSA: после зашифровки мы получим следующую строку – 4d61340690daa130f08825ae42984eff, размер которой 32 символа.

Преобразование с использованием таблицы ASCII: на данном этапе мы уже имеем зашифрованную строку, размером 32 символа, наша задача перевести данный набор в символы, а после в шестнадцатеричную систему счисления, (для чего будет рассмотрено в преобразовании изображения). Рассмотрим на примере первого символа: 4 в таблице ASCII 52, в шестнадцатеричной системе счисления это 34, таким образом мы получили первое значение, которое будем использовать в «осаждении».

Конечным результатом будем являться 34, 64, 36, 31, 33, 34, 30, 36, 39, 30, 64, 61, 61, 31, 33, 30, 66, 30, 38, 38, 32, 35, 61, 65, 34, 32, 39, 38, 34, 32, 39, 38, 34, 65, 66, 66.

Преобразования над изображением: Преобразования над текстом представляют собой несколько последовательных действий: разбиение изображение на условные квадраты 3x3 пикселя, разбиение этих квадратов на квадраты 1x1 пикселя, “осаждение” (помещение информации) пикселя в центре квадрата 3x3 пикселя, формирование финального изображения. Рассмотрим каждый шаг более подробно.

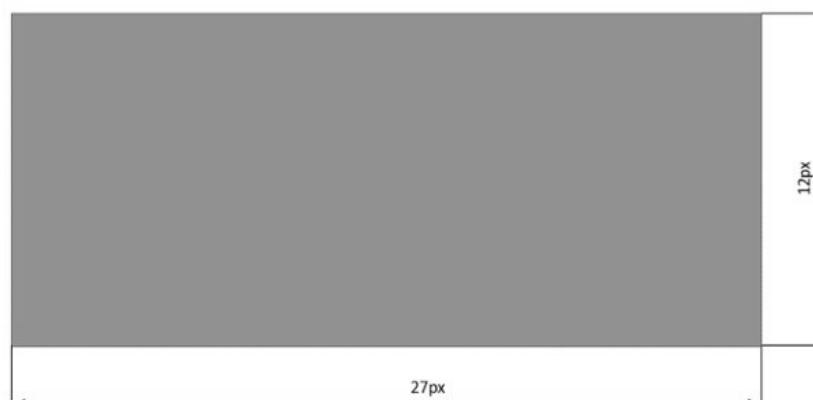


Рисунок 1 - Исходное изображение

В таблице 1 приведены соотношения популярных размеров и количество символов, которые можно поместить.

Таблица 1 - Соотношения размеров и количества символов

Разрешение	Количество символов
1366x768	116480
1600x900	159900
1920x1080	230400

Для произвольного размера количество символов, которые мы можем поместить в изображение мы рассчитываем по формуле (1).

$$l = (w \bmod 3) * (h \bmod 3) - 1, \quad (1)$$

где w – ширина изображения; h – высота изображения.

Разбиение изображения на условные квадраты размера 3x3 пикселя: Наш цикл проходит по всему изображению разбивая его на квадраты размером 3x3 пикселя, в данном примере их 36. Размер нашего ввода 32 символа, а это значит, что коллизия, связанная с нехваткой мест для “осаждения”, не возникнет (см. рис.2).

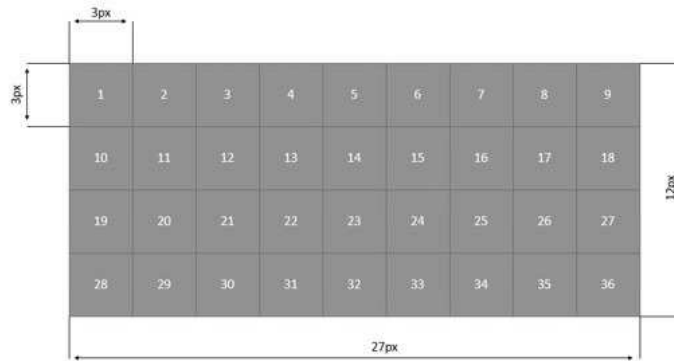


Рисунок 2 - Изображения, разбитое на квадраты 3x3

Разбиение каждого квадрата на квадраты размерами 1x1 пикселя: в данном шаге уже сформированные ячейки разбиваются на более маленькие, размерами 1x1 пикселя, для будущего осаждения. Разбиение происходит слева направо. “Осаждение” пикселя в центре квадрата 3x3 пикселя: на данном этапе первым действием является проверка каждой ячейки на коллизию. Если ее размер не равен 9, то она пропускается, а если равен 9, то алгоритм действий следующий: вычисляется значения цвета каждой ячейки в HEX, а также среднее значения всех ячеек без центральной (пятой) ячейки. В данном примере значение цвета равно #5B9BD5, соответственно и среднее равно #5B9BD5. Теперь мы должны сложить первое среднее значение с 1 символом 34. Мы получим следующее значение: 5B9C09. Далее применим этот цвет к 1 ячейке.

Формирование финального изображения: на данном этапе мы имеем уже сформированное изображения, но встает вопрос, как же отличить модифицированные ячейки от исходных? Для этого используется следующий алгоритм: при проверке мы используем преобразованный в шестнадцатеричную систему счисления ключ, а именно сумму всех элементов.

Формирование стоп-символа: для формирования стоп-символа используется следующий алгоритм: мы берем закрытый ключ, производим те же преобразования, что над текстом, а после складываем сумму всех элементов. Рассмотрим формирование стоп-символа для данного примера: 5ad17cbe5d90df75c45e4ba1f16bfe69.

При преобразовании каждого символа с использованием таблицы ASCII мы получим следующие значения: 53, 97, 100, 49, 55, 99, 98, 101, 53, 100, 57, 48, 100, 102, 55, 53, 99, 52, 53, 101, 52, 98, 97, 49, 102, 49, 54, 98, 102, 101, 54, 57. После этого требуется получить сумму всех чисел, в данном случае мы получим 2438. Это слишком много, поэтому мы должны поделить нацело на 256. Получим 134, после этого прибавим 256 для того, чтобы ключ не совпадал с текстом. Мы получим 390 и переведем число в шестнадцатеричную систему счисления,

получим 186. Цвет исходной ячейки 5B9BD5, прибавив 186 получим 5B9D5B (см. рис. 3).

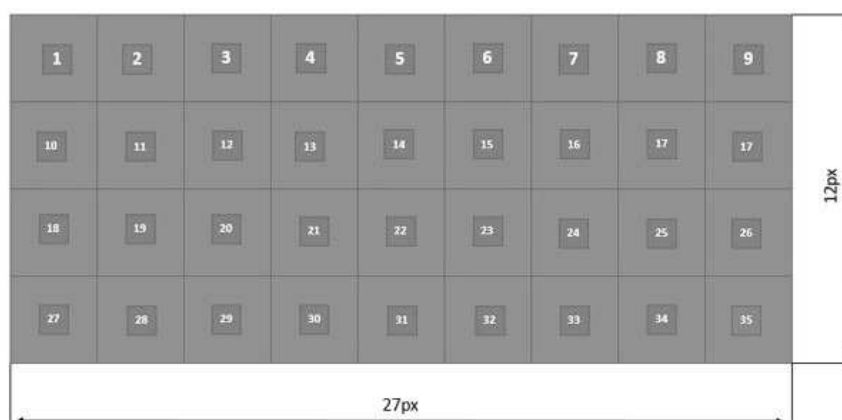


Рисунок 3 - Финальное изображение

Вывод: в данной научной работе рассмотрены варианты совместного использования “осаждения” и шифрования. Данный способ является очень полезным из-за своей универсальности, так как изображения присутствуют во всех сферах реальной жизни, что в свою очередь поможет скрыть факт шифрования данных в изображении.

ЛИТЕРАТУРА

1. Конахович Г. Ф., Пузыренко А. Ю. Компьютерная стеганография. Теория и практика. — К.: МК-Пресс, 2006. — 288 с
2. Грибунин В. Г., Оков И. Н., Туринцев И. В. Цифровая стеганография. — М.: Солон-Пресс, 2002. — 272 с, ил.

УДК 004.042

Студ. Э. В. Ральцевич

Науч. рук. канд. техн. наук Н. А. Жиляк
(кафедра информационных технологий БГТУ)

РАЗРАБОТКА ВЕБ-РЕСУРСА «RESOURCE PLAINNING SYSTEM»

Сегодня большинство компаний занимается предоставлением аутсорсинговых услуг. Продуктом, который продает такая компания, является человеческий ресурс. По этой причине руководители данных компаний нуждаются в программном средстве, которое позволяло бы отобразить информацию о сотрудниках, их занятости и времени, когда сотрудник освобождается. Проводя небольшое исследование, было обнаружено, что существующие приложения не рассчитаны на то,

чтобы ресурсом являлся человек. По этой причине было решено разработать собственное веб-приложение, которое позволяло бы вести учет сотрудников и список занятости внутри компании.

Перед началом разработки были рассмотрены различные способы построения структуры приложения. На сегодняшний день существует многочисленное количество различных подходов, например, монолитное приложение (все сервисы жестко связаны между собой и приложение не разбивается на уровни абстракции) и, например, модульный подход разработки (приложение разбивается на уровни абстракции и каждый уровень слабо связан с предыдущим). Вторым подходом является приоритетным, так как во время разработки на базе него внесение изменений в функциональность приложения упрощается в несколько раз и не влечет за собой изменение существующего кода. Для разрабатываемого приложения были выделены следующие уровни абстракции: уровень данных, уровень доступа к данным, уровень графического интерфейса.

Приложение будет состоять из двух отдельных частей: серверной части (реализующая *API* для доступа к данным) и клиентской части (предоставляющая пользователю графический интерфейс взаимодействия с серверной частью).

Серверная часть разрабатывалась в виде открытого *API*. Для разработки использовался *Node.js* [1]. Данная технология предназначена для разработки небольших коммерческих веб-приложений. Использование данной технологии позволяет разворачивать приложение на различных операционных системах, а именно: *Windows*, *Mac OS X*, *Linux*. Это означает, что данная технология является кросс-платформенной. Для взаимодействия с базой данных *MySQL* использовалась *ORM*-система *Sequelize*. Данная позволяет взаимодействовать с базой данных на уровне схемы.

Для разработки графической части приложения использован подход *Single Page Application*, основная особенность данного подхода указывает на то, что при переходе между элементами приложения, единственный обмен данными который осуществляется предназначен для получения данных от серверной части приложения, все статические файлы загружаются. Для реализации данной части приложения был выбран фреймворк *React* [2].

Во время первого запуска приложения пользователь попадает на страницу входа, на которой он должен ввести свой логин и пароль. Данная страница изображена на рисунке 1.

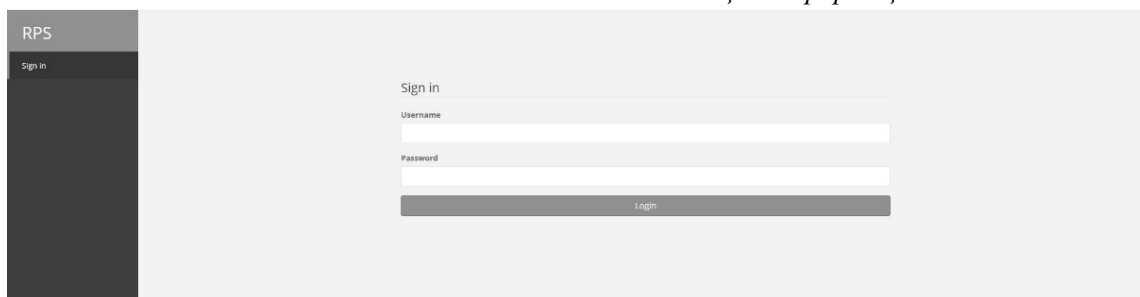


Рисунок 1 – Страница входа в приложение

После того как пользователь совершит вход в приложение ему открывается доступ к списку сотрудников, вид страницы отображен на рисунке 2.

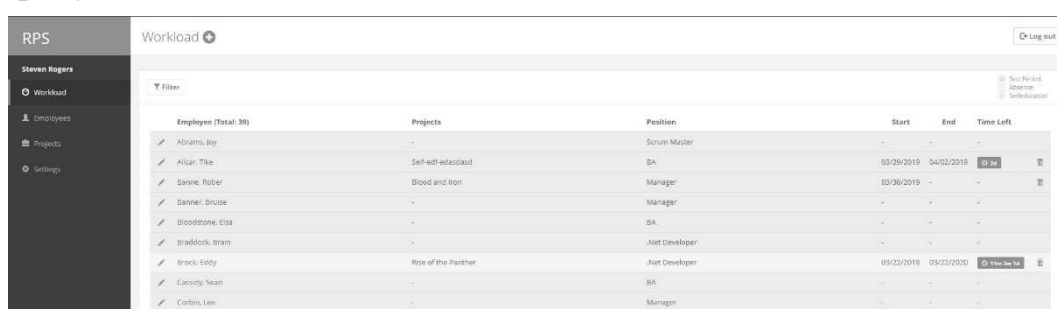


Рисунок 2 – Страница отображения списка сотрудников

По результату выполненной работы, было разработано веб-приложение, которое позволяет производить распределение человеческих ресурсов внутри компании.

ЛИТЕРАТУРА

1. Официальная документация Nodejs – ode.js Everywhere: [Электронный ресурс]. – 2019. – Режим доступа: <https://nodejs.org>. – Дата доступа: 20.03.2019.
2. Официальная документация React – React Documentation: [Электронный ресурс]. – 2019. – Режим доступа: <https://reactjs.org>. – Дата доступа: 20.03.2019.

УДК 004.042

Студ. А. А. Ермакович
 Науч. рук. канд. техн. наук Н. А. Жилияк
 (кафедра информационных технологий БГТУ)

РАЗРАБОТКА ВЕБ-РЕСУРСА «ОТОБРАЖЕНИЕ АНАЛИТИЧЕСКИХ ДАННЫХ ФАРМАЦЕВТИЧЕСКИХ КОМПАНИЙ»

На сегодняшний день современная фармацевтическая промышленность из обычного производства лекарств для лечения болезней превратилась в большие числа, а именно: прибыль, убытки, спрос. По

этой причине каждая компания, которая занимается производством и продажей лекарственных средств, хотела бы видеть актуальную информацию о продажах своих продуктов по территориальным и временным критериям. По этой причине появились компании, которые выполняют анализ по продаваемости лекарств, количеству заболевших той или иной болезнью, на различных территориях земного шара. Одной из таких компаний является «IQVIA». Конечно каждая фармацевтическая компания, которая обращается к «IQVIA», хочет не только слышать различные цифры о продажах и так далее, а хочет видеть данную информацию в виде красивых графиков и иметь возможность в любой момент их просмотреть самостоятельно. В связи с этим было решено разработать приложение, которое выводит аналитические данные в виде ключевых показателей эффективности.

Перед началом разработки были рассмотрены различные подходы к построению архитектуры приложения. Из многочисленных вариантов был выбран подход разделения приложения на многочисленные уровни и модули, а именно: уровень данных, уровень доступа к данным, уровень пользовательского интерфейса. Данный подход был выбран по причине того, что основное его преимущество заключается в возможности простого внесения изменений в отдельные части приложения, а также упрощения расширения функционала с помощью добавления новых компонентов, не изменяя старые.

Приложение состоит из двух частей: серверной и клиентской.

Серверная часть приложения разработана в виде *RESTful API* на платформе *ASP.NET Core 2.2* [1]. Данная технология предназначена для разработки коммерческих веб-приложений. *ASP.NET Core* получила свою популярность из-за возможности запуска приложения, написанного на данной технологии, в различных операционных системах, таких как: *Windows, Mac OS X, Linux*, т.е. является кросс-платформенной средой разработки. Для работы с базой данных был использован фреймворк *Entity Framework Core* от компании *Microsoft*. Выбор данного фреймворка был обусловлен тем, что он реализует уже многочисленные паттерны разработки и позволяет работать с базой данных, как с простым набором объектов без явной необходимости писать *SQL*-запросы, т.е. позволяет работать с данными на высоком уровне абстракции.

Клиентская часть приложения разрабатывалась как *Single Page Application*, что позволяет не обновлять страницы при переходе между элементами меню, что ускоряет работу приложения, так как обмен данными заключается только в получении информации от *API*-сервера. Для реализации был использован фреймворк *Angular* вер-

сии 5 от компании *Google* [2]. Для написания исходного кода был использован язык программирования *TypeScript*, так как он является строго типизированным и упрощает разработку приложения. Данный фреймворк представляет собой высокоуровневую надстройку для *Javascript*.

При запуске приложения пользователь попадает на страницу с графиками. На данной странице пользователь может посмотреть интересующую его информацию по выбранным критериям, пользовательский интерфейс данной страницы изображен на рисунке 1. При работе с приложением пользователь может изменить критерий по которому будет отображаться информация, а именно: продукт, регион и временной период. Данные, которые отображаются пользователю, берутся из баз данных, список баз данных указывается при разворачивании приложения.

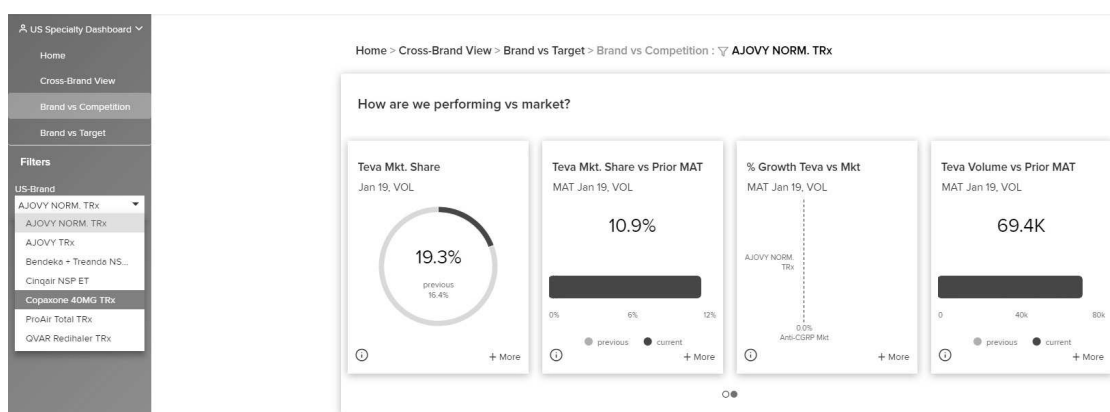


Рисунок 1 – Страница отображения графиков

Результатом разработки стало веб-приложение, которое состоит из двух частей: клиентской, написанной с помощью фреймворка *Angular*, и серверной на платформе *ASP.NET Core* и базы данных *MS SQL Server*.

Во время разработки веб-приложения использовались популярные среды разработки *Visual Studio Community 2017*, *Visual Studio Code* и *Microsoft SQL Server Management Studio*, а также *NPM* и *Angular CLI*.

ЛИТЕРАТУРА

1. Официальная документация Microsoft - Introduction to ASP.NET Core: [Электронный ресурс]. – 2019. – Режим доступа: <https://docs.microsoft.com/enus/aspnet/core>. – Дата доступа: 20.03.2019.
2. Официальная документация Angular – Angular Documentation: [Электронный ресурс]. – 2019. – Режим доступа: <https://angular.io/docs>. – Дата доступа: 20.03.2019.

УДК 004.442

Студ. Д. А. Плотников
Науч. рук. ст. преп. А. С. Наркевич
(кафедра программной инженерии, БГТУ)

ОБЗОР АРХИТЕКТУРЫ LINUX X64

Низкоуровневым программированием обычно называют программирование процессора или контроллера, который непосредственно связан своими периферийными устройствами со специфической внешней аппаратурой. Обычно такое программирование ведется на языках низкого уровня, например, таком как язык ассемблера. На сегодняшний день существует множество архитектур и вместе с этим современные компиляторы с языка ассемблера транслируют код в тот, который может быть исполнен на процессоре, аббревиатура архитектуры которого передается с параметром запуска.

Поэтому современное низкоуровневое программирование под ПК зависит как правило лишь от разрядности процессора и особенностей компилятора, который используется. Ниже приведен список популярных ассемблеров:

- MASM;
- TASM;
- FASM;
- NASM;
- AS (Unix).

Рассмотрев код, написанный на каждом из них можно заметить много общего, за исключением последнего компилятора, являющийся стандартным практически во всех разновидностях UNIX, в том числе Linux и BSD. Эти два, казалось бы, абсолютно разные на вид синтаксиса получили название:

- Intel-синтаксис (MASM, TASM, встроенный ассемблер Visual Studio, и т.д.);
- AT&T-синтаксис (GAS, при установке соответствующих флагов можно и NASM)

В первом случае синтаксис не сильно отличается от компилятора к компилятору, общие черты остаются теми же. Второй, как правило, используется только в Unix системах.

Основные отличия AT&T от Intel:

- комментарий символа начинается с символа «#», а не «.», в свою очередь символ «;» разделяет команды и позволяет записывать несколько команд в одной строке;

- отсутствие префикса операнда указывает на адрес в памяти, поэтому `movl $foo,%eax` помещает адрес переменной `foo` в регистр `eax`, а `movlfoo,%eax` помещает в `eax` содержимое переменной `foo`;
- имена регистров начинаются с символа `%`, то есть `%eax`, `%dl`, вместо `eax`, `dl`, и т.д. Это позволяет включать в код внешние переменные `C`, не опасаясь ошибок и не используя префиксов с подчеркиванием
- размер операнда определяется как суффикс имени инструкции.

В таблице 1 и таблице 2 приведены основные отличия AT&T и Intel синтаксиса в архитектуре x86 на примере команды пересылки данных.

Таблица 1 – Отличия синтаксиса Intel и AT&T для операндов типа «регистр-регистр»

Intel syntax x86	AT&T syntax x86	Human
<code>mov eax, ebx</code>	<code>movl %ebx, %eax</code>	<code>eax = ebx</code>

Таблица 2 – Отличия синтаксиса Intel и AT&T для операндов типа «память-непосредственное значение»

Intel syntax	AT&T syntax
<code>mov BYTE PTR [ESI], 5</code>	<code>movb \$5, (%esi)</code>

Важным моментом при переходе из x86 в x64 являются расширенные регистры. Они имеют расширенные до 64 разрядов версии. Например, 64-разрядная версия регистра `eax` именуется как `rax`. Как и раньше можно обращаться к младшей части регистра используя те же имена. В 64-разрядном режиме доступно 16 целочисленных регистров общего назначения и добавились новые регистры с именами от `R8` до `R15`. Принцип обращения к младшим 8-, 16- и 32-битам регистров общего назначения приведен на рисунке 1.

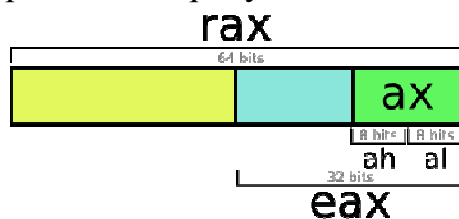


Рисунок 1 – Схема обращения к именованным частям регистра RAX

Благодаря такому подходу, код, написанный для x32 архитектуры, может быть скомпилирован и выполнен на 64-разрядной ОС.

В отличие от x32, x64 Unix архитектура использует лишь одно соглашение о вызовах, называемое System V AMD64 ABI. В нем первые шесть параметров функций передаются исключительно через регистры: RDI, RSI, RDX, RCX, R8, R9, XMM0–7. Если функция имеет более шести параметров, оставшиеся передаются через стек в порядке справа налево.

Изложенные в статье материалы были использованы при разработке компилятора с собственного языка программирования PDA-2018 выполненной в 64- разрядной ОС Linux. На выходе файлом компилятор файл на GNU ассемблере в архитектуре x64. Для удобства работы с компилятором создана IDE (среда разработки) с использованием QT версии 5.11.

ЛИТЕРАТУРА

1. Соглашение о вызовах x64. Свободная энциклопедия Википедия: [Электронный ресурс] – Электронные данные. Режим доступа: https://en.wikipedia.org/wiki/X86_calling_conventions
2. GNU_Assembler. Свободная энциклопедия Википедия: [Электронный ресурс] – Электронные данные. Режим доступа: https://en.wikipedia.org/wiki/X86_assembly_language
3. GAS_Syntax. Свободная энциклопедия Википедия: [Электронный ресурс] – Электронные данные. Режим доступа: https://en.wikibooks.org/wiki/X86_Assembly/GAS_Syntax
4. Advanced Programming in the UNIX Environment third edition - W.Richard Stevens, Stephen A. Rago

УДК 004.056

Студ. Ю. С. Богдан
Науч. рук. ассист. О. Л. Панченко
(кафедра программной инженерии, БГТУ)

БЭК-ОФИС СИСТЕМА ДЛЯ УПРАВЛЕНИЯ СЕТЬЮ МАГАЗИНОВ

Понятие система бэк-офис (back-office) прочно закрепилось в сфере ритейла. Это тот инструмент, который позволяет консолидировать и согласовать все бизнес-процессы. В битве за кошельки покупателей касса – фронт-офис (front-office) – является передней линией. Тылом, который обеспечивает командование, является бэк-офис – торговая система предприятия, охватывающая все процессы, проте-

кающие в магазине – от обеспечения продаж до аналитической отчетности. От того, насколько она будет соответствовать задачам, стоящим перед конкретным предприятием, и насколько эффективно будет работать, зависит, окажется ли владелец магазина победителем в этой битве.

В наше время бэк-офис – это полноценная система управления предприятием, отвечающая за все основные бизнес-процессы на торговом предприятии. Бэк-офисная система должна в себя включать складской учёт, управление взаимодействием с поставщиками, ценообразование, подготовку товара к реализации, дисконт, взаимодействие с торговым оборудованием, реализацию сетевого обмена, производство, управление персоналом и многое другое. При этом, несмотря на полноту функциональной части, бэк-офис должен быть простым в понимании и управлении, иметь высокую скорость развертывания на торговом объекте и обладать стабильностью и оперативностью обработки различной информации. Немаловажным критерием хорошего бэк-офисного решения является поддержка большого перечня торгового оборудования.

Несмотря на многообразие задач, функции бэк-офиса сводятся к нескольким важнейшим пунктам.

Прежде всего, бэк-офис должен разделять бизнес-процессы предприятия на несколько отдельных процессов, каждый из которых выполняется различными сотрудниками с разными правами доступа к информации. Это позволяет снизить требования к квалификации сотрудников и решает задачи обеспечения безопасности работы с информацией. Главный же плюс такого подхода – в возможности сделать систему максимально гибкой. Например, важнейшая функция бэк-офисной программы – управление розничными ценами. Это не такая простая задача, как может показаться. Цена товара, проданного отдельно, может отличаться от цены товара, проданного в мелкооптовой упаковке. Цена товара может зависеть и от того, какие другие товары покупаются вместе с ним. Наконец, цена может меняться при покупке товара по дисконтной карте и в ходе различных маркетинговых акций. Торговая система должна учитывать все эти моменты. Причём не просто учитывать, а позволять изменением цены оптимизировать прибыль. И это – тоже одна из целей разделения процессов.

Для разработки программного обеспечения, обеспечивающего решение задач пользователей бэк-офис систем, были поставлены следующие задачи:

- изучить рынок BOS;
- выбрать сегмент рынка, для разработки BOS;

- разработать BOS, которая будет удовлетворять основным нуждам клиентов в выбранной нише;
- изучить рынок POS и выбрать наиболее перспективную POS для интеграции.

Целью данной работы является разработка программного средства, представляющего собой веб-приложение. Для создания приложения, обеспечивающего управление каталогом товаров, складом и интеграцию с POS-системой необходимы серверная и клиентская часть приложения.

Клиентская часть приложения должна:

- обеспечивать кроссплатформенность;
- обладать удобным интерфейсом;
- предоставлять UI для управления сетью магазинов;
- быть легко поддерживаемым и масштабируемым.

Серверная часть приложения должна:

- обеспечивать хранение данных;
- основываться на REST-архитектуре;
- выполнять интеграцию с POS-системой;
- обеспечивать легкую поддержку и расширение.

Для выполнения всех вышеперечисленных условий был выбран следующий набор технологий:

Технологии для клиентской части приложения:

- TypeScript;
- Angular 6;
- Bootstrap 4;
- SCSS;
- Webpack.

Технологии для серверной части приложения:

- C#;
- SQL/T-SQL;
- CQRS (MediatR);
- FluentMigrator;
- Autofac;
- Odata v4;
- ASP.NET WEB API 2;
- ASP.NET Core;
- RabbitMQ;
- MSSQL;
- LiteDB;

Для решения задачи управления сетью магазинов, приложение должно обладать следующим функционалом:

- создание, редактирование, просмотр департаментов;
- сортировка, фильтрация департаментов;
- настройка ограничений департаментов;
- создание, редактирование, просмотр продуктов;
- сортировка, фильтрация, продуктов;
- расчёт стоимости, розничной цены и прибыли продукта;
- определение штрихового кода продукта;
- создание, редактирование, просмотр мультипаков продукта;
- создание, редактирование, просмотр мультиплайеров продукта;
- создание, редактирование, просмотр скидок;
- поиск, фильтрация скидок;
- создание, редактирование, просмотр групп продуктов
- поиск, фильтрация групп продуктов
- расчёт стоимости, розничной цены и прибыли группы продукта
- добавление, удаление продуктов из группы
- настройка правил изменения розничной цены группы продуктов в зависимости от зоны
- создание, редактирование, просмотр поставщиков
- создание, редактирование, просмотр инвойсов
- загрузка инвойса из файла
- возможность переключения между сайтами
- отправка департаментов в POS
- отправка скидок в POS
- отправка продуктов в POS
- создание, редактирование, просмотр кассиров

В ходе разработки получилось приложение, удовлетворяющие основные потребности пользователя в управлении сетью магазинов.

Заложенная архитектура позволяет эффективно расширять приложение, благодаря CQRS и компонентно-ориентированной архитектуре. Также кроссплатформенный сервис для интеграции с POS максимально упрощает внедрение продукта.

ЛИТЕРАТУРА

1. StackOverFlow [Электронный ресурс] / stackoverflow.com – Режим доступа: https://stackoverflow.com/search?q=* Дата доступа 14.04.2019.
2. GitHub.com [Электронный ресурс] / github.com. – Режим доступа: <http://github.com/jbogard/MediatR/wiki>. – Дата доступа 14.04.2019.

Студ. П. И. Карпович
Науч. рук. ст. преп. А. С. Наркевич
(кафедра программной инженерии, БГТУ)

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА СОЗДАНИЯ ОДНОТИПНЫХ РЕКЛАМНЫХ РОЛИКОВ

Видеореклама — один из наиболее эффективных и сильных по воздействию способов передачи информации о бренде, услуге, продукте, направленный на создание имиджа компании, продвижение услуг или товаров, предоставление информации с целью повышения продаж.

Отличительной чертой видеорекламы в интернете является диалогичность во взаимодействии между компанией, размещающей рекламу, и аудиторией, просмотревшей рекламный видеоролик. В связи с тем, что видеореклама зачастую размещается на популярных видеохостингах, таких как YouTube, Vimeo и др., аудитория получает возможность оставлять отзывы и формировать свой собственный рейтинг видео. Представители компании получают в свою очередь информацию о количестве просмотров записи, географическую статистику — место проживания зрителей, статистику просмотров по дням. Эти данные позволяют оценивать эффективность рекламной деятельности, давать прогнозы о конверсии. Также популярность видеорекламы в интернете обусловлена низкой ценой на размещение того или иного ролика.

Преимущества видеорекламы:

- возможность установления диалога между пользователем и рекламным роликом;
- наличие таргетинга для определения целевой аудитории;
- максимальный охват аудитории и привлечение потенциальных покупателей;
- низкая стоимость рекламных показов;
- высокая вовлекаемость и кликабельность.

Создание рекламных роликов – это наиболее частая, но, тем не менее, одна из самых сложных, ответственных и интересных задач для любой продакшн-студии. Ведь важно не просто снять рекламный ролик, а сделать так, чтобы он стал качественным, интересным и действительно эффективным инструментом коммуникации. Рекламные агентства тратят много времени в месяц на создание однотипных рекламных роликов.

Цель данной работы: разработка программного средства, представляющего собой веб-приложение, которое позволяет создавать однотипные рекламные ролики на основе шаблонов и параметров.

Были поставлены следующие задачи:

- изучить рынок систем автоматизации процесса создания однотипных видеороликов;
- веб-приложение должно позволять создавать видеоролик на основе параметров, которые задает пользователь;
- веб-приложение должно давать возможность просмотра созданных видеороликов;
- веб-приложение должно хранить видеоролики в облачном хранилище;
- веб-приложение должно хранить данные о шаблонах для видеороликов в базе данных;
- веб-приложение должно общаться с Adobe After Effect, установленном на сервере через API.

Для автоматизации процесса создания однотипных видеороликов на основе шаблона и параметров клиентская часть приложения должна:

- обеспечивать кроссплатформенность;
- обладать удобным интерфейсом;
- предоставлять UI для управления видеороликами;
- быть легко поддерживаемым и масштабируемым.
- серверная часть приложения должна:
- обеспечивать хранение данных;
- основываться на REST-архитектуре;
- выполнять интеграцию с Adobe After Effect;
- обеспечивать легкую поддержку и расширение.

Для выполнения всех вышеперечисленных условий был выбран следующий набор технологий:

Технологии для клиентской части приложения:

- JavaScript;
- Vue.JS;
- SCSS;
- Webpack.
- Babel

Технологии для серверной части приложения:

- JavaScript;
- Node.JS
- Express
- Lerna

- LowDb
- Fluent Ffmpeg
- Google Cloud Storage;
- RabbitMQ.

Для решения задачи управления сетью магазинов, приложение должно обладать следующим функционалом:

- веб-приложение должно позволять создавать видеоролик на основе параметров, которые задает пользователь;
- веб-приложение должно давать возможность просмотра созданных видеороликов;
- веб-приложение должно хранить видеоролики в облачном хранилище;
- веб-приложение должно хранить данные о шаблонах для видеороликов в базе данных;
- веб-приложение должно общаться с Adobe After Effect, установленном на сервере через API.
- создание и просмотр видеороликов;
- предоставлять выбор шаблона;
- предоставлять возможность заполнения параметров шаблона;

В ходе разработки получилось веб-приложение, удовлетворяющее основным потребностям пользователя в создании рекламных видеороликов.

Заложенная архитектура позволяет эффективно расширять приложение, благодаря микросервисной архитектуре.

ЛИТЕРАТУРА

1. StackOverFlow [Электронный ресурс]/ stackoverflow.com – Режим доступа: https://stackoverflow.com/search?q=* Дата доступа 14.04.2019.
2. StackOverFlow [Электронный ресурс]/ stackoverflow.com – Режим доступа: https://stackoverflow.com/search?q=* Дата доступа 14.04.2019.

Магистрант И. А. Литвинович
Науч. рук. ст. преп. А. С. Наркевич
(кафедра программной инженерии, БГТУ)

АЛГОРИТМЫ И ПОДХОДЫ К ОПРЕДЕЛЕНИЮ ПРИНАДЛЕЖНОСТИ ПРОФИЛЯ В СОЦИАЛЬНОЙ СЕТИ

Работа представляет собой анализ алгоритмов, используемых при написании магистерской диссертации, направленной на изучение и разработку алгоритмов поиска профилей в всемирно известных социальных сетях. Тема поиска профилей социальных сетей по фотографии является актуальной на сегодняшний день, так как социальные сети являются неотъемлемой частью жизни современного общества и одним из основных источников информации.

Пользователи социальных сетей выкладывают на своих страницах сотни фотографий, на которых фигурируют не только они сами, но и их друзья, знакомые, случайные люди, попавшие в кадр, животные и так далее. Для работы алгоритмов распознавания лиц, достаточно всего одной фотографии, например, аватарки. Но будет ли эта аватарка содержать фото профиля? Пользователи ставят на аватарки фотографии звёзд, а иногда профиль содержат только групповые фотографии. Таким образом, можно сделать вывод, что для определения принадлежности того или иного профиля конкретному человеку, встречающемуся на фотографиях этого профиля, необходимо разработать алгоритм и определить список наиболее важных признаков, по которым можно сделать однозначный вывод о принадлежности профиля.

Если рассматривать профили, которые состоят только из групповых фотографий, то можно определить владельца аккаунта, если учитывать его пол и возраст, а также друзей, профили которых были построены ранее.

Построение пользовательских профилей делится на следующие этапы.

– Выбор наиболее качественных фотографий пользователя. Если фотографий слишком много, необходимо использовать не более ста лучших. Качество фотографий определяется на основе:

- наличия отметок пользователей на фото (фотопинов) ручным способом;
- метаданных фотографии (фото загружено с мобильного телефона, снято на фронтальную камеру, в отпуске);
- фото было на аватарке.

- Поиск лиц на выбранных фотографиях.
- Вычисление характеристического вектора лица.

Кластеризация векторов. Задача этой кластеризации – определить, какой именно набор векторов принадлежит владельцу аккаунта. Основная проблема – это наличие друзей и родственников на фотографиях. Для кластеризации используется алгоритм DBScan.

Алгоритм DBSCAN был предложен Мартином Эстер, Гансом-Питером Кригель и коллегами в 1996 году как решение проблемы разбиения (изначально пространственных) данных на кластеры произвольной формы. Большинство алгоритмов, производящих плоское разбиение, создают кластеры по форме близкие к сферическим, так как минимизируют расстояние документов до центра кластера. Авторы DBSCAN экспериментально показали, что их алгоритм способен распознать кластеры различной формы.

Идея, положенная в основу алгоритма, заключается в том, что внутри каждого кластера наблюдается типичная плотность точек (объектов), которая заметно выше, чем плотность снаружи кластера, а также плотность в областях с шумом ниже плотности любого из кластеров. Ещё точнее, что для каждой точки кластера её соседство заданного радиуса должно содержать не менее некоторого числа точек, это число точек задаётся пороговым значением.

Определение лидирующего кластера.

Для каждого кластера необходимо рассчитать вес на основании:

- размера кластера;
- качества фотографий, по которым построены эмбединги в кластере;
- наличия фотопинов, привязанных к лицам из кластера;
- соответствия пола и возраста лиц в кластере с информацией из профиля;
- близость центроида кластера к профилям друзей, вычисленным ранее.

Коэффициенты параметров, участвующих в вычислении веса кластера, обучаются линейной регрессией.

Чтобы кластер считался лидером, нужно чтобы его вес был больше ближайшего конкурента на константу, рассчитанную на обучающей выборке. Если лидер не найден, необходимо еще раз перейти к пункту 2, но использовать большее число фотографий. Для некоторых пользователей необходимо сохранить два кластера. Такое бывает для совместных профилей — некоторые семьи имеют общий профиль.

Получение эмбедингов пользователя по его кластерам.

Построение вектора, который будет характеризовать внешность

владельца аккаунта, – «эмбединг пользователя».

Эмбединг пользователя – это центроид отобранного для него (лидирующего) кластера. Строить центроиды можно множеством разных способов. После многочисленных экспериментов было принято решение использовать самый простой из них: усреднение входящих в кластер векторов. Как и кластеров, эмбедингов у пользователя может быть несколько.

ЛИТЕРАТУРА

1. Handbook of Face Recognition - New York, Dover Publications, September 4, 2001, 272p
2. Reliable Face Recognition Methods: System Design, Implementation and Evaluation/ Harry Wechsler, Springer; 2007 edition (November 2, 2006), 329p
3. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие / Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В. — М.: МИЭМ, 2011. — 272 с.
4. Duda, R.O., Hart, P.E., Stork, D.G. Pattern classification. Second edition. New York: Wiley, 2000. 680 p.
5. Гонсалес Р., Вудс М. Цифровая обработка изображений. М.: Техносфера, 2012. 1104 с. Gonzales, R., Woods, R. Digital image processing. Third edition. New Jersey: Pearson Education Inc., Prentice Hall, 2008. 954 p

UDK 004.72

PhD student Ghassan Jaber
Supervise N. V. Patsei
(Department of software engineering, BSTU)

SCALABLE CONTENT ROUTING SYSTEM FOR ICN NETWORKS

Information-Centric networks (ICN) is a future architecture that aims to cache content objects in the network nodes or routers and allow the access to these contents from any location by ensuring in-network storage for caching contents, decoupling the content from the host address. The decoupling between publisher and subscriber removes the role of IP address, which works only as an identifier, locator and enabling multiparty communication through replication. Various projects like DONA, NDN, PURSUIT, SAIL, COMET, CONVERGENCE, MobilityFirst, PSIRP, CBCB, NetInf, KBN presented proposals in the ICN field in order to suit the new

requirements including the effective distributions of contents [1]. Three ICN architectures (i.e. PURSUIT, SAIL, and Mobility First) use name resolution approach for their data routing. However, another three architectures (i.e. CBCB, NDN, and CONET) use name-based data routing [1].

There are a number of challenges exists on the way of ICN architectures regarding the name resolution and data routing. The challenges include ensuring delivery of required content, detection of the nearest copy of required content, scalability, excessive current on routing tables, single point for failure, and finally, security and filtering. As a result, we propose a new routing scheme in this field - semantic routing scheme for ICN (SICN) [2].

To design the new network architecture (SICN), a closer look for the types of data communication that should be taken into consideration. In this manner, we classify data transmission into four types based on the number of subscriptions and frequency of use of the data object (table 1).

Another problem that resides in the proposed ICN schemes is the limitation to deal with knowledge searching.

Our ICN works on Publisher/Subscriber technique. The publisher sends message include structure naming, and the subscriber sends message with his interest. The publisher has broker that transforms the informal language into formal and construct the naming then send it to the network. The network, in its turn, then makes routing according to SQL query found in the naming. And where it finds positive match it makes as a channel (matching) between publisher message and the subscriber interest.

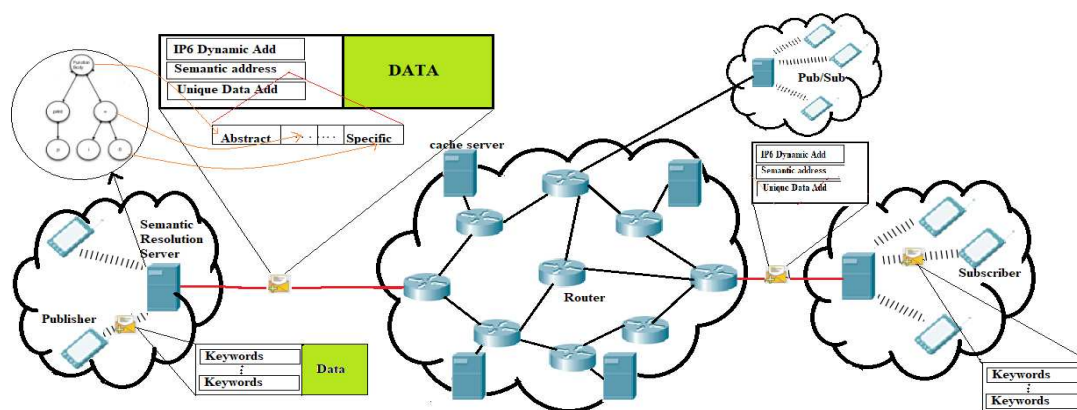
Table 1 – Data transmission classes

Type	Description	Example
R1	Requesting Any Data Content from specific Publisher	Voice call with specific user
R2	Requesting Specific Data Content from a specific Publisher	Accessing Cloud Storage
R3	Requesting Specific Data Content from Any Publisher	Downloading a song or a software installation file.
R4	Requesting Information with Any Data Content from Any Publisher	Searching for information using Google “How to make a kite”.

The naming scheme is consisted of the following: IPv6: for dynamic terminal mobility and fast routing; semantic tree based address (128 bit) - it represents the content keywords; unique data address is the Publisher ID

address; at least one address should be specified by the Subscriber his message (picture 1).

Routing Scheme: each router holds a routing table that match Semantic keywords to geographical location of the nearest publisher or cache; each Record can be hold for a specific Time to Live (TTL); TTL is directly proportional to the number of subscribers and the frequency of Request (picture 1).



Picture 1 - SICN system architecture

In this way, our proposed naming scheme is based on three dimensions knowing that the user (publisher/subscriber) should label the data with least one dimension. As part of the work has been developed routing table structure and the caching policy used by SICN scheme.

REFERENCES

1. G. Jaber, N. V. Patsei, Information Centric Networking for web-based content distribution and manipulation// Proceeding of BSTU. – Issue3, No2(200) 2017. – P. 88-91
2. G Jaber, N. Patsei, Routing and caching survey of information-centric networking// материалы Междунар. науч.-техн. конф. «Автоматический контроль и автоматизация производственных процессов», Минск, 3-6 октября 2018– Минск: БГТУ, 2018. С166-169.

УДК 004.42

Студ. П. В. Бернацкий
Науч. рук. ст. препод. А. С. Наркевич
(кафедра программной инженерии, БГТУ)

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ «ФОРУМ ДЛЯ РАЗРАБОТЧИКОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ» С ИСПОЛЬЗОВАНИЕМ JAVA EE

Цель работы: разработать веб-приложение форум.

Веб-приложение должно иметь следующий функционал:

- просмотр существующих, а также добавление новых тем, а также постов к ним;
- создание учетной записи пользователя и ее изменение;
- блокировка содержимого и участников форума модераторами;
- отправка частных сообщений между пользователями, а также ведение списка друзей и заблокированных пользователей;
- размещение и просмотр медиа файлов таких как изображения, видео и презентации в темах;
- добавление примеров исходного кода и их последующий запуск для выполнения;
- фильтрация создаваемых постов и тем на наличие не подходящего содержимого.

Разрабатываемое веб-приложение работает на двухзвенной архитектуре клиент серверного взаимодействия. На рисунке 1 приведена данная схема.

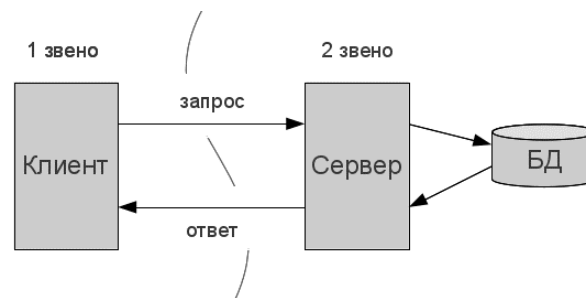


Рисунок 1 – Двухзвенная архитектура клиент серверного взаимодействия

Так как клиентом в разрабатываемом веб-приложении является браузер, то основной задачей является разработка сервера и базы данных.

В качестве платформы для разработки сервера была выбрана Java EnterpriseEdition 8. В качестве архитектуры сервера была выбрана архитектура Model-View-Controller. В качестве MVC фреймворка была выбрана технология JSF. В отличие от прочих MVC-фреймворков,

которые управляются запросами, подход JSF основывается на использовании компонентов. Состояние компонентов пользовательского интерфейса сохраняется, когда пользователь запрашивает новую страницу и затем восстанавливается, если запрос повторяется. На рисунке 2 приведена схема работы приложения, построенного на основе JSF.

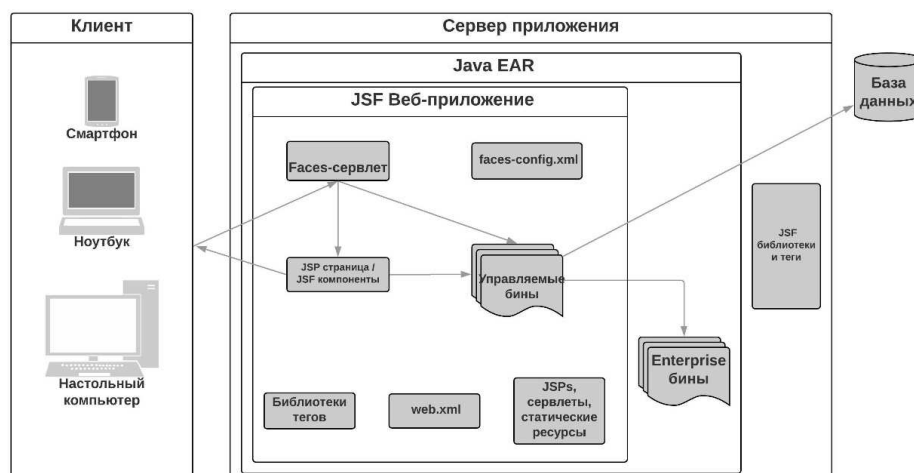


Рисунок 2 – Архитектура JSF приложения

JSF обеспечивает простоту использования благодаря целому ряду факторов:

- упрощает формирование пользовательского интерфейса из набора повторно используемых компонентов пользовательского интерфейса;
- упрощает перенос данных приложения в пользовательский интерфейс и из него;
- помогает управлять состоянием пользовательского интерфейса при запросах к серверу;
- предоставляет простую модель установления связи между созданными клиентом событиями и кодом приложения на стороне сервера;
- упрощает сборку и повторное использование компонентов пользовательского интерфейса.

В разрабатываемом веб-приложении используются такие библиотеки готовых JSF компонентов, как:

- PrimeFaces 7.0;
- OmniFaces 3.2;
- Apache MyFaces 2.3.1;
- RichFaces 4.5.17.

В качестве фреймворка для работы с базой данных на сервере выбран EclipseLink. EclipseLink — это свободный фреймворк для язы-

ка программирования Java, предназначенный для решения задач объектно-реляционного отображения ORM, разрабатываемый Eclipse Foundation.

Для сборки веб-приложения и его зависимостей использовался Apache Maven. Maven – это инструмент для сборки Java проекта: компиляции, создания jar, создания дистрибутива программы, генерации документации. Так как для разработки использовалась платформа JavaEE, для запуска приложения требуется сервер приложений либо контейнер сервлетов. Для данного веб-приложения был выбран сервер приложений WildFly. WildFly – JavaEE-сервер приложений с открытым исходным кодом, разработанный компанией JBoss и поддерживаемый компанией RedHat. В качестве базы данных для хранения информации веб-приложения была выбрана PostgreSQL. PostgreSQL – свободная объектно-реляционная система управления базами данных, разрабатываемая сообществом PostgreSQL.

Логическая структура разработанной базы данных для данного веб-приложения представлена на рисунке 3.

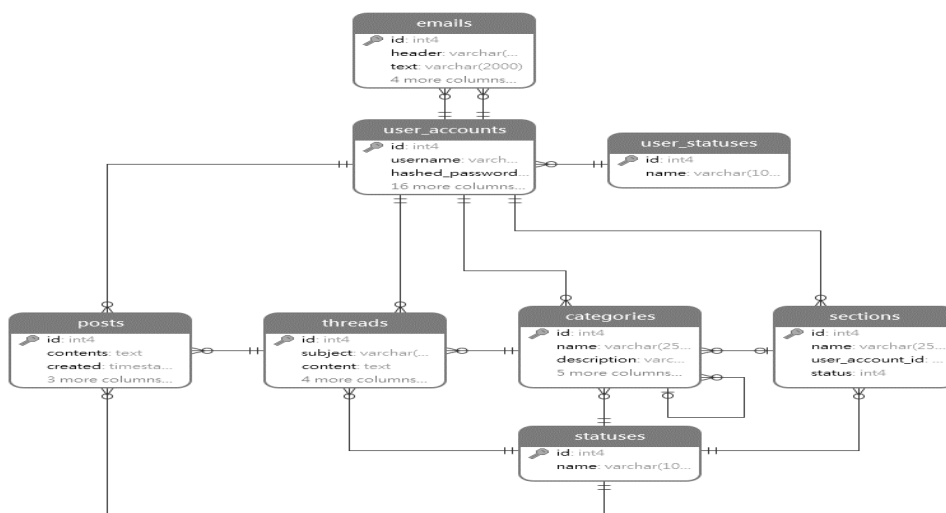


Рисунок 3 – Логическая структура базы данных

Разработанное веб-приложение поможет в поиске и распространении информации для разработчиков программного обеспечения. На форуме разработчики смогут создавать темы самостоятельно для поиска недостающей информации или же изучить уже готовый контент.

ЛИТЕРАТУРА

1. Шилдт Г. Искусство программирования на Java / Г. Шилдт – 2005 г., 336 с.
2. Дэвид М. Гери, Кей С. Хорстманн. JavaServerFaces. Библиотека профессионала 3-е издание / Дэвид М. Гери, Кей С. Хорстманн – 2011 г., 544 с.

УДК 004.414+004.738:347.453.3

Студ. Н. И. Мелентьев

Науч. рук. доц., канд. техн. наук А. И. Парамонов
(кафедра программной инженерии, БГТУ)

ВЕБ-ОРИЕНТИРОВАННАЯ СИСТЕМА ПОДДЕРЖКИ РАБОТЫ РИЭЛТЕРСКОЙ КОМПАНИИ

Во все времена недвижимость имела важную роль в жизни человека и продолжает является её неотъемлемой частью. Рынком жилой недвижимости интересуются люди в любом возрасте. Кто-то планирует приобретение квартиры, кто-то ищет нового жильца для пустующей комнаты, кто-то желает продать или приобрести квартиру. Все эти люди ищут способы для того, чтобы донести свои предложения другим людям [1]. Для этого многие пользуются различными веб-сайтами на тематику недвижимости или просто публикуют объявления в средствах массовой информации. Интернет-ресурсы, которые посвящены тематике недвижимости, представлены в большом количестве и в любой стране, где имеется рынок недвижимости. Таким образом, потенциальный покупатель или продавец может получить доступ к объявлениям о продаже жилой недвижимости или ее покупке в любой точке мира, где имеется доступ к сети интернет.

Идея проекта заключается в том, чтобы объединить данные о продаже или покупке жилой недвижимости с различных интернет ресурсов одном месте. Таким образом, для того, чтобы сориентироваться в большом количестве предложений по жилищному рынку и ускорить процесс покупки-продажи подходящего объекта на коммерческом рынке, потенциальным клиентам целесообразно обращаться к услугам посредников, которыми выступают риэлтерские агентства [2].

Основная цель агентства – сбор данных о рынке недвижимости, оказание услуг по купле-продаже и аренде недвижимости, создание базы данных с целью предоставления клиентам информационных и консультационных услуг.

Основные этапы работы сотрудников агентства недвижимости:

- формирование базы объектов недвижимости – сбор информации о продаваемых и сдаваемых в аренду объектах недвижимости;
- заявка, консультация. На первом этапе клиент звонит в агентство, предварительно прочитав о спектре риэлтерских услуг, либо о предложенном варианте из какого-либо информационного источника;
- подбор вариантов. После того, как потенциальный клиент принял решение воспользоваться услугами агентства, происходит

подбор вариантов. Риелторы предлагают существующие варианты из своей базы данных;

- выбор из предложенных объектов. Агенты по недвижимости совместно с клиентом рассматривают выбранные варианты, при необходимости осуществляется показ каждого объекта недвижимости.

В организации всех этих этапов и состоит цель предлагаемой веб-ориентированной системы. Подобная информационная система необходима для формирования и поддержки внутренней базы курируемых «помещений» агентства, в том числе – ввода и редактирования архитектурных особенностей квартир, характеристики планировки, указание мест их расположения, учет данных владельцев и прочее.

При разработке данной системы будут использованы SAP технологии. Система SAP — это программное обеспечение, при помощи которого можно автоматизировать профессиональную деятельность представителей разных специализаций. SAP разработали не только сами оболочки, но и особую методологию внедрения ПО, которая существенно облегчает переходные этапы и позволяет внедрить комплекс без простоев в работе и длительной адаптации [3]. Также это полезный и нужный инструментарий, который существенно облегчает взаимодействие между управляемыми и управляющими структурами в любом бизнесе, особенно крупном и сложно организованном. С их помощью можно обмениваться сведениями с партнерами, интегрировать новые решения с уже имеющимися.

Для разработки нашей информационной системы будут использованы: HANA, ABAP cds, SAP Gateway и фреймворк SAP UI5.

HANA — это высокопроизводительная NewSQL платформа для хранения и обработки данных, в основе которой лежит технология вычислений in-memory с использованием принципа поколоночного хранения данных, платформы, разработанной и выведенной на рынок компанией SAP SE. Архитектура HANA обеспечивает как высокоскоростную обработку транзакций, так и работу со сложными аналитическими запросами, совмещая решение этих задач в рамках единой платформы. Кроме самой СУБД, в составе HANA предусмотрены встроенный веб-сервер и репозиторий управления версиями, необходимые для разработки приложений. Приложения HANA могут создаваться с использованием кода JavaScript на стороне сервера и HTML-кода.

SAP Gateway – это открытая основанная на стандартах структура, которую разработчики могут использовать для более простого подключения non-SAP-приложений к приложениям SAP. Он также используется для подключения к приложениям SAP и доступа к ним с

мобильных устройств. В результате он является главным средством для разработки некоторых из новейших и наиболее известных технологий SAP, включая мобильную платформу SAP и SAP Fiori. SAP Gateway позволяет создать сервис Open Data Protocol (OData), который позволяет получать данные из системы.

Для создания UI составляющей приложения будет использоваться SAP Web IDE и фреймворк SAP UI5.

SAP Web IDE — облачная система разработки от SAP, она представляет собой инструмент для быстрого создания новых приложений SAP UI5[4].

SAP UI5 — это фреймворк для разработки пользовательских интерфейсов на базе кросс-браузерной JavaScript библиотеки. Также это многофункциональный инструмент для создания современных пользовательских интерфейсов бизнес-приложений как на базе стандартных компонентов, так и на базе пользовательских.[5]

Результатом выполнения данной работы является информационная система, с помощью которой можно собирать информацию о жилой недвижимости (квартиры и комнаты) и предоставлять эти данные в удобной для восприятия пользователя форме. Работник агентства сможет получить доступ к системе как через стационарный компьютер, так и с помощью портативного гаджета (телефона или планшета), так как данная система имеет адаптивный дизайн, который дает возможность работы с различных устройств. Благодаря веб-ориентированной архитектуре система доступна везде, где есть выход в сеть Интернет.

ЛИТЕРАТУРА

1. Рынок недвижимости и ипотека: учебно-методический комплекс / Н. А. Устюшенко; ГИУСТ БГУ, Каф. управления недвижимостью. – Минск: ГИУСТ БГУ, 2014. – 170 с.
2. Рынок недвижимости как составляющая конкурентоспособности экономики: сб. науч. ст. / ГИУСТ БГУ, Каф. управления недвижимостью; редкол.: В. П. Киреенко (отв. ред.) [и др.]. – Минск: ГИУСТ БГУ, 2016. – 97 с.
3. ABAP platform [Электронный ресурс] – <https://developers.sap.com/topics/abap-platform.html>
4. UI Development [Электронный ресурс] – <https://developers.sap.com/topics/ui-development.html>
5. SAPUI5 custom libraries: Deploy to ABAP Repository and use in applications [Электронный ресурс] – <https://blogs.sap.com/2019/04/05/sapui5-custom-libraries-deploy-to-abap-repository-and-use-in-applications/>

АНАЛИЗ АЛГОРИТМОВ СЖАТИЯ ИНФОРМАЦИИ

Сжатие данных - это процесс, который используется для уменьшения физического размера блока информации. Задача сжатия состоит из двух компонентов: алгоритма кодирования, который принимает сообщение и генерирует сжатое представление (возможно, с меньшим количеством битов), и алгоритма декодирования, который восстанавливает исходное сообщение или некоторую его аппроксимацию из сжатого представления. Алгоритмы сжатия делятся на два класса – алгоритмы сжатия без потерь и алгоритмы с потерями. В данной работе были рассмотрены три часто используемых метода: кодирование Хаффмана, Лемпеля-Зива и RunLength.

Целью исследования является сравнение методов сжатия данных без потерь, чтобы установить следующее:

- степень (коэффициент) сжатия, которая может быть достигнута с помощью каждого из алгоритмов;
- уровень эффективности различных алгоритмов;
- сравнение методов по типу данных.

Метод кодирования Хаффмана назначает более короткие коды символам, которые встречаются чаще, а более длинные коды - тем, которые встречаются реже. Прежде чем мы сможем назначить битовые комбинации для каждого символа, мы назначаем каждому символу вес в зависимости от частоты его использования. Кодирование Хаффмана использует специальный метод выбора представления для каждого символа, в результате чего получается код префикса (иногда называемый «кодами без префикса»), то есть строка битов, представляющая некоторый конкретный символ. Для набора символов с равномерным распределением вероятностей и числом членов, которое является степенью двойки, кодирование Хаффмана эквивалентно простому двоичному блочному кодированию, например, кодированию ASCII.

Техника сжатия Run-Length, вероятно, является наиболее простым методом сжатия. Он может использоваться для сжатия данных, составленных из любой комбинации символов, и может быть достаточно эффективным, если данные представлены в виде 0 и 1. Общая идея этого метода состоит в том, чтобы заменить последовательные повторяющиеся вхождения символа одним вхождением символа, за которым следует количество вхождений.

Метод сжатия Lempel-Ziv (LZ) является одним из самых популярных алгоритмов без потерь. Код, который выводит алгоритм LZ,

может иметь любую произвольную длину, но должен содержать больше одного символа. Первые 256 кодов (при использовании восьми битных символов) по умолчанию назначаются стандартному набору символов. Остальные коды присваиваются строкам в процессе работы алгоритма.

LZ-кодирование относится к категории алгоритмов, называемых словарным кодированием. Идея состоит в том, чтобы создать словарь (таблицу) строк, используемых во время сеанса связи. Если и отправитель, и получатель имеют копию словаря, то ранее встреченные строки можно заменить их индексом в словаре, чтобы уменьшить объем передаваемой информации. Есть два одновременных события: создание индексированного словаря и сжатие строки символов. Алгоритм извлекает наименьшую подстроку, которую невозможно найти в словаре, из оставшейся несжатой строки. Затем он сохраняет копию этой подстроки в словаре в качестве новой записи и присваивает ей значение индекса. Сжатие происходит, когда подстрока, за исключением последнего символа, заменяется индексом, найденным в словаре. Программа для архивирования/разархивирования была разработана с использованием delphi. Данные, которые использовались для тестирования трех (3) сравниваемых методов сжатия: текстовый документ, документ Microsoft Word, веб-документ и изображение.

После запуска тестов были получены данные по коэффициентам сжатия, представленные на рисунке 1.

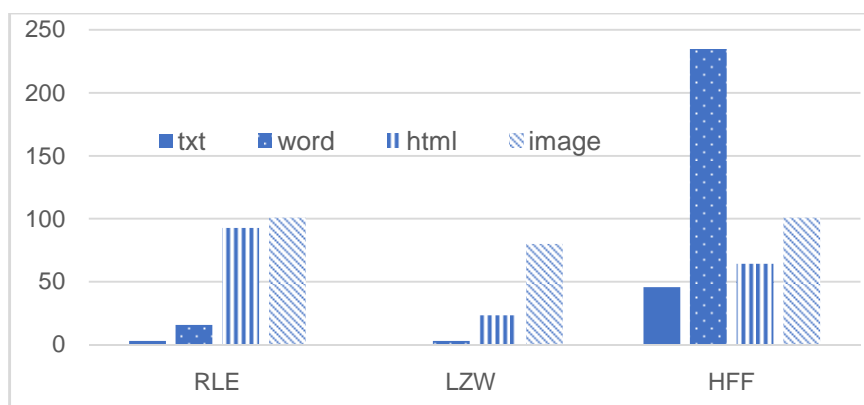


Рисунок 1 - Исследование коэффициента сжатия

Как видно из диаграммы для сжатия тестовых документов лучше подходит RLE алгоритм. Для сжатия документов в формате doc, а также html наименьший коэффициент сжатия дает алгоритм LZW. Для сжатия изображений исследуемые алгоритмы мало эффективны, так минимальный коэффициент сжатия -80% был получен при использовании алгоритма LZW.

Таким образом, метод Lempel-Ziv оказался наиболее эффективным и дает хорошие результаты для текстовых и веб-документов.

Студ. И. В. Гордеев
Науч. рук. ассист. О. Л. Панченко
(кафедра программной инженерии, БГТУ)

СОЦИАЛЬНАЯ ПЛАТФОРМА ДЛЯ ПРОДУКТОВ

В реалиях современного мира, нас постоянно окружает огромное количество магазинных сетей, в каждой из которых можно найти огромное множество неизвестных продуктов. И перед обычным покупателем каждый раз появляется проблема, когда он видит незнакомый для него продукт на прилавке. Существует большое количество сервисов, которые предоставляют отзывы на различные товары. Но большая проблема данных сервисов состоит в том, что они не конкретизируются на определённом типе товаров, часто имеют много фальшивых отзывов, а также обладают ограниченным ассортиментом среди продуктов. Что ещё не позволяют данные сервисы узнать, так это магазины, в которых данные продукты можно найти. Из-за этого обычный покупатель сталкивается с огромным количеством трудностей, когда пытается узнать какую-то информацию о продукте, либо же пытается узнать, где он может приобрести данный продукт.

Социальная платформа для продуктов должна предоставлять перечисленные выше функции в одном месте. Отличительная особенность социальной платформы в том, что в ней присутствует часть функционала социальных сетей. Существуют различные типы социальных сетей, но в текущем контексте нас интересуют социальные сети для отзывов и обзоров. В пример можно привести такие сети как Yelp, Urbanspoon. В основе социальных сетей Yelp и Urbanspoon лежит определение геолокации и возможность оставлять комментарии и рекомендации о локальном бизнесе. Такие социальные сети - это огромная база знаний, которая помогает пользователям собрать всю необходимую информацию для принятия решения о покупке.

Несмотря на довольно большое количество схожих платформ, на данный момент нет аналогичной платформы, которая ориентируется исключительно на продукты питания.

Поэтому главные критерии для данного приложения – это реализация честной системы отзывов, возможность добавления нового продукта на сайт, если до этого никто ещё не добавил его, нахождение продукта по магазинам. А также, что не менее важно – реализации социальной части сайта. В контексте данного приложения будет доступна система друзей, система оценки как самого продукта, так и отзывов других пользователей. Также будет доступна лента новостей

друзей для отслеживания их активности (добавление/редактирование отзывов, добавление нового продукта).

Для разработки программного обеспечения, обеспечивающего решение задач пользователей, были поставлены следующие задачи:

- изучить рынок аналогов;
- выбрать сегмент рынка, для разработки приложения;
- разработать приложение, которое будет удовлетворять основным нуждам клиентов в выбранной нише;

Целью данной работы является разработка программного средства, представляющего собой веб-приложение. Для создания приложения, обеспечивающего оценку и нахождение информации о продуктах необходимы серверная и клиентская часть приложения.

Клиентская часть приложения должна:

- обеспечивать кроссплатформенность;
- обладать удобным интерфейсом;
- предоставлять UI для управления продуктами и отзывами;
- быть легко поддерживаемым и масштабируемым.

Серверная часть приложения должна:

- обеспечивать хранение данных;
- основываться на REST-архитектуре;
- обеспечивать легкую поддержку и расширение.

Для выполнения всех вышеперечисленных условий был выбран следующий набор технологий:

Технологии для клиентской части приложения:

- React;
- Redux;
- SCSS;
- Webpack.

Технологии для серверной части приложения:

- C#;
- SQL;
- EF Core Migrator;
- ASP.NET WEB API 2;
- ASP.NET Core;
- MySQL;

Для решения задачи управления сетью магазинов, приложение должно обладать следующим функционалом:

- регистрация и логин пользователей;
- создание, редактирование, удаление и просмотр статей;
- глобальный поиск по сайту;

- создание, редактирование, просмотр, удаление продуктов;
- сортировка, фильтрация, продуктов;
- добавление, удаление, просмотр продуктов в избранное;
- добавление, удаление, просмотр, фильтрация, сортировка продуктов в списке покупок;
- добавление, удаление, просмотр отзывов на продукты;
- рейтинговая система продуктов;
- история действий пользователя;
- настройка профиля пользователя;
- система оповещений пользователя;
- просмотр всех пользователей;
- блокировка пользователей;
- система друзей;
- лента новостей друзей

В ходе разработки получилось приложение, удовлетворяющие основные потребности пользователя в создании и получении информации о продуктах.

Заложенная архитектура позволяет эффективно расширять приложение, благодаря сервисному и компонентно-ориентированному подходу. Также кроссплатформенность продукта упрощает его интеграцию с различными системами.

ЛИТЕРАТУРА

1. StackOverFlow [Электронный ресурс] / stackoverflow.com – Режим доступа: https://stackoverflow.com/search?q=* Дата доступа 14.04.2019.

2. GitHub.com [Электронный ресурс] / github.com. – Режим доступа: <http://github.com/jbogard/MediatR/wiki>. – Дата доступа 14.04.2019.

УДК 004.042

Студ. Е. Ю. Курлович
Науч. рук. доцент А. А. Дятко
(кафедра программной инженерии, БГТУ)

ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ СОВЕРШЕНСТВОВАНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА В УНИВЕРСИТЕТАХ

Каждый из нас знает, что представляют собой студенческие годы. Особенно сложна была жизнь многих студентов во время зачётной недели и экзаменационной сессии: часть студентов делает упор в течение семестра только на некоторые профильные дисциплины, а

кто-то и вовсе не открывал ни одного учебного пособия или же лекции.

На просторах Интернета есть немало информации по тому или иному вопросу, который вы вводите в поисковую строку. Однако, не всегда эта информация бывает достоверной, что порой вызывает трудности в процессе подготовки к сдаче зачёта или экзамена.

Целью представленной работы является:

1. Желание помочь студентам и преподавателям в оптимизации процесса подготовки к зачётам и сессии.
2. Накопление полученного опыта и распространение достоверной информации в рамках университета.

Для достижения цели, были решены следующие задачи:

1. Спроектирована и разработана база данных для хранения информации об университетах, факультетах, кафедрах, преподавателях и т.д.
2. Спроектирована архитектура информационной системы.
3. Реализован интерфейс для администратора, который позволит управлять информацией об университетах, факультетах, учётных записях пользователей и т.д.
4. Создан пользовательский интерфейс: веб-приложение, которое взаимодействует с сервером, написанным с использованием технологии ASP.NETCORE.

Информационная система имеет следующий функционал:

1. Подключение к серверу и предоставление возможности пройти авторизацию, регистрацию.
2. Заполнение информации о преподавателях университетов.
3. Создание, улучшение ответов на экзаменационные вопросы.
4. Добавление новых университетов, факультетов, специальностей.
5. Управление привилегиями и аккаунтами пользователей администратором.
6. Размещение лекций, конспектов, презентаций, справочной информации преподавателями.
7. Управление личным кабинетом пользователя.

В качестве СУБД был использован MSSQLServer. Доступ к базе данных из веб-приложения осуществляется с помощью Entity-Framework. Для разработки клиентской части используется Bootstrap 4, HTML5. В качестве системы авторизации и аутентификации используется стандартная ASP.NETCOREIdentity.

Основной технологией приложения является ASP.NET CORE MVC. Схема работы паттерна MVC предоставлена на рисунке 1.

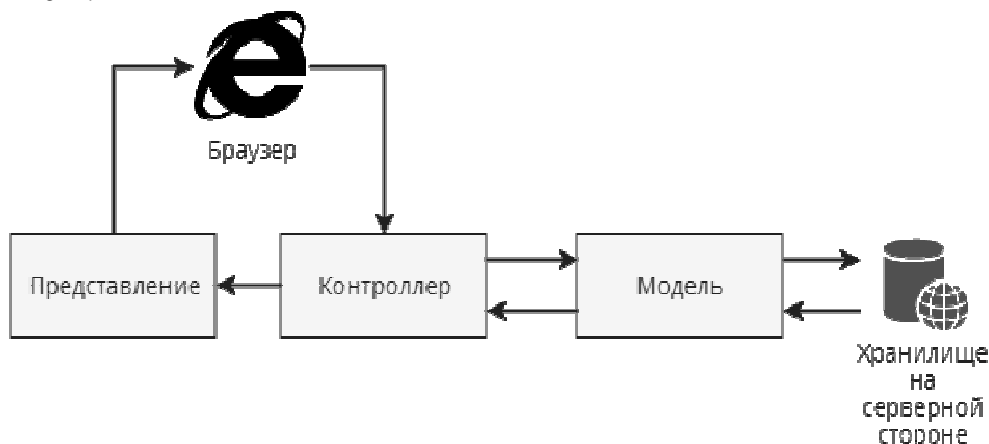


Рисунок 1 – Схема работы паттерна MVC

Данное веб-приложение может найти практическое применение среди студентов, обучающихся в рамках одного университета. Использование программы позволит оптимизировать образовательный процесс в университете во время экзаменационной сессии.

ЛИТЕРАТУРА

1. Сайт о программировании Metanit. [Электронный ресурс] / www.metanit.com / – Режим доступа: <https://metanit.com/>– Дата доступа: 12.04.2019

УДК 004.056

Студ. А. Д. Сеница

Науч. рук. ассист. Н. А. Северинчик
(Кафедра программной инженерии, БГТУ)

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОЦЕНКИ НЕДВИЖИМОСТИ

Ипотечный кредит — одна из форм залога, при которой закладываемое недвижимое имущество остаётся во владении и пользовании должника, а кредитор, в случае невыполнения должником своего обязательства, приобретает право получить удовлетворение за счёт реализации данного имущества, которое поступает в залог банку как гарантия возврата кредита.

Таким образом, становится ясно, что банкам и ипотечным агентствам особо важно понимать реальную рыночную стоимость недвижимости. В связи с этим возникла необходимость производить качественную оценку недвижимости разного типа. Специалисты лично

производят оценку зданий, и составление отчетов во время таких «оценок», может занять огромное количество времени.

Чтобы оптимизировать данный процесс, был реализован сервис, упрощающий оценку недвижимости. Приложение позволяет создавать отчеты, на основании общедоступных данных, которые находятся в государственном реестре зданий. Другой особенностью является заполнение специфичной информации, которая влияет на конечную стоимость данного объекта. Ключевой особенностью является возможность сравнения объекта с другими объектами на основе различных показателей, а также возможность составления карт со схожими объектами.

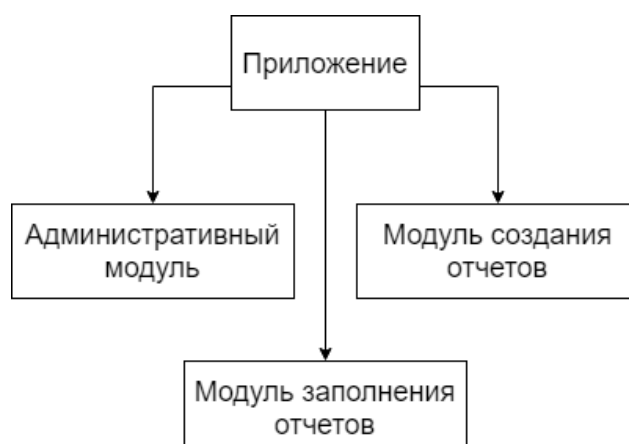


Рисунок 1 – Структура приложения

На рисунке 2 представлен пользовательский интерфейс по поиску объектов для сравнения на карте. Данный поиск также поддерживает фильтрацию объектов и сортировку.

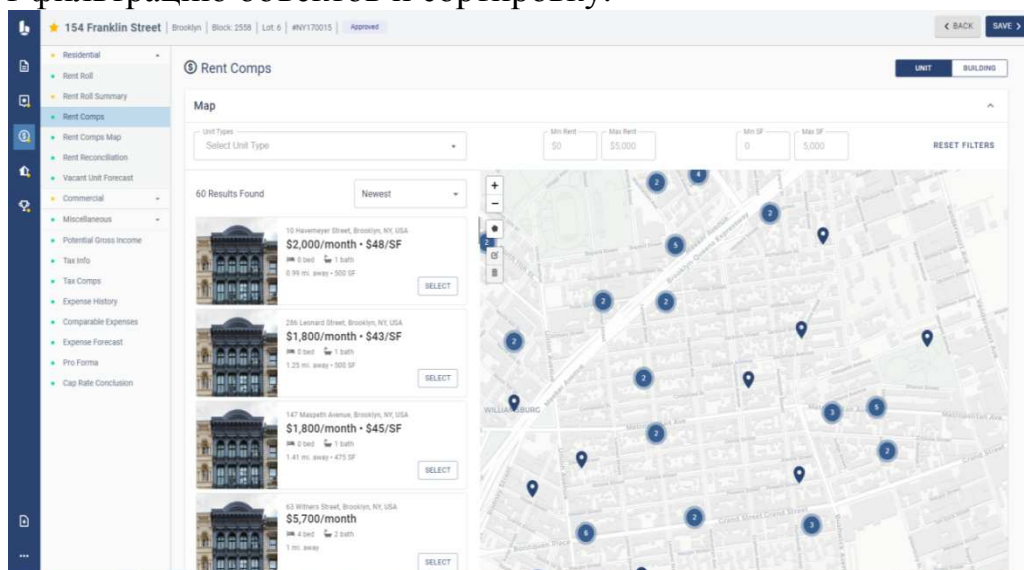


Рисунок 2 – Страница поиска объектов для сравнения

Данная автоматизированная система оценки недвижимости, имеет следующие преимущества:

– **увеличивается скорость выполнения повторяющихся задач.** За счет автоматического режима одни и те же задачи могут выполняться быстрее, т.к. автоматизированные системы более точны в действиях и не подвержены снижению работоспособности от времени работы;

– **повышается точность управления.** За счет применения информационных технологий в автоматизированных системах появляется возможность сохранять и учитывать большее количество данных о процессе, чем при ручном управлении;

– **быстрое принятие решений в типовых ситуациях.** В автоматизированных системах решения, связанные с типовыми ситуациями, принимаются гораздо быстрее, чем при ручном управлении.

На основании открытых данных составлена диаграмма эффективности использования данного приложения, которая отражает увеличение количества создаваемых отчетов в месяц на одного специалиста.

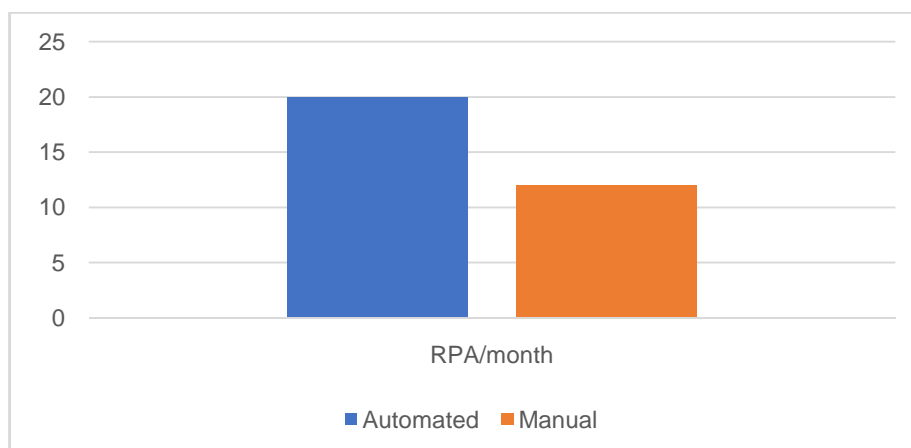


Рисунок 3 – Диаграмма эффективности приложения

Исходя из описанных выше преимуществ, можно заключить, что данный продукт получит широкое применение. Так как он является инновационным инструментом автоматизации в области ипотечных бизнес-процессов.

ЛИТЕРАТУРА

1. McElroy, K. The ABC's of property management: What you need to know to maximize your money now. – New York: Business plus, 2008. – 147 p.
2. Alex, B. How to Deliver Client/Server Applications That Work. – New York: Prentice Hall, 1995. – 200p.

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ОЦЕНКИ ЭМОЦИОНАЛЬНОЙ ТОНАЛЬНОСТИ ОТЗЫВОВ НА ФИЛЬМЫ

При принятии того или иного решения человек часто обращается к советам, рекомендациям и мнениям других людей. Если еще несколько десятков лет назад люди обращались за традиционным советом к своим знакомым, то сейчас все изменилось. С бурным развитием информационно-вычислительных технологий и глобальной сети Интернет, процесс получения рекомендаций значительно упростился. Но вместе с увеличением числа пользователей сети Интернет, возрастает и количество генерируемого ими контента. Объемы информации в Интернете на сегодняшний день так велики, что необходимые данные попросту теряются в море бесполезных данных.

Исходя из вышесказанного, в настоящее время чрезвычайно востребованной является задача автоматического анализа мнений пользователей о товарах или услугах.

Тональность текста – это эмоциональная оценка, выраженная в тексте. Анализ тональности текста (или сентимент-анализ) – это область компьютерной лингвистики, которая занимается извлечением из текста эмоционально окрашенной лексики и получением представления об эмоциональном отношении авторов к объектам, о которых идёт речь в тексте, что влияет на формирование определённого образа у реципиентов [1].

Задача анализа тональности текста возникла сравнительно недавно, поэтому исследования в этой области проводятся в настоящее время, и на данный момент все еще актуальна задача построения системы извлечения мнений и анализа тональностей.

Целью данной работы является создание веб-приложения, при помощи которого можно отыскать интересующий фильм, после просмотреть его детальную информацию и отзывы на него, а также оценку эмоциональной тональности для каждого из отзывов, вычисленную с помощью обученного байесовского классификатора или веб-сервиса *IBM Watson Natural Language Understanding*.

Рассмотрим подробнее оценку эмоциональной тональности с помощью наивного Байесовского классификатора.

Байесовский классификатор – один из самых часто используемых классификаторов, из-за сравнительной простоты в имплементации и тестировании. В то же время, наивный Байесовский классифи-

катор демонстрирует не худшие результаты по сравнению с другими, более сложными классификаторами.

В основе наивного Байесовского классификатора лежит теорема (или формула) Байеса [2].

Основная идея байесовской стратегии заключается в оценке вероятности некоторого вывода с учетом фактов, подтверждающих или опровергающих этот вывод. Формулировка теоремы Байеса, известная из теории вероятностей, следующая.

Пусть имеется n несовместных событий H_1, H_2, \dots, H_n . Известны вероятности этих событий: $P(H_1), P(H_2), \dots, P(H_n)$, причем $P(H_1) + P(H_2) + \dots + P(H_n) = 1$. Это означает, что события H_1, H_2, \dots, H_n образуют полную группу событий, т.е. одно из них происходит обязательно. С событиями H_1, H_2, \dots, H_n связано некоторое событие E . Известны вероятности события E при условии того, что какое-либо из событий H_1, H_2, \dots, H_n произошло: $P(E/H_1), P(E/H_2), \dots, P(E/H_n)$. Пусть событие E произошло. Тогда вероятность того, что при этом какое-либо из событий H_i ($i=1, \dots, n$) произошло, можно найти по следующей формуле (формула Байеса):

$$P(H_i/E) = \frac{P(E/H_i)P(H_i)}{P(E/H_1)P(H_1) + P(E/H_2)P(H_2) + \dots + P(E/H_n)P(H_n)} = \frac{P(EH_i)}{P(E)}. \quad (1)$$

События H_1, H_2, \dots, H_n называются гипотезами, а событие E – свидетельством. Вероятности гипотез $P(H_i)$ без учета свидетельства (т.е. без учета того, произошло событие E или нет) называются доопытными (априорными), а вероятности $P(H_i/E)$ – послеопытными (апостериорными). Величина $P(EH_i)$ – совместная вероятность событий E и H_i , т.е. вероятность того, что произойдут оба события вместе. Величина $P(E)$ – полная (безусловная) вероятность события E .

Пусть необходимо определить принадлежность отзыва к позитивному отзыву D_1 либо негативному отзыву D_2 . Для этих двух типов отзывов известны условные вероятности $P(S|D_j)$, характеризующие появление набора слов S в отзыве D_j . Здесь $S = \{S_1, S_2, \dots, S_k\}$, где S_k – различные слова. Пусть далее $P(D_j)$ – априорная вероятность отзыва D_j . Тогда задача определения, является ли отзыв положительным, может быть сведена к статистической задаче выбора гипотез, решение которой основывается на использовании теоремы Байеса:

$$P(D_j|S) = \frac{P(D_j) \cdot P(S|D_j)}{\sum_{j=1}^n P(D_j) \cdot P(S|D_j)}, \quad (j = 1, \dots, n). \quad (2)$$

Если для какого-нибудь сообщения D_j рассчитанная вероятность принадлежности отзыва к положительному по формуле (2) значитель-

но превосходит значение вероятности принадлежности отзыва к негативному, то оптимальное правило приписывает это сообщение к позитивному.

Использование формулы (2) встречается со следующими трудностями: условные вероятности $P(S | D_j)$ можно сравнительно легко вычислить лишь при небольшом числе признаков S_1, S_2, \dots, S_k . Во-первых, это может быть связано с тем, что число различных комбинаций признаков может быть очень большим. Например, если каждое слово имеет только два различных значения (позитивное/негативное), а число слов равно 30, то число всех возможных вариантов может исчисляться величиной 2^{30} . Поэтому будем считать слова в отзыве независимыми друг от друга.

В веб-приложении для создания и обучения модели Байесовского классификатора использовался фреймворк для машинного обучения *Accord.NET*.

Вторым способом оценки эмоциональной тональности отзывов является использование веб-сервиса *IBM Watson Natural Language Understanding*.

IBM Watson Natural Language Understanding – это набор *API*, которые предлагают анализ текста посредством обработки естественного языка. Этот набор *API* может анализировать текст, чтобы помочь вам понять его концепции, сущности, ключевые слова, настроения и многое другое.

На рисунке 1 представлен пользовательский интерфейс разработанного веб-приложения. На данном представлении можно найти фильм по названию или выбрать интересующие жанры, год выпуска и порядок сортировки.

На рисунке 2 представлено модульное окно для просмотра детальной информации о фильме, а также продемонстрированы отзывы и их оценки эмоциональной тональности.

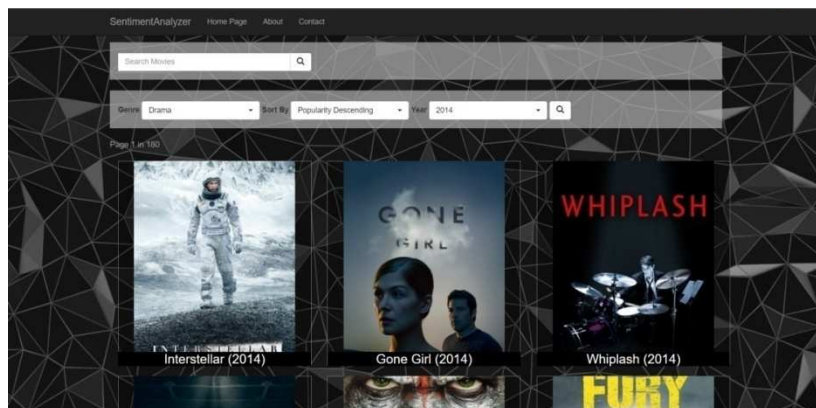


Рисунок 1 – Поиск фильмов

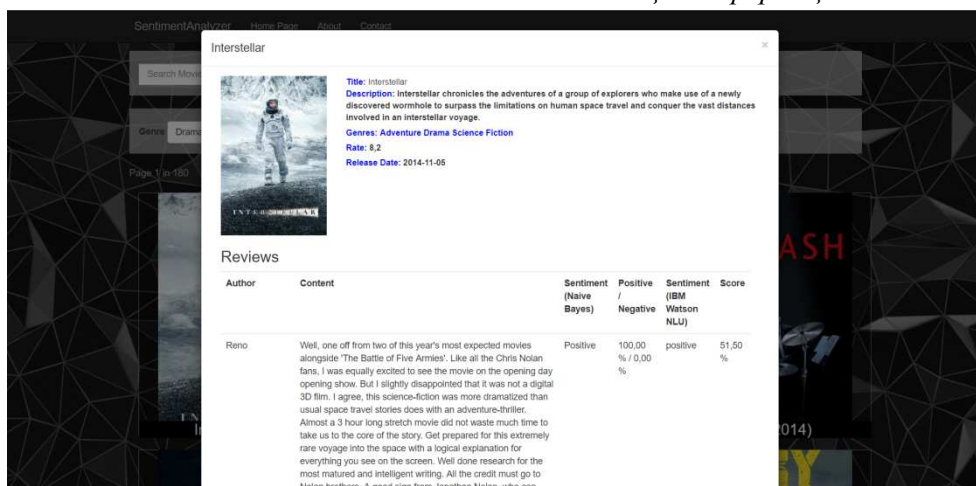


Рисунок 2 – Окно для просмотра информации о фильме

Таким образом, благодаря данному веб-приложению можно быстро и легко оценить рейтинг фильма исходя из полученных тональностей отзывов. Соответственно при использовании продемонстрированного веб-приложения можно значительно сэкономить время, особенно, если фильм имеет большое количество отзывов.

ЛИТЕРАТУРА

1. Pang B. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval / Б. Панг, Л. Ли. – Москва: Вильямс, 2008. – 235 с.

2. Теорема Байеса [Электронный ресурс] – Режим доступа: <http://www.nsu.ru/mmfm/tvims/chernova/tv/lec/node15.html>. Дата доступа: 19.02.2019.

УДК 004.056

Студ. Н. И. Борсук
 Науч. рук. доц. Д. В. Шиман
 (кафедра программной инженерии, БГТУ)

СЕРВИС ПОДБОРА НЕДВИЖИМОСТИ ДЛЯ ДОЛГОВРЕМЕННОЙ АРЕНДЫ

Первым делом при подборе недвижимости для долгосрочной аренды человек идет за помощью к сервисам в интернете для подбора жилья. Современные сервисы предлагают возможности подобрать жилье по существующим объявлениям. Сервис для размещения предложений аренды недвижимости должен выполнять следующие основные задачи:

- создание объявления об аренде;
- поиск объявлений по параметрам;

- просмотр необходимой информации об недвижимости и контакты арендодателя.

Именно благодаря таким сервисам можно, потратив некоторое время, найти себе жилье на долгосрочную перспективу.

Целью проекта была реализация сервиса для размещения предложений об аренде недвижимости. Данные объявления можно искать по различным параметрам и получать уведомления о новых предложениях. На основании вышеизложенных проблем существующих решений сервисов по работе с недвижимостью для аренды и необходимостью реализовать базовый функционал данных сервисов, были сформулированы следующие задачи:

- создать клиент-серверное приложение;
- клиент должен реализовывать принцип SPA;
- пользователь может создавать объявления об аренде с описанием условий проживания;
- пользователи могут производить поиск по городу, параметрам недвижимости и условий проживания;
- пользователи могут сохранять поисковые запросы для дальнейшего оповещения на почту;
- пользователь может в полной мере просматривать информацию об недвижимости.

Клиентское приложение должно отображаться в большинстве браузеров таких как GoogleChrome, MozillaFirefox, Safari, Opera и так далее.

Серверная часть должна соответствовать REST архитектуре. Это необходимо для реализации сервиса на различные клиентские платформы, будь то десктопное приложение, браузер, мобильное приложение.

Разработка. Для разработки серверной части приложения была выбрана платформа Node.js. Архитектура выполнена в лучших традициях методологии “Чистой архитектуры”. Данный вид архитектуры позволяет упростить тестирование благодаря слабой связности компонентов, а также дает возможность без проблем менять фреймворки для работы с внешними данными, такие как фреймворки баз данных и HTTP-серверы.

В качестве базы данных была использована PostgreSQL. Она была выбрана из-за некоммерческой основы использования и обширной документацией.

Клиентская часть приложения написана с использованием ReactJS. Это фреймворк для разработки пользовательских интерфейсов. Вместе с ReactJS на клиентской стороне приложения использу-

ется Redux и Redux-Saga. В совокупности эти три библиотеки позволяют разрабатывать SPA приложения, которые отличаются отсутствием перезагрузки страницы при запросах и повышенным пользовательским опытом. В качестве библиотеки, которая позволяют использовать готовые CSS стили была использована Bootstrap4. Готовые компоненты позволяют оперативно составлять внешний вид сайта.

Проект был полностью реализован на языке JavaScript. Данное решение позволило быть полностью сфокусированным на разработке на одном языке программирования, пользоваться некоторыми библиотеками как на серверной части приложения, так и на клиентской. Также была отличная возможность разделять кодовую базу между двумя частями приложения.

В приложении существует две роли, а именно обычный пользователь и администратор.

В привилегии администратора входит бан пользователей за нарушение правил пользования сервиса, а также удаление объявлений об продаже за явное нарушение правил оформления и цензуры. Аккаунт администратора создается без использования веб-интерфейса.

Обычный пользователь после регистрации обязан подтвердить свой аккаунт переходом по ссылке из письма на почту, после чего он обязан заполнить о себе необходимую контактную информацию и выбрать тип арендодателя. Тип арендодателя может быть “Частный арендодатель” либо “Агентство”.

Основной функционал пользователя сосредоточен на таких функциях как создать объявление, закрыть объявление, найти объявления по фильтрам и сохранить фильтры для дальнейших уведомлений. В строке поиска в хедере пользователь может начать вводить город, где он хочет найти предложения. Либо же он может это сделать на главной странице. Помимо поиска по городу, существует поиск по таким критериям как промежуток цены аренды в месяц, промежуток общей площади, количества комнат, тип арендодателя и другие. Важной частью фильтрации являются указания особенностей сдачи в аренду, такие как проживание с домашними животными, вредные привычки, отсутствие предоплаты и так далее.

Каждый поисковой запрос зарегистрированный пользователь может сохранить к себе в профиль. При каждом создании объявления, которое по всем критериям подходит под пользовательские условия, пользователь получит уведомление на ранее подтвержденную почту.

Со страницы профиля пользователь может управлять статусом его объявлений, а также загружать фото для этих объявлений. Помимо этого, пользователь может удалять свои пользовательские фильтры, если он более не нуждается в услугах уведомлений.

В настоящее время существует достаточно количество сервисов, позволяющих размещать и производить поиск предложений об аренде недвижимости. Но большинство из них не решают такие насущные проблемы как описание условий проживания, таких как проживание с детьми, животными или вредные привычки. А также не всегда существует возможность своевременного оповещения пользователя при создании предложения, которое может ему подойти.

В описываемом проекте был реализован основной функционал, позволяющий производить полный цикл создания объявления, с дальнейшим управлением его состоянием; поиск по городу и фильтрам предложений; сохранение поисковых запросов и дальнейшее оповещение пользователей по сохраненным фильтрам.

ЛИТЕРАТУРА

1. Официальный сайт платформы Node.JS [Электронный ресурс]. – Режим доступа: <https://nodejs.org/>. – Дата доступа 24.04.2017.
2. Официальный сайт PostgreSQL [Электронный ресурс]. – Режим доступа: <https://www.postgresql.org/>. – Дата доступа 24.04.2017.
3. Описание архитектуры “Чистая архитектура” [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/269589/>. – Дата доступа 24.04.2017.
4. Официальный сайт ReactJS [Электронный ресурс]. – Режим доступа: <https://reactjs.org/>. – Дата доступа 24.04.2017.

УДК 004.056

Студ. Е. С. Сазонова
Науч. рук. доц. Д. В. Шиман
(кафедра программной инженерии, БГТУ)

ВЕБ-ПЛАТФОРМА ДЛЯ МУЗЫКАНТОВ

Разработанная веб-платформа для музыкантов и людей, интересующихся творчеством различных исполнителей, базируется на двух вещах: первая – это проблема получения информации своевременно, а вторая – это область досуга людей, а точнее – музыка.

Музыка, в наше время, является неотъемлемой частью жизни буквально каждого человека, а также, разумеется, что из всего обилия предлагаемых музыкальных групп и артистов, каждый человек выде-

ляет для себя особенных, любимых исполнителей. Именно о концертах таких, любимых, исполнителей, человек хотел бы быть всегда в курсе, а особенное, если данные мероприятия проводятся в городе, где он проживает.

Разработанное программное средство служит для решения проблемы получения своевременной информации о планируемых музыкальных мероприятиях артистов, интересующих конкретного человека.

Первый этап разработки данного веб-приложения включал выбор и проектирование архитектуры. Для реализации была выбрана микросервисная архитектура.

Основная идея микросервисной архитектуры заключается в разбиении большого приложения на независимо функционирующие модули, отвечающие за определённые задачи и взаимодействующие друг с другом посредством API.

Одной из самых важных характеристик в любом архитектурном решении является масштабируемость. Микросервисная архитектура предоставляет хорошие возможности для масштабирования отдельных, особо нагруженных сервисов, а также она предоставляет возможности для функциональной декомпозиции. Так как отдельная функциональность представляет собой отдельный сервис, имеется возможность изменять и доставлять каждый из микросервисов независимо.

Back-end программного средства реализован с использованием технологии ASP.Net Core 2.2. ASP.NET Core является кроссплатформенной, высокопроизводительной средой с открытым исходным кодом. Кроссплатформенность является одним из решающих факторов выбора данной технологии. Front-end приложения написан на Angular 7 с использованием Bootstrap 4.

В качестве основной базы данных используется Microsoft SQL Server.

Центральным объектом базы является таблица с информацией о пользователях.

Необходимо сказать о том, что в разработанном приложении существует два типа пользователей, музыканты и обычные пользователи, желающие следить за чьим-либо творчеством. Основываясь на этом разделении и формировался ряд таблиц в базе данных, специфичных для каждого из этих типов юзеров.

Также есть таблицы, никак не относящиеся к данным разделением. Например, таблицы с информацией о местоположениях или жанрах музыки.

На рисунке 1 представлена архитектура разработанного веб-приложения.

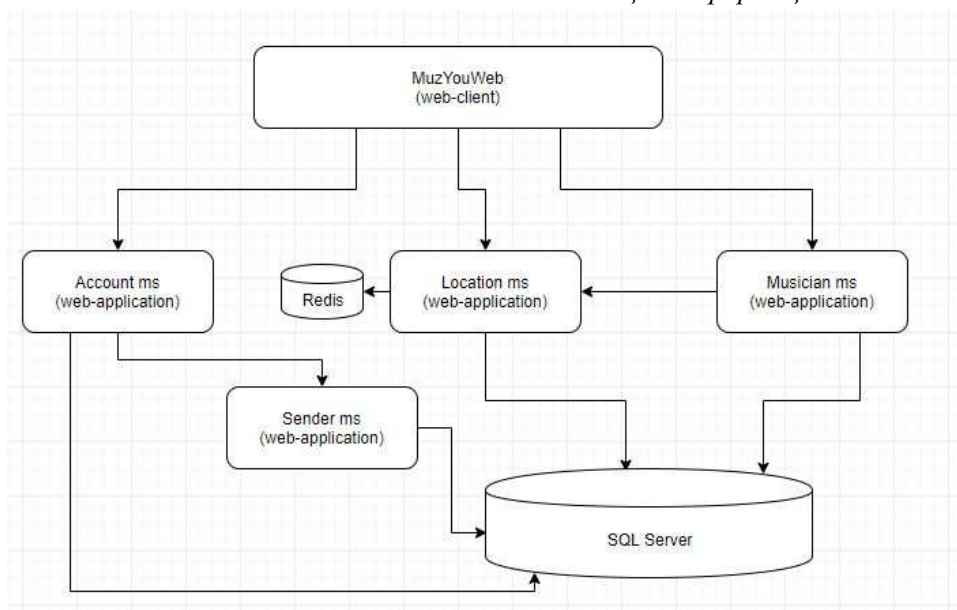


Рисунок 1 – Архитектура веб-приложения

В процессе построения архитектуры приложения были разработаны 5 микросервисов.

MuzYouWeb является веб-клиентом в данной системе, таким образом запросы во все остальные микросервисы производится через него.

Account микросервис отвечает за аккаунты пользователей в приложении. Основная функциональность данного микросервиса включает: регистрацию, авторизацию и настройку пользователей.

Location микросервис отвечает за работу со сторонним API, предоставляющим информацию о странах, регионах и городах, а также за сохранение локаций или местоположений в базу.

Musician. Данный микросервис построен для работы с профилями музыкантов, а именно: подписки и отписки на исполнителей, работы с концертами и получения информации о музыкальных группах.

Также существует Sender, другими словами - Отправитель. Этот микросервис отвечает за рассылку сообщений пользователям. Реализованы два типа рассылки:

- отправка сообщения пользователю на email для подтверждения регистрации на сайте;
- рассылка информации о новых концертах музыкантов, на которые подписаны пользователи.

Важной функциональностью данного приложения является возможность оповещения пользователей по почте о важных для них событиях. Каждый пользователь сам выбирает включать или нет данных тип рассылки.

На рисунке 2 представлена схема нотификации пользователя, после создания нового концерта музыкантом, на которого он подписан.

Основные этапы системы нотификации пользователей:

1. Создание нового концерта музыкантом.
2. Вызов процедуры базы данных для внесения всего списка пользователей для оповещения в специальную таблицу.
3. Выборка всех нотифицируемых пользователей по заданному таймеру.
4. Группировка данных о концертах для каждого юзера в отдельные сообщения.
5. Рассылка оповещений.
6. Удаления данных, которые уже использовались для оповещения юзеров из специальной таблицы.

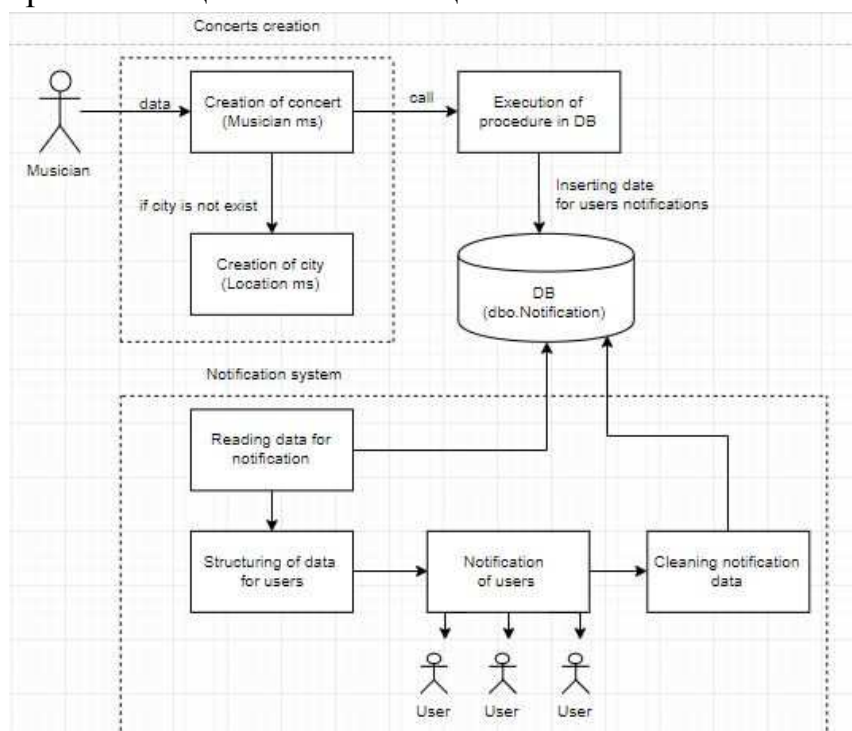


Рисунок 2 – Схема нотификации пользователя

Таким образом, разработано приложение, помогающее людям своевременно получать информацию об интересующих их концертах и музыкальных группах.

ЛИТЕРАТУРА

1. Docs.microsoft.com [Электронный ресурс] / docs.microsoft.com – Режим доступа: <https://docs.microsoft.com/> Дата доступа 21.04.2019.
2. Wikiprdia.org [Электронный ресурс] / wikipedia.org – Режим доступа: <https://ru.wikipedia.org/wiki> Дата доступа 21.04.2019.

Студ. Е. В. Коляго
Науч. рук. ст. преп. А. С. Наркевич
(кафедра программной инженерии, БГТУ)

ВЕБ-ПРИЛОЖЕНИЕ «АФИША МЕРОПРИЯТИЙ»

Жизнь современного человека сложна, многогранна, динамична, интересна и очень переменчива. 21 век – век прогресса, подаривший нам множество технических открытий, благодаря которым человеку стало легче жить, потому что некоторую физическую работу, которую он раньше выполнял, теперь выполняет машина, робот.

Существует большое количество сервисов, которые предоставляют пользователям списки мероприятий. Но большая проблема данных сервисов состоит в том, что они не содержат полной информации обо всех событиях, имеют неудобную навигацию и поиск. Другой отличительной чертой всех существующих приложений является отсутствие небольших мероприятий. Веб-приложение «Афиша мероприятий» было создано с целью решения этих проблем путём предоставления возможности обычному пользователю создавать свои мероприятия и добавления многофакторного подтверждения заявок.

Несмотря на довольно большое количество схожих приложений, на данный момент нет аналогичного сервиса, который ориентирован на мелкие и средние мероприятия.

Поэтому главные критерии для данного приложения – это возможность добавления нового мероприятия на сайт, управление и фильтрация зарегистрированными заявками. А также, что не менее важно – реализации социальной части сайта. В контексте данного приложения будет доступна система оценки мероприятий и система комментариев пользователей.

Для удобного просмотра и управления мероприятиями был определён следующий функционал:

- регистрация и логин пользователей;
- просмотр мероприятий и фильтрация мероприятий по категориям и дате;
- получение подробной информации о событиях;
- добавление комментариев и выставление рейтингов;
- покупка билетов на мероприятие;
- создание заявок на регистрацию мероприятия;
- просмотр галереи мероприятия и добавление фотографий;
- возможность управление пользовательскими полномочиями;

– подтверждение и менеджмент полученных от пользователей заявок.

Целью данной работы является разработка программного средства, представляющего собой веб-приложение. Для создания приложения, хранящего в себе большое количество мероприятий и информации о них, а также предоставляющего удобный графический интерфейс для работы с ними необходимы серверная и клиентская часть приложения.

Клиентская часть приложения должна:

- обеспечивать кроссплатформенность;
- обладать удобным интерфейсом;
- предоставлять UI для мероприятиями;
- быть легко поддерживаемым и масштабируемым.

Серверная часть приложения должна:

- обеспечивать хранение данных;
- основываться на REST-архитектуре;
- обеспечивать легкую поддержку и расширение.

Для выполнения всех вышеперечисленных условий был выбран следующий набор технологий:

- React;
- Redux;
- Material-ui;
- Webpack;
- .NET Core v2.2;
- SQL;
- EF Core.

В данной системе было определено 3 типа пользователей: неавторизированный пользователь, зарегистрированный пользователь и администратор. Неавторизированный пользователь обладает правами только для просмотра мероприятий, фильтрации их по дате и категориям. Зарегистрированный пользователь имеет возможность создавать заявки на регистрацию событий, добавлять фотографии, оставлять комментарии и выставлять рейтинг. Администратор обладает всеми вышеперечисленными правами, а также имеет доступ к панели администратора, с помощью которой он может управлять всеми пользователями в приложении, изменять информацию о мероприятии, а также управляет заявками от пользователей. Все авторизованные пользователи имеют доступ к личному кабинету, где отображаются все приобретённые билеты и заявки на регистрацию мероприятий.

Для удобного управления заявками и полной уверенности в их положительном влиянии на общество была разработана многоуровне-

вая система подтверждения запросов на создание мероприятия, представленная на рисунке 1.

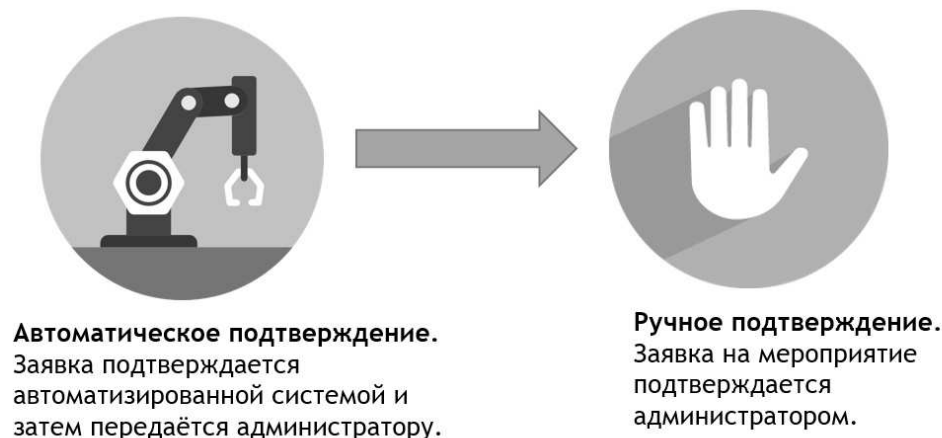


Рисунок 1 – Схема подтверждения заявки

Исходя из предоставленной схемы, можем увидеть, что заявка изначально передаётся для подтверждения сервису, анализирующему наличие в ней недопустимого контента. Следующим шагом является просмотр и подтверждение заявки администратором. Только после этих этапов может быть создано мероприятие.

В ходе реализации функциональных требований было разработано приложение, позволяющее пользователю найти всю необходимую информацию о мероприятии, принять в нём участие, а также при желании создать своё. Использование новейших технологий и заложенная архитектура позволяют эффективно расширять возможности приложения. Кроме того, кроссплатформенность продукта упрощает его интеграцию с различными системами.

ЛИТЕРАТУРА

1. DotNet.com [Электронный ресурс] / dotnet.microsoft.com – Режим доступа: <https://dotnet.microsoft.com/> – Дата доступа 12.04.2019.
2. GitHub.com [Электронный ресурс] / github.com. – Режим доступа: <http://github.com/jbogard/MediatR/wiki>. – Дата доступа 13.04.2019.

УДК 004.42

Студ. Д. А. Татарович
Науч. рук. ст. преп. А. С. Наркевич
(кафедра программной инженерии, БГТУ)

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ «ТАКСИ АГРЕГАТОР»

Цель работы: разработать веб-приложение, предоставляющее широкий спектр функциональных возможностей для водителей такси, клиентов службы такси и администраторов веб-приложения.

Веб-приложение «Такси агрегатор» должно выполнять следующие задачи:

- создание аккаунтов с различными ролями: водителя, клиента и администратора;
- учёт данных о водителях, клиентах и автомобилях;
- предоставление пользователям функционала по созданию и обработке заказов такси;
- защита данных пользователей на всех этапах обработки и хранения;
- учёт поездок и хранение подробных данных для отчетов;
- осуществление поиска по учтённым данным;
- хранение всей информации в базе данных;
- создание и получение статистических отчётов;
- предоставление административного функционала для аккаунтов с ролью администратора.

Данное веб-приложение предназначено для использования как клиентами службы такси, так и водителями такси.

Несмотря на то, что на сегодняшний день в Беларуси услуги заказа такси предоставляет множество приложений, новые приложения продолжают появляться на рынке, а значит разработка такого веб-приложения остаётся актуальной и востребованной.

Веб-приложение должно предоставлять максимально удобный и простой в освоении функционал для каждой роли пользователей, осуществлять безопасную обработку, транспортировку и хранение всех пользовательских данных и выполняться одинаково вне зависимости от устройства и операционной системы.

На сегодняшний день в сфере разработки веб-приложений существует множество технологий, предоставляющих возможность реализовать приложение, полностью соответствующее сформулированным требованиям. Поэтому, на стадии проектирования веб-приложения был проведён сравнительный анализ подходящих техно-

логий. Таким образом был подобран соответствующий целям и задачам проекта набор технологий.

Приложение реализует клиент-серверную модель с внешней базой данных на основе архитектурного шаблона MVC.

В качестве платформы для разработки была выбрана платформа .NET Core, соответственно, был использован веб-фреймворк ASP.NET Core.

.NET Core — это универсальная платформа разработки с открытым кодом, которую поддерживает корпорация Майкрософт и сообщество .NET на сайте GitHub. Она является кроссплатформенной (поддерживает Windows, macOS и Linux) и может использоваться для создания приложений для различных устройств, облачных приложений и Интернета вещей.

ASP.NET Core является кроссплатформенной, высокопроизводительной средой с открытым исходным кодом для создания современных веб-приложений, и предоставляет множество инструментов для разработчика.

Разработку веб-приложения на основе архитектурного шаблона MVC существенно упрощает такой компонент фреймворка ASP.NET Core, как ASP.NET Core MVC.

ASP.NET Core MVC — это упрощенная, эффективно тестируемая платформа с открытым исходным кодом, оптимизированная для использования с ASP.NET Core. К возможностям ASP.NET MVC относятся: маршрутизация, привязка модели, проверка модели, внедрение зависимостей, фильтрация, тестируемость, работа с представлениями.

Для отрисовки представлений ASP.NET Core MVC используют подсистему просмотра Razor. Razor — это компактный, выразительный и гибкий язык разметки шаблонов для определения представлений с помощью встроенного кода C#. Razor используется для динамического создания веб-содержимого на сервере. Код на стороне сервера можно полностью комбинировать с содержимым и кодом на стороне клиента.

Для облегчения работы с данными был использован Entity Framework Core. Entity Framework Core — это простая, кроссплатформенная и расширяемая версия популярной технологии доступа к данным Entity Framework с открытым исходным кодом.

EF Core может использоваться как объектно реляционный модуль сопоставления (ORM), позволяя разработчикам работать с базой данных с помощью объектов .NET и устраняя необходимость в напи-

сании большей части кода, требуемого для доступа к данным. EF Core поддерживает множество СУБД, в том числе postgresql 11.

Структура приложения реализована в соответствии с паттерном MVC: клиент обращается к контроллеру, контроллер возвращает представление, представление взаимодействует с контроллером и моделью, модель работает с базой данных, обновляя данные, и возвращая необходимые данные контроллеру. Таким образом, мы получаем чёткую структуру проекта, обеспечивающую удобную разработку и дальнейшую поддержку приложения.

В качестве СУБД для проекта была выбрана PostgreSQL 11. Эта СУБД предоставляет всё, что необходимо для удобного хранения и работы с данными веб-приложения.

В соответствии с планируемым функционалом была спроектирована структура базы данных, состоящая из девяти таблиц. Структура базы данных соответствует третьей нормальной форме, таким образом, достигается целостность данных, независимость данных и минимальная избыточность.

Составленный список технологий позволяет реализовать весь требуемый функционал, и является удобным как для пользователей веб-приложения, так и для разработчика.

ЛИТЕРАТУРА

1. Документация по .NET [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/index> – Дата доступа 15.04.2019.
2. Проектирование реляционных баз данных с использованием нормализации [Электронный ресурс] / С.Д. Кузнецов. – Режим доступа: http://citforum.ru/database/osbd/glava_23.shtml – Дата доступа: 15.04.2019.
3. Документация к PostgreSQL 11.2 [Электронный ресурс] – Режим доступа: <https://postgrespro.ru/docs/postgresql/11> – Дата доступа: 15.04.2019.

Магистрант А. Д. Самаль
Науч. рук. зав. кафедрой Н. В. Пацей
(кафедра программной инженерии, БГТУ)

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ И ИХ РЕАЛИЗАЦИЙ

Важное прикладное значение имеют задачи классификации изображений с большим числом категорий объектов. Проблемы подобного рода возникают в различных областях: например, поиск изображений по ключевым словам, корректное распознавание объектов для робототехники и т.п. До последнего времени алгоритмы для решения данной задачи мало исследовались, в том числе и ввиду вычислительной сложности. Одни из первых проведенных вычислительных экспериментов подобного рода на базах изображений Caltech-101 и ImageNet показали, что применение общепризнанных на настоящее время алгоритмов не дает удовлетворительной точности распознавания. Лучшие результаты для задач данного типа продемонстрировали два класса алгоритмов. Первый из них основан на использовании векторов Фишера [1] для описания изображений и линейной машины опорных векторов [2] в качестве классификатора. Второй класс алгоритмов использует для описания изображений непосредственно интенсивности пикселей, в качестве классификатора используются конволюционные нейронные сети. Альтернативным подходом к улучшению точности распознавания является использование информации о структуре связей между категориями объектов.

Алгоритм обучения с учителем «случайный лес». Случайный лес (randomforest) [3] является одним из наиболее универсальных и эффективных алгоритмов обучения с учителем, применимым как для задач классификации, так и для задач восстановления регрессии. Идея метода заключается в построении ансамбля параллельно обучаемых независимых деревьев решений. Итоговая классификация объекта выполняется путем голосования всех деревьев, входящих в состав ансамбля. Среди достоинств алгоритма можно выделить высокое качество предсказания, способность эффективно обрабатывать данные с большим числом категорий и признаков, внутреннюю оценку обобщающей способности модели, высокую распараллеливаемость и масштабируемость. Также метод обладает всеми преимуществами деревьев решений, в том числе отсутствием необходимости предобработки входных данных и обработкой как вещественных, так и категориальных признаков, поддержкой работы с отсутствующими значениями. Алгоритм CART построения дерева решений строит разбиение

пространства признаков на непересекающиеся области путем его рекурсивного разбиения. Фактически каждому узлу дерева соответствуют некоторая область пространства признаков и правило, по которому осуществляется ее разделение на две, приписанные к дочерним вершинам. Так как задача построения оптимального дерева, минимизирующего ошибку на обучающей выборке при заданном ограничении на его размеры, является NP-трудной [4], для обучения данной модели используется жадная стратегия максимального уменьшения функции, описывающей неоднородность данных $I(D)$, где D есть область в пространстве признаков, соответствующая некоторому узлу.

Метод опорных векторов относится к наиболее известным и широко используемым на практике алгоритмам обучения с учителем. Изначально данный метод был предназначен для задач бинарной классификации, но легко обобщается на задачи классификации с большим числом классов, а также на задачи восстановления регрессии. Если предположить, что выборка линейно разделима, тогда очевидно, что разделяющая гиперплоскость не единственна, поскольку существуют и другие её положения, реализующие то же самое разбиение выборки. Идея метода заключается в том, чтобы выбрать w и $0w$ оптимальным с точки зрения некоторого критерия способом. Можно потребовать, чтобы разделяющая гиперплоскость максимально далеко отстояла от ближайших к ней объектов обоих классов. Первоначально данный принцип классификации возник из эвристических соображений: вполне естественно полагать, что максимизация зазора (margin) между классами должна способствовать более уверенной классификации. В дальнейшем этот принцип получил мощное теоретическое обоснование.

Построение случайного леса с использованием модифицированной матрицы штрафа. Одним из возможных областей использования семантической иерархии объектов при обучении ансамбля деревьев решений является её применение при построении каждого отдельного дерева в ансамбле путем модификации функции неоднородности (индекса Джини). Основная цель данной модификации заключается в том, чтобы увеличить «штраф» за нахождение в узле объектов, помеченных категориями, находящимися далеко друг от друга в семантической иерархии.

Иерархическая классификация. Один из наиболее известных подходов к решению задачи классификации объектов с большим числом категорий заключается в использовании иерархической классификации [5]. Основная идея данного подхода заключается в построении иерархии категорий объектов, организованной в виде дерева та-

ким образом, что категории, относящиеся к каждому узлу дерева, делятся на некоторое число кластеров, каждый из которых соответствует одному узлу-потомку. Процесс продолжается до тех пор, пока в каждом терминальном узле дерева не будет содержаться ровно по одной категории. В каждом узле дерева строится классификатор (обычно бинарный), который обучает модель классификации кластеров категорий, относящихся к узлам-потомкам. Для классификации нового объекта применяется цепочка из классификаторов, ведущая из корня дерева к терминальному узлу, соответствующему тому или иному классу.

ЛИТЕРАТУРА

1. Perronnin F., Liu Y., Sanchez J., Poirier H. Large-Scale Image Retrieval with Compressed Fisher Vectors // Proc.of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2010. P. 3384–3391
2. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. Springer, 2008.
3. Breiman L. Random Forests // Machine Learning. 2001. V. 45. No1.P. 5–32.
4. Гэри М., Джонсон Д. Вычислительные машины труднорешаемые задачи. М.: Мир, 1982.
5. Griffin G., Perona P. Learning and using taxonomies for fast visual categorization // Proc.of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2008. P. 1–8.

УДК 004.042

Студ. Д. С. Столяров, М. И. Трифонов
Науч. рук. ст. преп. А.С. Наркевич
(кафедра программной инженерии, БГТУ)

РАЗРАБОТКА ИГРЫ «ЗМЕЙКА» НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON 3.7.2

Целью данной работы является изучение объектно-ориентированного программирования на примере разработки игры «Змейка» на языке программирования Python 3.7.2. Для этих целей была использована сторонняя библиотека Pygame. Pygame — это библиотека модулей для языка Python, предназначенная для написания компьютерных игр и мультимедиа-приложений, которая базируется на мультимедийной библиотеке SDL.

Кроссплатформенная мультимедийная библиотека SDL (Simple DirectMedia Layer) предоставляет независимый от платформы низко-

уровневый API для реализации единого программного интерфейса к графической подсистеме, звуковым устройствам и средствам ввода, полезна для разработки игр. Pygame-приложения могут работать под Android на телефонах и планшетах с использованием подмножества Pygame для Android. На этой платформе поддерживаются звук, вибрация, клавиатура, акселерометр.

Приложение разработано и использованием принципов объектно-ориентированного программирования (ООП).

Объект — это сущность, экземпляр класса, которой можно посылать сообщения и которая может на них реагировать, используя свои данные.

Объекты создаются с помощью классов.

Класс описывает объект, но является независимым от него. Иными словами, класс — это экземпляр, описание или определение объекта.

Классы оформляются с помощью ключевого слова `class` и в виде блока с отступом, содержащего методы класса (которые являются функциями). Часть класса Snake игры змейка приведена на рисунке 1.

```
class Snake(object):

    """description of class"""
    #Конструктор
    def __init__(self):
        #Параметры змеи
        self.head = [221,232]
        self.body = [[221,232],[210,232],[199,232]]

    #Движение змейки
    def move(self,control):
        if control.flag_direction == "RIGHT":
            self.head[0]+=11;
            control.checkMove = True
        if control.flag_direction == "LEFT":
            self.head[0]-=11;
            control.checkMove = True
        if control.flag_direction == "UP":
            self.head[1]-=11;
            control.checkMove = True
        if control.flag_direction == "DOWN":
            self.head[1]+=11;
            control.checkMove = True
```

Рисунок 1 – Фрагмент класса Snake.

Метод `__init__` — самый важный метод класса. Он вызывается, когда создается экземпляр (объект) класса; имя класса используется как функция.

Все методы должны иметь `self` в качестве своего первого параметра; хотя `self` непосредственно не передается, Python добавляет инструкцию `self` в список сам. При вызове методов также не нужно

включать `self`. В пределах определения метода, инструкция `self` относится к экземпляру класса, вызывающего этот метод.

Таким образом, внутри метода `__init__` с помощью `self.attribute` можно задать начальное значение атрибутов экземпляра, как показано выше.

Можно использовать и другие методы, расширяющие функциональность классов. Первым параметром всех методов должен быть `self`.

Класс `Menu` содержит два метода, которые используются для отображения меню игры и счёта. На рисунке 2 представлено главное окно игры змейка.

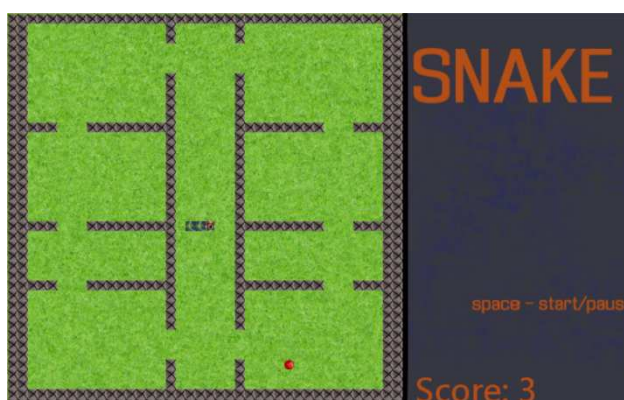


Рисунок 2 – Главное окно игры змейка

Для импортирования классов и настройки окна используется пять основных классов:

- класс змейки: отвечает за создание, движение, взаимодействие с блоками и прорисовку змейки;
- класс еды: отвечает за прорисовку в случайном месте блока еды;
- класс карты: отвечает за расстановку блоков, смену уровня и прорисовку блоков;
- класс контроля: отвечает за управление игрой с клавиатуры;
- класс меню: отвечает за боковую панель игры;
- импортируем все классы в один файл.

Окно игры создается размером 700x441 пикселей и игровое поле в нем размером 441x441 пикселей.

Игровые объекты создаются при помощи конструктора (метода `__init__`).

Игровой цикл игры представляет собой 2 цикла, один цикл вложен в другой. Внешний цикл отвечает за отображение главного меню.

Внутренний цикл служит для реализации игровой логики и прорисовки объектов игры.

ЛИТЕРАТУРА

1. Эл Свейгард «Учим Python, делая крутые игры» — М.: Эксмо, 2018. — 418 с.
2. Сообщество Pygame : [Электронный ресурс] – Электронные данные. Режим доступа: <https://www.pygame.org/news>
3. Проект it-black.ru: [Электронный ресурс] – Электронные данные. Режим доступа: <https://it-black.ru/>
4. Свободная энциклопедия Википедия: [Электронный ресурс] – Электронные данные. Режим доступа: <https://ru.wikipedia.org>

УДК 004.41

Магистрант Д.А. Радиванович
Науч. рук. проф. И. Г. Сухорукова
(кафедра программной инженерии, БГТУ)

ИСПОЛЬЗОВАНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ КЛАССИФИКАЦИИ ЭЛЕКТРОННОЙ КОРРЕСПОНДЕНЦИИ

Наиболее популярным и эффективным подходом для фильтрации электронной корреспонденции на данный момент является использование машинного обучения с учителем по различным признакам, основанными как на содержании сообщений, так и на свойствах отдельных профилей пользователей.

Машинное обучение – это область научного знания, имеющая дело с алгоритмами, «способными обучаться». Необходимость использования методов машинного обучения объясняется тем, что для многих сложных – «интеллектуальных» – задач очень сложно (или даже невозможно) разработать «явный» алгоритм их решения, однако часто можно научить компьютер обучиться решению этих задач. Одним из первых, кто использовал термин «машинное обучение», был изобретатель первой самообучающейся компьютерной программы игры в шашки А. Л. Самуэль в 1959 г [1]. Под обучением он понимал процесс, в результате которого компьютер способен показать поведение, которое в нее не было заложено «явно». Это определение не выдерживает критики, так как не понятно, что означает наречие “явно”. Более точное определение дал намного позже Т. М. Митчелл: говорят, что компьютерная программа обучается на основе опыта E по отношению к некоторому классу задач T и меры качества P , если каче-

ство решения задач из T , измеренное на основе P , улучшается с приобретением опыта E [2].

На этапе классификации и оценки были протестированы пять алгоритмов: наивный байесовский классификатор (NaiveBayes classifier), метод k ближайших соседей (k -nearest neighbors algorithm, k -NN)), метод опорных векторов (SVM), дерево принятия решений (Decision tree), случайные леса (Random forest).

Для обучения было использовано 40% начальной выборки. В процессе оптимизации гиперпараметров для каждого классификатора использовалась кросс-валидация (CV-10). Кросс-валидация способствует минимизации риска переобучения, и, следовательно, смещения в оценке качества классификатора. Оптимизация проводилась для обеих групп признаков (каждый признак был нормализован и принимал значение в диапазоне $[-1, 1]$). Кроме этого для настройки SVM было отобрано 30% наиболее значимых признаков по критерию хи-квадрат, поскольку это более оптимально с точки зрения вычислительной нагрузки.

Далее приведены результаты оптимизации гиперпараметров с помощью полного перебора по сетке.

После оптимизации гиперпараметров каждый из алгоритмов оценивался на оставшихся 60% данных с использованием обеих групп признаков и CV-10. Также как и в случае поиска по сетке, каждый признак был нормализован, а для SVM использовалось 30% наиболее значимых признаков по критерию хи-квадрат.

В качестве метрик оценки классификаторов были выбраны полнота (R), точность (P) и F -мера. Метрика полноты представляет из себя отношение числа сообщений, корректно классифицированных как спам (True Positives) и общего числа спамовых сообщений (True Positives + False Negatives):

$$R = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (1)$$

Точность – это отношение числа сообщений, корректно классифицированных как спам (True Positives) и общего числа сообщений, классифицированных как спам (True Positives + False Positives):

$$P = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2)$$

F -мера может быть проинтерпретирована как гармоническое среднее между метриками точности и полноты:

$$F1 = \frac{2PR}{P+R} \quad (3)$$

Как видно из таблицы 1, классификаторы на основе дерева показали лучший результат. Random Forests обошел остальные классификаторы по показателю F-показателю F-меры. Наилучший известный мне результат в данной задаче был достигнут у Lee [3] и составляет 98% по той же оценке F-меры. Однако Lee использовал исторические признаки, что является ограничением в нашем случае, следовательно, результат можно считать вполне неплохим.

Таблица 1 — Сравнение показателей классификаторов

Классификатор	Точность	Полнота	F-мера
Naive Bayes	76%	76%	76%
k-NN	86%	86%	86%
SVM	86%	86%	86%
Decision Tree	89%	89%	89%
Random Forest	93%	93%	93%

В результате работы были исследованы 5 алгоритмов машинного обучения: наивный байесовский классификатор, метод k ближайших соседей, метод опорных векторов, решающее дерево и случайный лес. Лучший результат продемонстрировал алгоритм Random Forest - 93%. Наиболее высокий известный в данной задаче составляет 98%, однако в нем отсутствуют какие-либо ограничения на историчность признаков.

ЛИТЕРАТУРА

1. Mitchell, Thomas M. Machine Learning / Thomas M. Mitchell. — 1 edition. — New York, NY, USA: McGraw-Hill, Inc., 1997.
2. Samuel, Arthur L. Some studies in machine learning using the game of checkers / Arthur L. Samuel // IBM JOURNAL OF RESEARCH AND DEVELOPMENT. — 1959. — P. 71–105
3. Lee, Kyumin. Seven months with the devils: a long-term study of content polluters on twitter / Kyumin Lee, Brian David Eoff, James Caverlee // In AAAI Int'l Conference on Weblogs and Social Media (ICWSM. — 2011.

Студ. А.Н. Мушук
 Науч. рук. ст. преп. И. Г. Сухорукова
 (кафедра программной инженерии, БГТУ)

РАЗРАБОТКА ПРИЛОЖЕНИЯ «ОНЛАЙН-БИБЛИОТЕКА»

В последнее время использование информационных технологий в различных областях деятельности находит широкое применение. Не стало исключением и чтение литературы. С развитием информационных технологий появление различных десктопных и онлайн приложений для чтения книг было лишь вопросом времени. Кроме того, создавались устройства, предназначенные специально для чтения. Все это имело своей целью, во-первых, позволить людям не тратить огромные деньги на покупку бумажных экземпляров книг; во-вторых, облегчить и сделать куда более удобным процесс чтения, в-третьих, экономится время, которое было бы потрачено на процесс выбора и покупки книг.

При планировании разработки мною был проведен опрос среди небольшого, активно читающего сообщества. Объектами опроса было то, что аудитория обязательно хотела бы видеть в приложении для чтения книг. Результаты оказались достаточно необычны, поскольку подавляющее большинство хотело не уникальных для подобных приложений возможностей, вроде изменения размера шрифта, а возможности добавлять закладки и удобной постраничной навигации. Результаты опроса в виде диаграммы можно увидеть на рисунке 1.

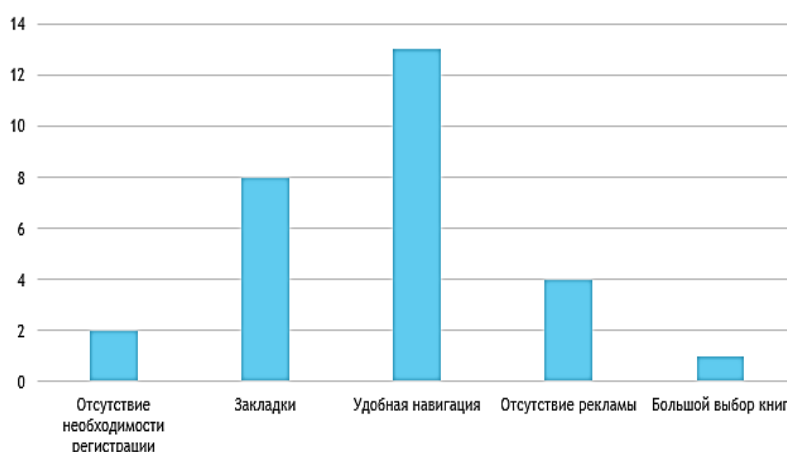


Рисунок 1 – Результаты опроса

Учитывая пожелания аудитории, а также важные для подобных приложений возможности, был проведен обзор четырех популярных веб-приложений, предоставляющих возможность чтения книг. Объектами сравнения были: наличие закладок, удобство чтения, поисковая система, а также возможность добавления книги в избранное (для то-

го, чтобы не забыть прочитать ее в дальнейшем, если на данный момент отсутствует такая возможность и т.п.). Безусловно, можно подобрать дополнительные критерии сравнения, однако перечисленные представляют из себя те базовые функции, которые должно иметь любое подобное приложение.

Первым рассмотренным аналогом был ресурс mybook.ru. Из плюсов можно отметить добавление книги в избранное.

Из минусов: отсутствие системы закладок; неудобная читалка без постраничной навигации и местами некорректный поиск.

Следующим ресурсом является сайт litmir.me. Плюсами данного ресурса являются добавление книги в избранное, удобная читалка с постраничной навигацией и кастомизацией шрифта и хорошо работающий поиск. Минус же всего один: некорректно работающая кнопка создания закладок.

Третьим был рассмотрен сайт librebook.me. Его плюсы: возможность добавлять книги в избранное; удобная читалка с кастомизацией и возможность создания закладок.

Минус, как и у предыдущего ресурса один, а именно навигация по главам (является минусом, поскольку главы могут быть достаточно объемными).

И в конце был оценен сайт loveread.ec. Плюсы данного ресурса: читалка с постраничной навигацией; хорошая поисковая система, которая в отличии от предыдущих ресурсов, имеет возможность поиска серии книг.

Минусы: отсутствие закладок; отсутствие добавлять книги в избранное; неработающая регистрация.

Можно заметить, что из всех рассмотренных ресурсов не было ни одного, который бы удовлетворял всем четырем критериям отбора. Поэтому одной из конечных целей разработки ставилось условие, что разрабатываемое приложение должно удовлетворять всем критериям, перечисленным выше. Кроме того, оно должно быть интуитивно понятным в использовании и иметь простой, но в то же время удобный интерфейс.

Помимо этого, было принято решение добавить форум для обсуждения книг. На первый взгляд необходимость этого, в лучшем случае, спорна. Однако стоит задать себе следующие три вопроса: «Был ли у меня случай, когда я не знал/ла что почитать?», «Было ли у меня желание поделиться мыслями о прочитанной книге с другими людьми или даже посоветовать ее?», «Был ли случай, когда мне приходилось искать дополнительные сведения о сюжете для полноты его понимания?» Скорее всего у каждого человека, который читает боль-

ше одной-двух книг в год были такие мысли. Поэтому появляется необходимость в функционале, который позволит аудитории решить эти проблемы. Таким функционалом обладает форум.

Планируется, что на форуме будут отдельные темы под каждого автора, внутри которых будут темы по каждой книге данного автора, и внутри тем по книгам будут создаваться пользовательские обсуждения, касающиеся определенной книги.

Разработка приложения будет вестись с использованием платформы ASP.NET CORE версии 2.1, back-end часть разрабатывается на языке программирования C#, front-end часть разрабатывается с использованием языка разметки html, с использованием вставок кода на C# благодаря возможностям движка представлений Razor[1]. Разработка ведется в соответствии с концепцией MVC[2]. Все данные хранятся в базе данных SQL. База данных спроектирована с использованием СУБД Microsoft SQL Server. Взаимодействие между базой данных и приложением осуществляется при помощи Entity Framework Core версии 6.2 [3].

В приложении предусмотрены три роли:

- администратор (обладает полным доступом к ресурсу);
- модератор (обладает полным доступом к ресурсу, за исключением возможности менять роли пользователей);
- пользователь (возможность читать книги, создавать обсуждения и комментарии, редактирование своего списка избранной литературы и закладок).

Разработанное приложение обладает следующим функционалом:

- постраничная навигация;
- поддержка закладок, редактируемых в профиле пользователя;
- поиск литературы по жанрам, авторам и названиям;
- форум.

Подводя итог, можно сказать, что разработка данного приложения- это не только ценный опыт, но и возможность создать продукт, позволяющий полностью удовлетворить запросы потребителя в во-просе чтения литературы.

ЛИТЕРАТУРА

1. ASP.NET Core полное руководство [Электронный ресурс] / metanit.com – 2012-2019 – Режим доступа: <https://metanit.com/sharp/aspnet5/> – Дата доступа: 07.04.2019 г.

2. Model-View-Controller [Электронный ресурс] / Википедия – 2019. – Режим доступа: <https://ru.wikipedia.org/wiki/Model-View-Controller/> – Дата доступа: 07.04.2019 г.

3. Руководство по Entity Framework Core [Электронный ресурс] / metanit.com – 2012-2019. – Режим доступа: <https://metanit.com/sharp/entityframeworkcore/> – Дата доступа: 07.04.2019 г.

УДК 004.62

Студ. С.С. Яромич

Научный руководитель ст. преп. И.Г. Сухорукова
(кафедра программной инженерии, БГТУ)

WEB-ПРИЛОЖЕНИЕ ДЛЯ ОРГАНИЗАЦИИ ДЕЛОПРОИЗВОДСТВА ДЕКАНАТА

Работа любой организации, сопряжена с обработкой большого количества документов, потребностью их составления и обработки. В то же время внедрение бумажной документации в большей степени неэффективно. Одним из методов упрощения задачи работы с документами считается внедрение в организации систем автоматизированного формирования документов. Эти системы имеют все шансы быть сделаны с использованием web-технологий, где участник документооборота считается пользователем сервиса сети фирмы, а обработчиком выступает сервер, выполняющий прием, обработку, сбережение и выдачу информации по запросу.

В качестве аналога была рассмотрена платформа Free Dean's Office [1] (Электронный деканат) – это модуль для среды дистанционного обучения Moodle, который добавляет возможность управления процессом обучения, типичным для российских школ, колледжей и ВУЗов. Free Dean's Office позволяет оперировать такими объектами, как "Специальность", "Дисциплина", "Курс" ("Параллель"), "Академическая группа" ("Класс"), "Семестр" ("Учебный год"), "Учебный план слушателя", "Нагрузка преподавателя", "Итоговые оценки по дисциплинам", "Расписание", "Текущие оценки и посещаемость", "Журнал успеваемости и посещаемости", "Зачетная книжка" ("Дневник"), "Табельный номер преподавателя" и т.д.

Free Dean's Office является свободным программным обеспечением и распространяется под лицензией GNU GPL. Целью проекта Free Dean's Office является помощь отечественным образовательным учреждениям в повышении их эффективности путем обеспечения их современным инструментом управления образовательным процессом.

При проектировании системы разработчики стремятся наладить механизм обратных связей, организующих всех участников образовательного процесса.

Актуальностью моего приложения считается необходимость в разработке системы автоматизированного формирования документов для деканата, на основании данных вводимых различными пользователями, их редактировании и удалении. Разрабатываемая система позволит упростить работу с документацией деканата, в частности с заполнением бланков аттестаций и позволит автоматизировать создание отчётов о работе кураторов учебных групп.

Куратор — тот, кто смотрит за ходом определённой работы или же другого процесса. Воспитательная работа куратора охватывает различные направленности: вопросы успеваемости и посещаемости, подготовки к сессии, поддержке в сессионный этап и анализа итогов сессии, работу с родителями учащихся, информирование о проведении всевозможных событий, мероприятий, социально-психологическая помощь и так далее. В прямые обязанности каждого куратора входит формирование отчёта о проведенной им работе, данный процесс можно оптимизировать с помощью системы автоматизации формирования документов, которая является частью системы автоматизации документооборота.

Функции и возможности куратора в приложении:

- мероприятия, в предоставленном разделе куратор имеет возможность просматривать события, в коих принимает участие он или же его группа. Добавлять личные мероприятия;
- итоги семестра, в предоставленном разделе куратор заносит свежие или же изменяет уже имеющиеся данные об успеваемости группы за подобранный им этап обучения;
- информация о группе, этот раздел разрешает юзеру заносить свежую и менять старенькую информацию о группе за подобранный им период;
- отчёты, этот раздел разрешает юзеру получить отчёт о проведенной работе, на базе имеющейся в основе данных инфы.

Для класса «Декан» интерфейс выглядит следующим образом:

- кураторы, в предоставленном разделе декан имеет возможность добавлять в систему новых кураторов и укреплять за ними группы. Еще в предоставленном разделе выполняется редактирование и удаление кураторов из системы;
- группы, в предоставленном разделе декан имеет возможность добавлять записи в перечень групп или же удалять записи из перечня групп;

- студенты, в этом разделе админ декан имеет возможность добавлять, удалять, редактировать информацию о студентах, а также их принадлежность к группам.

При входе на вебсайт происходит авторизация пользователя по логину в операционной системе. В случае если пользователь прошёл авторизацию как куратор, то он получает доступ к следующим разделам: мероприятия, итоги семестра, информация о группе, актив группы, отчёты. Профиль куратора разрешает манипулировать практически всей информацией, заложенной в базе.

В разделе «Мероприятия» куратор имеет возможность просматривать события, отсортированные по дате, или все незамедлительно, в коих принимает участие его группа, или же куратор проводит их сам, удалять собственные события, замечать произведенные события.

Переход к форме добавления мероприятия происходит по нажатию кнопки «Добавить мероприятие». Добавление событий случается при помощи формы, содержащей надлежащие поля: название мероприятия; запланированная дата.

Декан в отличие от куратора имеет право на добавление в систему новых кураторов, новых групп, а также может выполнить закрепление куратора за группой.

Данное web-приложение выстроено на микросервисной архитектуре с внедрением языка Typescript и базы данных MySQL. Серверная часть предоставляет удобный REST API [2], который могут использовать клиенты при общении с сервером. Выбор предоставленного стека технологий обоснован надлежащими основаниями:

- быстрая развёртывания на отдельном хосте;
- отличная интеграция базы данных MySQL с приложениями.

Следует отметить, что данное приложение решает некоторое множество проблем, которые имеются на сегодняшний день в деканате и не решаются современными приложениями-аналогами. В частности, данный проект дает возможность преподавателям удаленно выставлять аттестацию, а представители деканата получают возможность формирования отчетов по итогам аттестации или импорта итогов аттестации в MicrosoftExcel для дальнейшей обработки.

ЛИТЕРАТУРА

1. Электронный деканат (FreeDean'sOffice) [Электронный ресурс]. Режим доступа: <http://www.deansoffice.ru>.
2. REST [Электронный ресурс] / Википедия– 2019. – Режим доступа: <https://ru.wikipedia.org/wiki/REST>. – Дата доступа: 07.04.2019г.

УДК 004.42

Студ. Т. М. Шкода
Науч. рук. проф. И. Г. Сухорукова
(кафедра программной инженерии, БГТУ)

СИСТЕМА УПРАВЛЕНИЯ ПРОЕКТАМИ НА БАЗЕ ПЛАТФОРМЫ «SALES FORCE»

Понятие CRM (Customer Relationship Management) системы все чаще встречается в бизнес-моделях, в которых главным направлением деятельности компании являются меры по обеспечению эффективного маркетинга, продаж и обслуживания клиентов. Поддержка этих бизнес-целей включает сбор, хранение и анализ информации о потребителях, поставщиках, партнёрах, а также о внутренних процессах компании. Функции для поддержки этих бизнес-целей включают продажи, маркетинг, поддержку потребителей.

Система управления взаимоотношениями с клиентами (CRM-система, сокращение от англ. Customer Relationship Management) — прикладное программное обеспечение для организаций, предназначенное для автоматизации стратегий взаимодействия с заказчиками (клиентами), в частности для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними, установления и улучшения бизнес-процессов и последующего анализа результатов.

Несмотря на все многообразие CRM систем, все они имеют одинаковые свойства:

- фронтальную часть, обеспечивающую обслуживание клиентов на точках продаж с автономной, распределенной или централизованной обработкой информации;
- операционную часть, обеспечивающую авторизацию операций и оперативную отчётность;
- хранилище данных;
- аналитическую подсистему;
- распределенную систему поддержки продаж: репликацию данных на точках продаж или смарт-карты.

Так же, прежде всего, CRM система должна отвечать основным принципам, без которых использование CRM системы теряет весь смысл.

Самый главный из них – наличие единого хранилища информации. В него собираются все сведения о клиенте и взаимодействии с ним (расширенная клиентская база данных);

Система управления проектами на базе платформы Salesforce позволяет автоматизировать процессы, сократить время на ведение отчетности, отслеживать ситуацию на проекте и в компании в целом. Так же помогает управлять клиентами, вести статистику продаж и анализировать это все.

Для разработки системы управления проектами, были поставлены следующие задачи:

- изучить и выбрать CRM систему для реализации приложения;
- изучить площадки поиска заказчиков;
- изучить бизнес процессы в управлении проектом;
- реализовать бизнес процессы в выбранной CRM системе.

Целью данной работы является разработка программного средства, представляющего собой CRM систему. Для создания приложения, обеспечивающего управление проектами, выбрана платформа Salesforce.com. На платформе Salesforce.com должна быть реализована клиентская и серверная часть приложения.

Клиентская часть приложения должна:

- обеспечивать кроссплатформенность;
- обладать удобным интерфейсом, стилистически похожим на salesforce.

Серверная часть приложения должна:

- обеспечивать хранение данных;
- обеспечивать легкую поддержку и расширение;
- основываться на сервис-ориентированной архитектуре.

При реализации данного проекта были использованы следующие технологии:

- Salesforce.com;
- Ant;
- Migration Tool;
- Salesforce Process Builder;
- Apex;
- Visualforce;
- Salesforce Custom Objects;
- JavaScript;
- HTML;
- CSS;
- Salesforce Workflow rules;
- Salesforce Formula.

Для решения задачи по управлению проектом, приложение должно обладать следующим функционалом:

- создание, редактирование, просмотр заказчиков;
- сортировка, фильтрация заказчиков;
- создание, редактирование, просмотр сотрудников;
- сортировка, фильтрация сотрудников;
- расчет зарплаты сотрудников;
- создание, редактирование, просмотр проектов;
- сортировка, фильтрация проектов;
- расчет прибыли проектов;
- создание, редактирование, просмотр milestones;
- сортировка, фильтрация milestones;
- расчет прибыли milestones;
- отслеживание бюджета компании с помощью транзакций;
- возможность бронирования митинга с ProjectManager с публичного сайта;
- отправка подтверждения/отказа встречи по email.

В ходе разработки получилось приложение, удовлетворяющее основные потребности пользователя в управлении проектами. Текущая система позволяет работать с ней в любом месте, где есть интернет и браузер. Так же не требует поддержки со стороны программиста.

ЛИТЕРАТУРА

1. StackExchange [Электронный ресурс] / salesforce.stackexchange.com – Режим доступа: https://salesforce.stackexchange.com/search?q=* Дата доступа 15.04.2019.
2. Developer Salesforce [Электронный ресурс] – Режим доступа: <https://developer.salesforce.com/docs/>. – Дата доступа 15.04.2019.

УДК 004.42

Студ. И.Е. Кремко
Науч. рук. ст.преп. Сухорукова И.Г.
(кафедра программной инженерии, БГТУ)

ОНЛАЙН ПЛАТФОРМА ДЛЯ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА В УЧРЕЖДЕНИИ СРЕДНЕГО ОБРАЗОВАНИЯ

Число информационных систем, автоматизирующих различные процессы, растёт с каждым днем. Мы можем наблюдать это на своих рабочих местах, в кафе, на улице, на различных производствах. Мир не стоит на месте, что побуждает разработчиков, работодателей, мировые корпорации, разрабатывать и внедрять новые, более совершенные методы автоматизации процессов. Одной из таких систем является электронный аналог школьных журналов.

Электронный журнал – информационная система, позволяющая хранить и обрабатывать данные об учебном процессе в учреждении среднего образования. Сам по себе электронный журнал обладает рядом преимуществ. Электронный журнала доступен в любое время и в любом месте. Имея доступ в интернет, а также доступ к специализированной системе, персонал школы освобождается от необходимости поиска конкретного бумажного экземпляра журнала. Электронный журнал позволяет автоматизировать составление отчетов, это освобождает учителя от большого количества рутинной работы. По мере заполнения журнала, в системе накапливается большое количество информации, связанной с выполнением учебной программы, практических работ, а также качестве и количестве оценок. На основании этих данных, автоматизированная система может выполнить расчёты и составить отчеты. Электронный журнал позволяет подсчитывать различные показатели. Расчёт таких показателей как средний балл, качество знаний и т.д. требует большой концентрации внимания, так как небольшой просчёт может привести к искажению результата. Всё что требуется от учителя в данной ситуации – лишь своевременное выставление оценок. Такая автоматизированная система, как журнал, позволяет своевременно обеспечивать учеников и их родителей новой информацией о домашних заданиях, следить за выставляемыми оценками, видеть, какие темы уроков были пропущены и что нужно «подтянуть» в учебном плане. К тому же, электронный журнал, при его правильном хранении и постоянном резервном копировании украсть или уничтожить практически невозможно.

Существует большое количество различных электронных журналов, имеющих, в общем, один и тот же функционал. Поэтому может возникнуть закономерный вопрос: Зачем нам нужна ещё одна платформа, когда на рынке их достаточно много? Ответ прост – внедрение модуля, ответственного за подсчёт рабочего времени учителя. Давайте проведём краткий экскурс в историю и узнаем, как же раньше подсчитывали часы на производстве.

Одна из основных форм учета рабочего времени персонала – табельный учет. Табели заполнялись специально предназначенными для этой работы людьми. По окончании каждого из месяцев табели сдавались в расчётный отдел, где на их основании каждому сотруднику назначалась заработная плата.

Со временем, система совершенствовалась. Через некоторое время, чтобы указывать максимально точное время прихода и ухода были введены контрольные часы. Рабочий отмечал в контрольных часах время прихода (и ухода) и оставлял карточку на проходной. Далее этот табель, как и ранее, сдавался в расчётный отдел.

Такая система учета используется и до сих пор, с тем лишь отличием, что вместо табеля используется прокси-карта. При считывании данных с карты, в базу заносится информация о сотруднике, а также о времени его прихода и ухода. То есть мы наблюдаем следующее: табельный учет был автоматизирован, и теперь является информационной системой.

Рабочее время учителя, нельзя считать по времени прихода и ухода, так как ставка учителя – количество уроков в неделю. Например, у учителя в день 3 урока, но в школе он проводит по 8 часов. Учёт времени будет вестись на основании количества отведённых уроков в день. То есть заработная плата будет начисляться за 3 часа, а не за 8. В связи с этим, был реализован следующий алгоритм подсчёта рабочих часов:

- учитель вносит в систему данные о проведенном уроке: тему урока и домашнее задание. Наличие этих данных является подтверждением, что урок был проведён данным учителем;
- вся внесённая информация сохраняется в базу данных;
- на основании расписания учителя, высчитывается количество часов, которые необходимо отработать;
- на основании внесённых в базу уроков, высчитывается количество отработанных часов;
- по окончании месяца, все внесенные часы должны быть подтверждены руководством школы.

Пример рабочей области учителя по отслеживанию рабочих часов представлен на рисунке 1.

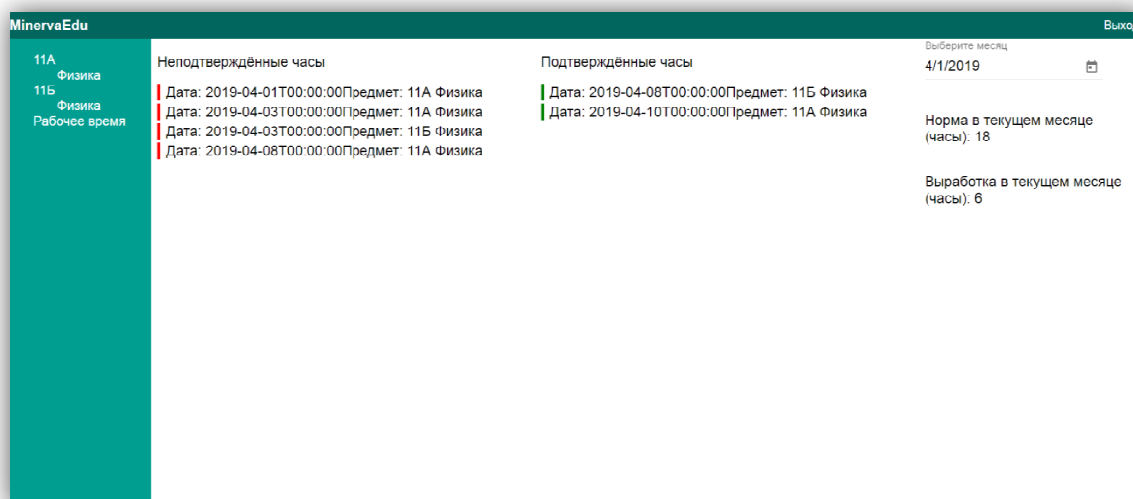


Рисунок 1 – Рабочая область учителя по отслеживанию рабочих часов

На рисунке 1 видно, что представленный учитель ведет уроки астрономии в 11А и 11Б классах. Также можно видеть, что за март

норма и фактическая выработка составили по восемь часов каждая, но, тем не менее, подтверждены только два из восьми уроков.

Использование подобного модуля в электронном журнале, значительно упрощает работу, связанную с подсчётом времени. Больше не нужно заглядывать в журнал и подсчитывать часы вручную. Автоматизированная система всё сделает сама, при этом, не ошибившись в подсчётах. Подобная информационная система в будущем, посредством различных API, может быть внедрена в бухгалтерские приложения. Таким образом, процесс, начиная от внесения выработанных часов, и, заканчивая начислением заработной платы, будет полностью автоматизирован.

ЛИТЕРАТУРА

1. История табельного учета [Электронный ресурс] / HABR.COM – 2013. – Режим доступа: habr.com/ru/post/177987. – Дата доступа: 15.04.2019 г.
2. Табель [Электронный ресурс] / WIKIPEDIA.ORG – 2016. – Режим доступа: wikipedia.org/wiki/Табель – Дата доступа: 15.04.2019 г.
3. Информационная система [Электронный ресурс] / WIKIPEDIA.ORG – 2014. – Режим доступа: wikipedia.org/wiki/Информационная_система – Дата доступа: 15.04.2019 г.

УДК 004.42

Студ. И.И. Разумовский
Науч. рук. проф. И. Г. Сухорукова
(кафедра информационных систем и технологий, БГТУ)

ИНФОРМАЦИОННЫЙ РЕСУРС «POSTIM.VU»

Если раньше интернет пространство характеризовалось наличием строго информационного контента, который представлял ценность для небольшой группы пользователей (да и интернет, собственно, был лишь у единиц), то сейчас здесь можно встретить множество развлекательных интернет-сайтов. Причем их в Сети значительно больше, чем каких-либо других. Скорее всего, это оправдано тем, что людям больше нравится развлекаться, веселиться и смеяться, в чем нет ничего плохого, напротив — только хорошее. И если ранее это были банальный и скучный текст в виде анекдотов и небольшой истории, то сейчас это изображения, видео и даже флэш-анимации.

Порталами называются web-сайты, которые выступают в качестве единого источника для всей информации в конкретной области. Качественный web-портал предлагает пользователю широкий спектр информации, расположенной таким образом, который будет наиболее удобным для доступа пользователя. В условиях правильной разработки, внедрения и поддержки, web-портал становится отправной точкой входа большинства web-пользователей, предоставляя доступ к информации, размещенной даже на других ресурсах сети Интернет. Он становится местом сбора обобщенной группы людей, которые часто сходятся относительно возрастной категории, профессии или хобби.

Web-порталы, по сути, приложения на базе браузера, которые позволяют различную деятельность, включая деловые операции, например, настройку цепочки поставок того или иного продукта. Порталы открывают доступ к структурированным и неструктурированным данным, интегрируют приложения, которые помогают выполнять функции пользователя. А также обеспечивают доступ в режиме реального времени к текущей и последовательной информации. Примерами популярных web-порталов служат Yahoo, MSN и другие.

Проект Postim.by — рекомендательный сервис лучших мест, по отзывам посетителей, который поможет каждому найти самые интересные места в Беларуси. Главные преимущества Postim.by — удобство использования, бонусная программа и возможность самостоятельно добавлять новые и редактировать старые места. На сайте можно, чтобы не потерять, сохранять места и новости в «Избранное», делиться опытом, комментировать и получать важные уведомления.

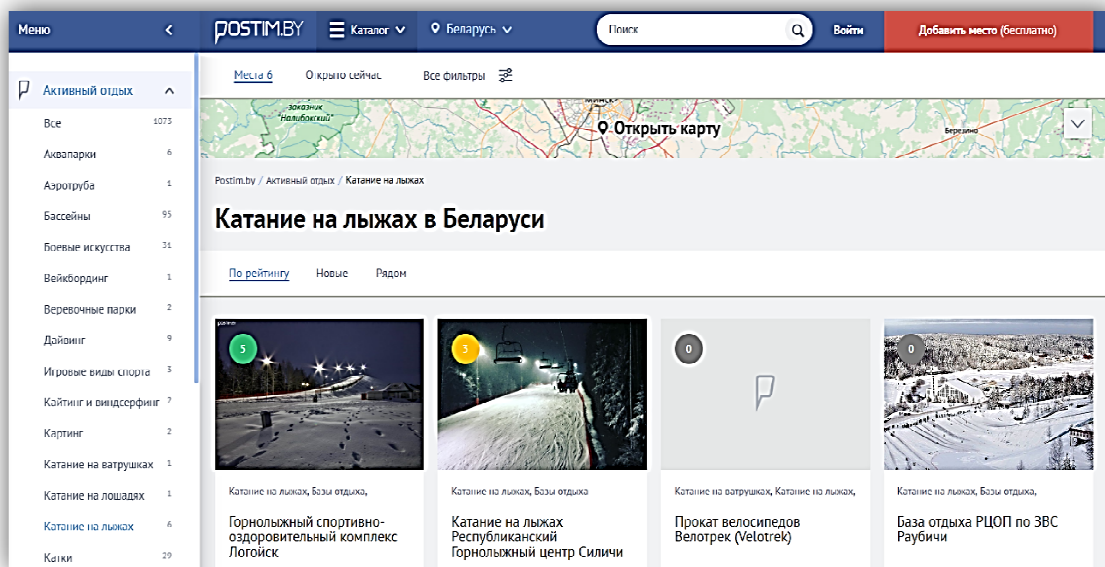


Рисунок 1 – Разработанный проект Postim.by

Главными задачами, которые решают подобные системы, являются вмещение как можно большей актуальной информации о ключевых местах и событиях текущего города либо региона, что является критически важным аспектом при выборе времяпрепровождения людей, проживающих там. Также структуризация информации по определенной области знаний, накопление и обновление информации по заданной тематике, опубликование новостей по тематике, объединение пользователей в группы сообщества.

Актуальность данного веб-приложения обусловлена тем, что оно позволяет снизить нагрузку пользователей на поиск интересных мест, событий или скидок в городе, в котором они проживают на данный момент, либо в целом по стране. Также сервис позволяет автоматизировать процессы поиска, обработки, извлечения информации.

На данный момент существует большое количество аналогов данного сервиса, но они не удовлетворяют необходимым требованиям, так как, во-первых, являются платными, а во-вторых, являются законченными решениями, не предполагающими возможности модификации.

При разработке программного средства необходимо учитывать множество факторов, которые в дальнейшем могут оказать влияние на работоспособность приложения. Для разработки программного средства была выбрана среда разработки PHPStorm, которая обеспечивает высокую эффективность и поддержку многих инструментариев. Для реализации клиентской части использовалась эта же среда разработки.

Программное средство должно состоять из трех частей. Первым основным компонентом является Web сервер, который предоставляет интерфейс для получения, сохранения и обработки данных. Его основная задача – сохранение данных клиентской части и обеспечение доступа к ней в соответствии с правами пользователей. Данный компонент будет разработан с помощью технологии Yii2. Фреймворк построен на базе языка PHP и является крупной единицей в сфере построения веб-приложений.

Фреймворк является кроссплатформенным, что позволяет размещать приложение в разных окружениях, будь то Windows, Linux или же контейнеры Docker.

Для хранения данных использована база данных PostgreSQL в связке с ORM ActiveRecord, которая обеспечивает удобный доступ к данным.

Вторым компонентом является веб-клиент, который предназначен для взаимодействия пользователя с веб-сервером. Данная часть

системы является графическим интерфейсом для взаимодействия с сервером. Она реализована с помощью стека технологий jQuery/Pjax.

Третьим компонентом является поисковой сервер Sphinx, который предназначен для взаимодействия пользователя с поиском мест, новостей или скидок. В сервисе используется система полнотекстового поиска Sphinx, отличительной особенностью которой является высокая скорость индексации и поиска, а также интеграция с существующими СУБД и API для распространённых языков веб-программирования.

ЛИТЕРАТУРА

1. Wikipedia. [Электронный ресурс] / Wikipedia.org – Режим доступа: <https://wikipedia.org>. Дата доступа: 16.04.2019.
2. Github.com. [Электронный ресурс] / Github.com – Режим доступа: <https://github.com/yii2/yii2>. Дата доступа: 16.04.2019.

УДК 004.042

Студ. А. Д. Русакович
Науч. рук. ст. преп. А. С. Наркевич
(кафедра программной инженерии, БГТУ)

ПРИМЕНЕНИЕ ФРЕЙМВОРКА PRISM ПРИ СОЗДАНИИ НАСТОЛЬНЫХ ПРИЛОЖЕНИЙ, ОСНОВАННЫХ НА ТЕХНОЛОГИИ WPF

Составные приложения обычно имеют богатое взаимодействие с пользователем и визуализацию данных, которые воплощают значительную бизнес-логику. Эти приложения обычно взаимодействуют с несколькими серверными системами и службами, используя многоуровневую архитектуру, могут быть физически развернуты на нескольких уровнях. Ожидается, что приложение будет значительно развиваться в течение всего срока службы в ответ на новые требования и возможности для бизнеса. Короче говоря, эти приложения "построены до последнего" и "построены для изменений". Приложения, которые не требуют этих характеристик, не могут извлечь выгоду из использования Prism.

Проектирование и создание приложений в монолитном стиле может привести к решению, которое очень сложно и неэффективно поддерживать. В этом случае "монолитный" относится к приложению, в котором компоненты очень плотно соединены и нет четкого разделения между ними. Как правило, приложения, разработанные и по-

строенные таким образом, страдают от проблем, которые делают жизнь разработчика трудной. Трудно добавить новые функции в систему или заменить существующие функции, трудно устранить ошибки, не нарушая другие части системы, трудно протестировать и развернуть.

Большинство корпоративных приложений сложны и требуют для разработки целую команду программистов и дизайнеров. То, как разрабатывать приложения таким образом, чтобы несколько разработчиков или команд могли эффективно работать над различными частями приложения независимо и быть уверенным, что части приложения гладко соединятся при интеграции, может оказаться серьёзной проблемой.

Эффективным средством решения этих проблем является разделение приложения на ряд дискретных, слабо связанных, полунезависимых компонентов, которые затем могут быть легко интегрированы вместе в "оболочку" приложения для формирования согласованного решения. Приложения, разработанные и построенные таким образом, часто известны как составные приложения. Составные приложения хорошо подходят для различных сценариев клиентских приложений

На рисунке 1 показан пример составного приложения с несколькими серверными системами:

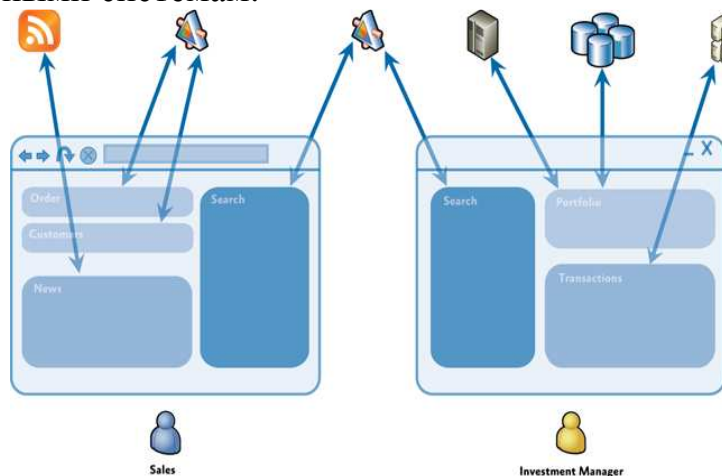


Рисунок 1 –Схема взаимодействия частей приложения с серверными системами.

Prism предоставляет возможности и шаблоны проектирования, которые могут быть вам неизвестны, особенно если вы новичок в разработке шаблонов и составных приложений. Этот раздел содержит краткий обзор основных концепций Prism и определяет некоторые термины, которые будут использоваться в документации и коде:

- **Модули.** Модули — это пакеты функций, которые могут быть независимо разработаны, протестированы и (необязательно) раз-

вернуты. Во многих ситуациях модули разрабатываются и обслуживаются отдельными группами. Типичное приложение Prism построено из нескольких модулей.

- **Shell.** Shell -это хост-приложение, в которое загружаются модули. Shell определяет общую компоновку и структуру приложения, но, как правило, не знает точных модулей, которые она будет размещать. Обычно он реализует общие службы и инфраструктуру приложений, но большая часть функциональности и содержимого приложения реализована в модулях.

- **Представления.** Представления — это элементы управления пользовательского интерфейса, инкапсулирующие пользовательский интерфейс для определенной функции или функциональной области приложения. Представления используются в сочетании с шаблоном MVVM, который используется для обеспечения четкого разделения проблем между пользовательским интерфейсом и логикой представления и данными приложения.

- **Модель представления.** Модели представлений — это классы, инкапсулирующие логику и состояние представления приложения. Они являются частью шаблона MVVM.

- **Модели.** Классы моделей инкапсулируют данные приложения и бизнес-логику. Они используются как часть шаблона MVVM. Модели инкапсулируют данные и любые связанные с ними проверки и бизнес-правила для обеспечения согласованности и целостности данных.

- **Команды.** Команды используются для инкапсуляции функциональных возможностей приложения таким образом, чтобы их можно было определять и тестировать независимо от пользовательского интерфейса приложения. Они могут быть определены как объекты команд или как методы команд в модели представления. Prism предоставляет класс `DelegateCommand` и класс `CompositeCommand`. Последний используется для представления коллекции команд, которые вызываются вместе.

- **Регионы.** Регионы — это логические заполнители, определенные в пользовательском интерфейсе приложения (в оболочке или в представлениях), в которые отображаются представления. Регионы позволяют обновлять макет пользовательского интерфейса приложения без внесения изменений в логику приложения.

- **EventAggregator.** Компоненты в составном приложении часто должны взаимодействовать с другими компонентами и службами приложения слабосвязанным образом. Для поддержки этого Prism предоставляет компонент `EventAggregator`, который реализует меха-

низ событий pub-sub, тем самым позволяя компонентам публиковать события и другим компонентам подписываться на эти события без необходимости ссылки на другие. EventAggregator часто используется, чтобы позволить компонентам, определенным в разных модулях, взаимодействовать друг с другом.

- **Контейнера внедрения зависимостей.** Шаблон инъекции зависимостей (DI) используется в Prism для управления зависимостями между компонентами. Инъекция зависимостей позволяет выполнять зависимости компонентов во время выполнения и поддерживает расширяемость и тестируемость. Prism предназначен для работы с Unity или MEF или с любыми другими контейнерами для инъекций зависимостей через ServiceLocator.

- **Bootstrapper.** Компонент Bootstrapper используется приложением для инициализации различных компонентов и служб Prism. Он используется для инициализации контейнера инъекции зависимостей для регистрации любых компонентов и служб уровня приложения. Он также используется для настройки и инициализации каталога модулей и представления оболочки и модели представления или презентатора.

На рисунке 2 показана типичная архитектура приложения Prism и показано, как все различные возможности Prism могут работать вместе в составном многомодульном приложении:

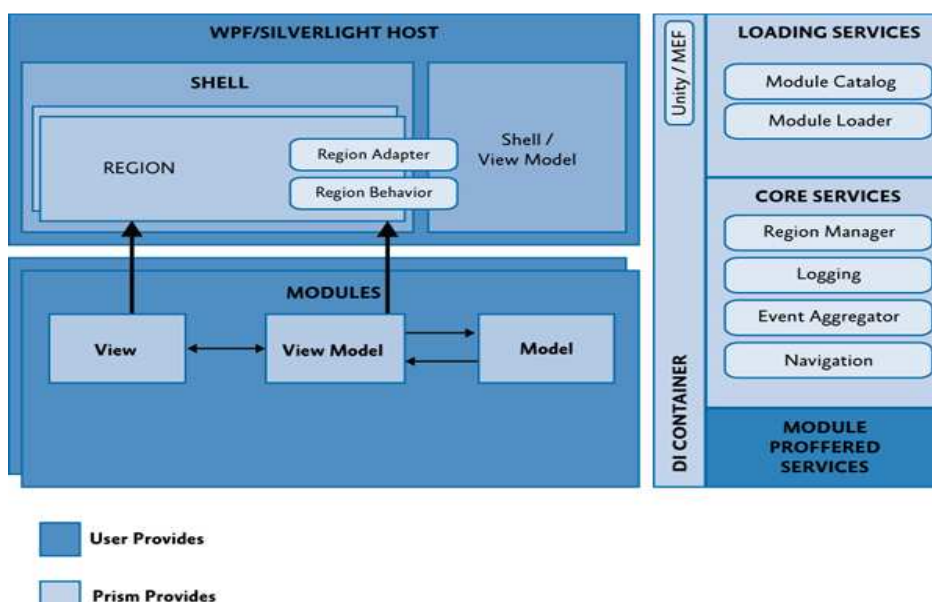


Рисунок 2 – Архитектура приложения, использующего Prism.

Prism разработан таким образом, что вы можете использовать любую из предыдущих возможностей и шаблонов проектирования по отдельности или все вместе, в зависимости от ваших требований и сценария приложения.

ЛИТЕРАТУРА

1. Prismlibrarydocumentation.[Электронный ресурс]. – Электронные данные. – Режим доступа:
<http://prismlibrary.github.io/docs/index.html>
2. Руководство разработчика Prism. [Электронный ресурс]. – Электронные данные. – Режим доступа:
<https://habr.com/ru/post/176851/>

УДК004.042

Студ. Д. С. Кунец

Науч. рук. ст. преп. А. С. Наркевич
(кафедра программной инженерии, БГТУ)

ПРОГРЕССИВНЫЕ ВЕБ-ПРИЛОЖЕНИЯ

Цель работы – разработать веб-приложение для создания и прохождения обучающих курсов на языке программирования Scratch.

Решаемые задачи данного приложения:

- составление, изменение и удаление уроков;
- возможность создавать класс и приглашать в него студентов;
- возможность ведения статистики по урокам;
- возможность поиска урока;
- поддержка режима без доступа к сети интернет, который основан на технологии PWA.

Термин «*Progressive Web App*» введен Алексом Расселом и Франсесом Берриманом. По словам Алекса, прогрессивные веб-приложения — это всего лишь веб-сайты, которые принимали правильные витамины. То, что лежит в основе PWA, нельзя назвать новым фреймворком или новой технологией. Это, фактически, набор передовых методов разработки, которые позволяют сделать так, чтобы поведение веб-приложения оказалось бы очень похожим на поведение классических настольных или мобильных приложений.

PWA дают пользователю новые удобные возможности посредством прогрессивных улучшений приложений. В целом, это означает, что PWA, на мобильных устройствах разных поколений, будут работать примерно одинаково. Конечно, некоторые возможности обычных телефонных приложений могут быть им недоступны, но такие приложения обычно работают на разных устройствах так, как они и должны на них работать.

PWA — это веб приложение, созданное с использованием определенных технологий для достижения заданных целевых показателей, которые представлены на рисунке 1.



Рисунок 1 – Целевые показатели PWA

Надежность (*Reliable*) — приложение загружается и показывается сразу же, вне зависимости от статуса и качества сетевого соединения.

Быстрота (*Fast*) — взаимодействие данными по сети происходит быстро, *UI* плавный и отзывчивый.

Привлекательность (*Engaging*) — делает для пользователя опыт работы с приложением комфортным и приятным.

Service Worker – это воркеры, управляемые событиями, которые выполняются в фоне приложения и действуют в роли посредников между приложением и сетевыми ресурсами. Сервис-воркеры представляют собой *JavaScript*-программы, которые перехватывают обработку определённых событий и выполняют некие задачи. Они могут перехватывать сетевые запросы и кэшировать информацию для приложения. Их можно использовать для загрузки данных, которые планируются использовать для обеспечения работы приложения в оффлайн-режиме. Из хранилищ у *Service Worker*'а есть доступ к *Cache Storage* для *web* ресурсов, и *IndexedDB* для данных. Но, самое главное, полная свобода для реализации бизнес логики. Можно, например, принять запрос от браузера, проверить состояние сети, взять данные из хранилища, произвести с ними операции и вернуть некий результат обратно в браузер — который будет думать, что ответ ему пришел от сервера. Схема работы PWA приложения представлена на рисунке 2.

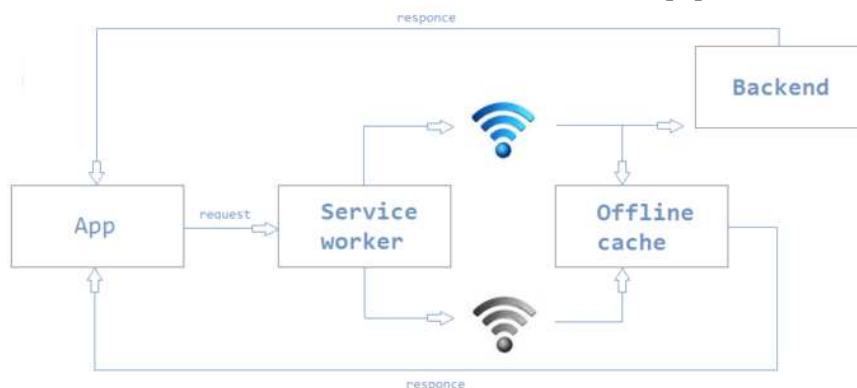


Рисунок 2—Схема работы PWA приложения

Следующие алгоритмы кэширования могут быть использованы:

- Cache with fallback to Network (кэш с резервом для сети) - используйте, если вы создаете автономное первое приложение. Если ответ уже находится в кэше, он будет передан пользователю, и онлайн-запрос никогда не будет выполнен. Если ответ еще не кэширован, приложение попытается загрузить его онлайн и затем поместить в кэш. Этот подход следует использовать для контента, который изменяется очень редко или не изменяется вообще.

- Network with fallback to Cache (сеть с откатом к кэшу) - это подход, при котором онлайн-пользователи всегда получают актуальную онлайн-версию, а автономные пользователи получают кэшированную версию. Используйте его для ресурсов, которые часто обновляются.

- Cache and Network race (кэш и сетевая гонка) - это когда вы ищете ответ в кеше, одновременно запрашивая онлайн-контент. Сначала вы показываете кэшированный ответ пользователя, а затем заменяете его новым контентом сразу после его появления или добавляете новый контент поверх кэшированных страниц, таких как Facebook и Twitter.

- HTTPS требуется, чтобы все передавалось по защищенному соединению. Критичным является тот факт, чтобы на сайте не было ссылок на незащищенные ресурсы.

Application shell— оболочка создаваемого приложения. При разработке PWA сайта «с нуля» для этого могут быть разные стратегии, но на готовом сайте удобно использовать деление на его статическую и динамическую части.

Web App manifest— JSON файл, декларативно определяющий для браузера название приложения, иконку, как будет выглядеть PWA (*fullscreen*, *standalone* и др.), цветовую тему, ориентацию экрана (книжная или альбомная) и некоторые другие параметры. Позволяет

«установить» PWA как отдельное приложение на домашний экран смартфона.

Push Notifications— самая популярная и самая злоупотребляющая технология PWA. Позволяет реализовать всплывающие уведомления на сайте.

Данное приложение позволяет преподавателю самостоятельно создавать уроки, при этом требуется только знание языка программирования *Scratch*. Перед студентами будет поставлена явная цель преподавателя, которую необходимо будет выполнить, используя уже имеющиеся знания по языку программирования *Scratch*. По результатам выполнения заданий преподаватель имеет возможность посмотреть статистику по урокам в группе и сделать некоторые выводы по тому, как студенты усваивают материал. Возможность использования приложения без доступа в интернет позволит активно использовать приложение в учреждениях образования.

ЛИТЕРАТУРА

1. *Progressive Web Apps*[Электронный ресурс] / PCmag. – Режим доступа: <https://developers.google.com/web/progressive-web-apps/>. – Дата доступа 10.04.2019.

УДК 004.738.1+004.85

Маг. Е. С. Помазай

Науч. рук. доц., канд. техн. наук А. И. Парамонов
(кафедра программной инженерии, БГТУ)

ПРОГНОЗИРОВАНИЕ РЕЗУЛЬТАТОВ ТЕННИСНЫХ МАТЧЕЙ С ПРИМЕНЕНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

В последние годы машинное обучение (ML – machine learning) превратилось, в большой бизнес – фирмы используют его, чтобы заработать денег, прикладные исследования бурно развиваются как в индустриальной, так и в академической среде. Прикладное машинное обучение совмещает в себе равные доли математических принципов и полученных эмпирическим путем приемов, – другими словами, это настоящее искусство [1]. С ростом популярности машинного обучения растут и его области применения. Помимо медицины, бизнеса, производства специалисты в области обработки данных, применяют алгоритмы машинного обучения при подборе рекомендаций, определении настроений, разработке искусственного интеллекта в компьютерных играх и т.д. Однако остаются сферы, в которых применение машинного обучения слабо изучено и целесообразность применения

его методов ставится под вопрос. Одной из таких сфер является область спортивных мероприятий.

В спорте, как в сфере деятельности, спортсмену и тренеру необходимо проводить как долгосрочное, так и краткосрочное планирование. При планировании тренировок и иных мероприятий необходимо не только учитывать имеющиеся контрольные показатели, но и уметь прогнозировать дальнейшие тренды их развития. Для «предсказания» этих самых результатов подходят методы машинного обучения. Теннис выбран в качестве области применения алгоритмов по причине его популярности, а значит, сбор данных для проведения исследований не будет затруднительным. Чтобы выявить самую успешную модель для составления наиболее точных предсказаний, необходимо рассмотреть основные алгоритмы машинного обучения и определить какие из них будут иметь наибольшую достоверность. Так как исходная задача – прогнозирование, то подразумевается наличие предварительных данных, обучившись на которых, модель сможет осуществлять предсказания. А значит это классический пример машинного обучения с учителем.

В ходе проведения компьютерных экспериментов проверялись следующие алгоритмы обучения с учителем:

- логистическая регрессия;
- метод опорных векторов;
- дерево принятия решений;
- метод К-ближайших соседей;
- случайный лес;
- наивный Байесовский классификатор.

Исследования позволят определить, на основе какого алгоритма модель будет давать наиболее точные результаты. Однако, в ходе выявления оптимального решения могут возникнуть следующие проблемы.

Поиск обучающей выборки. В ходе проведения исследования с целью выявления оптимальной модели для прогнозирования, необходимы данные для составления обучающей выборки, на основе которой проверяемый алгоритм будет обучаться, и прогнозировать результаты. Данных должно быть достаточно и для составления тестовой выборки, с помощью которой будет проведена оценка точности прогнозов моделей;

Переобучение. Обученные модели могут выдавать нерелевантные показатели предсказаний, из-за того, что алгоритм обучался на данных, которые слишком близко и точно соответствуют конкретному набору данных. Модель будет плохо применима к последующим

данным, так как она является избыточно сложной[2]. Существуют готовые решения, для предсказаний результатов теннисных матчей. Наиболее крупным и точным решением является программа «IBM Keys To Match». Система собирает и обрабатывает всю статистику турниров «Большого шлема». IBM утверждает, что в каждом матче «Keys to the Match» анализируется до 41 миллиона точек данных, в том числе очки, счет, длительность, скорость подачи, процент подач, количество ударов, типы ударов и т. д. На основании анализа система определяет 45 ключевых динамических показателей игры и выделяет из них три, которые наиболее важны для каждого игрока в данном матче. За турнир система создает в общей сложности 5500 предиктивных моделей[3]. Несмотря на все положительные стороны данной системы, она не является открытой и обычным пользователям не представляется возможным получить доступ в нее. Оценить точность системы становится также затруднительно.

Разработка программного средства, позволяющего осуществлять прогнозы теннисных матчей с помощью методов машинного обучения, дает возможность получать предсказания результатов последующих матчей, с помощью которых можно будет разрабатывать спортивные стратегии и программы тренировок. Для решения поставленной задачи использованы различные современные языки программирования, но в первую очередь это Python и R, а также библиотеки scikit learn и pandas.

ЛИТЕРАТУРА

1. Бринк Х., Ричардс Д. Машинное обучение – Санкт-Петербург, Питер, 2017, 330 с.
2. Muller A, Guido S. Introduction to Machine Learning with Python – Sebastopol, O'Reilly Media, 2016, 393 p.
3. IBM BigData and Analytics Hub [Электронный ресурс]. Режим доступа: <https://www.ibmbigdatahub.com/infographic/keys-match-ibm-big-data-and-analytics-powering-predictions>.

Студ. А. Н. Лущицкий
Науч. рук. ассист. Е. А. Чевжик
(кафедра программной инженерии, БГТУ)

ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ ЭФФЕКТИВНОГО УПРАВЛЕНИЯ ЭЛИТНЫМИ ЗДАНИЯМИ

Каждый человек стремится упростить и облегчить поиск решения для ситуаций, с которыми он сталкивается ежедневно. Для эффективного распределения своего времени многие вещи были автоматизированы для последующего комфортного и быстрого использования. В век информационных технологий очень сложно запоминать все вещи, которые окружают нас. Именно поэтому очень важным аспектом в жизни является процесс автоматизации рутинных работ.

Если Вы хотите организовать себе достойный отпуск, то непременно Вы захотите забронировать отель или другой вид жилья на время пребывания. В то же время отель представляет огромное количество услуг, за которыми очень сложно уследить вживую. Будь то хранение ключей, резервирование бассейна, создание нового запроса на починку крана или же доставка ваших почтовых посылок. Для решения всех этих вопросов очень удобно иметь автоматизированное, комфортное в использовании программное средство.

Во время нахождения в здании отеля у Вас могут возникать некоторые вопросы, которые вы хотите решить в минимальные сроки с наименьшими затратами по времени. Именно для этого и нужно программное средство, примером которого является веб-приложение компании BuildingLink [1], которое поможет комфортно, быстро и эффективно решить некоторые ситуации, к примеру:

1. Заявки на ремонт. В случае поломки, жилец создает запрос, с описанием проблемы. Менеджер, который обрабатывает данный запрос, вызывает необходимые службы и после выполнения всех работ закрывает запрос, выдаёт жильцу счёт-фактуру и счет.

2. Учёт почтовых посылок. Консьерж забирает посылки от почтовых служб (DHL, USPS), химчисток, курьеров и вносит эти посылки в систему, отмечая номер апартаментов, тип посылки, имя жильца, комментариев. Для некоторых типов посылок при выдаче требуется прикрепленная фотография электронная подпись или отсканированная ID-карта.

3. Инструкции для сотрудников на стойке регистрации. Жильцы могут оставлять инструкции с информацией кого разрешено впускать или кому разрешено выдавать ключи или с любой другой информации

ей, о которой нужно знать сотрудникам на стойке регистрации. Например, жилец может оставить инструкцию о том, что каждую среду нужно позволять входить домработнице и нужно выдавать ей ключи. Жилец также может указать срок действия инструкции.

4. Резервирование мест общего пользования. Жильцы могут резервировать места общего пользования, такие как солярии, террасы для барбекю, комнаты для вечеринок.

5. Экстренная связь. Управляющий персонал может использовать

рассылку SMS сообщений или голосовых записей для оповещения жильцов в случае крайней необходимости.

6. Разрешения на парковку. Управление местами для парковки автоматизировано, поэтому жильцам и их гостям нужно оставлять заявку, чтобы оставить свой транспорт.

Для создания такого программного обеспечения было принято решение использовать технологии семейства *.NET*. В основу приложения была выбрана технология *ASP.NET* [2], которая является частью экосистемы платформы *.NET* и представляет собой подсистему для построения веб ресурсов.

ASP.NET (*Active Server Pages* для *.NET*) — платформа разработки веб-приложений, в состав которой входит: веб-сервисы, программная инфраструктура, модель программирования, от компании Майкрософт. Входит в состав платформы *.NET Framework* и является развитием более старой технологии *Microsoft ASP*.

ASP.NET внешне во многом сохраняет схожесть с более старой технологией *ASP*, что позволяет разработчикам относительно легко перейти на *ASP.NET*. В то же время внутреннее устройство *ASP.NET* существенно отличается от *ASP*, поскольку она основана на платформе *.NET* и, следовательно, использует все новые возможности, предоставляемые этой платформой.

ASP.NET имеет ряд преимуществ:

1. Использование традиционных языков *.NET*-платформы – *C#* и *VB.NET* для создания логики приложения.

2. Возможность использования существующих элементов разметки, простая конфигурация и возможность создания новых или добавления сторонних пользовательских элементов управления.

3. Независимость от разрешения экрана: поскольку в *ASP.NET* все элементы измеряются в независимых от устройства единицах, приложения на *ASP* легко масштабируются под разные экраны с разным разрешением.

4. Богатые возможности по созданию различных приложений: это и мультимедиа, и двухмерная графика, и богатый набор встроенных элементов управления, а также возможность самим создавать новые элементы, создание анимаций, привязка данных, стили, шаблоны, темы и многое другое.

5. Создание приложений под множество ОС семейства *Windows*– от *Windows XP* до *Windows 10*.

6. Так как программной платформой для приложения был выбрана платформа *.NETFramework*, в качестве языка программирования можно использовать языки *C#* и *VisualBasic .NET*[3]. В разрабатываемой системе выбор пал на язык *VB.NET 6.0*. Одним из критериев выбора является тот факт, что язык *VB.NET* имеет большую популярность и имеет больший приоритет в развитии для корпорации *Microsoft*.

Результатом выполнения данной работы является программный модуль для жильцов элитных зданий, с помощью которого можно улучшить и облегчить некоторые аспекты проживания. К примеру, появится возможность узнать свободную дату и время для бронирования бассейна, тренажерного зала, лаунж-зоны или другого доступного общественного места. Оставлять заявки на ремонт в случае поломки оборудования через веб-приложение. Комфортное отслеживание и оповещение об ожидающих посылках, создание инструкций для сотрудников на стойке регистрации, экстренная связь с менеджерами, разрешение на парковку.

ЛИТЕРАТУРА

1. Официальный сайт компании «Билдинглинк» [Электронный ресурс] – Электронные данные. Режим доступа: <http://buildinglink.com/>
2. Официальный сайт компании Microsoft о технологии ASP.NET [Электронный ресурс] – Электронный данные. Режим доступа: <https://dotnet.microsoft.com/apps/aspnet>
3. Нейгел К., Ивсен Б., Глинн Дж., Уотсон К. - *C# 4.0 и платформа .NET 4 для профессионалов* – 2011 – 120с.

Студ. П. С. Шенец
Науч. рук. доцент, к.т.н. Н. В. Пацей
(кафедра программной инженерии, БГТУ)

РЕГРЕССИОННЫЙ АНАЛИЗ ПРИ РЕШЕНИИ ЗАДАЧИ ПРОГНОЗИРОВАНИЯ

Линейная и логистическая регрессия – наиболее известные виды регрессии. Конечно, они являются наиболее важными среди всех форм регрессионного анализа, но не единственными. Каждый тип регрессии имеет свое значение и определенную область применения. Целью данной работы является анализ семи наиболее часто используемых форм регрессии для их применения в соответствии областью и условиями использования.

Регрессионный анализ – это форма техники прогнозирующего моделирования, которая исследует взаимосвязь между зависимой (целевой) и независимой (-ыми) переменной (-ыми) (предиктором)[1]. Этот метод используется для прогнозирования, моделирования временных рядов и определения причинно-следственной связи между переменными. Например, взаимосвязь между погодными условиями и количеством ошибок в ходе передачи информации по спутниковым каналам лучше всего изучать с помощью регрессии.

Преимущества использования регрессионного анализа заключаются в следующем:

- 1) установление значимых отношения между зависимой и независимой переменной;
- 2) определение силы воздействия нескольких независимых переменных на зависимую переменную;
- 3) сравнение влияния переменных, измеренных в разных масштабах.

Методы регрессии, используемые для прогнозирования, в основном, определяются тремя метриками (или их комбинациями): количество независимых переменных, тип зависимых переменных и форма линии регрессии[2].

Линейная регрессия – одна из первых тем, которые изучают в прогнозирующем моделировании. Зависимая переменная при этом является непрерывной, независимая переменная (переменные) может быть непрерывной или дискретной, а характер линии регрессии – линейный [2]. Линейная регрессия устанавливает связь между зависимой переменной (Y) и одной или несколькими независимыми пере-

менными (X), используя прямую линию наилучшего соответствия, представленную уравнением:

$$Y = \alpha + \beta X,$$

где α – точка пересечения, β – наклон линии, X – независимая, Y – зависимая переменная.

Основная задача стоит в том, чтобы получить наиболее подходящую линию. Задача решается методом наименьших квадратов [2]. Это наиболее распространенный метод, используемый для подгонки линии регрессии. Он рассчитывает линию наилучшего соответствия для наблюдаемых данных, сводя к минимуму сумму квадратов вертикальных отклонений от каждой точки данных до линии. Поскольку отклонения сначала возводятся в квадрат, при добавлении не происходит компенсации между положительными и отрицательными значениями (рис.1).

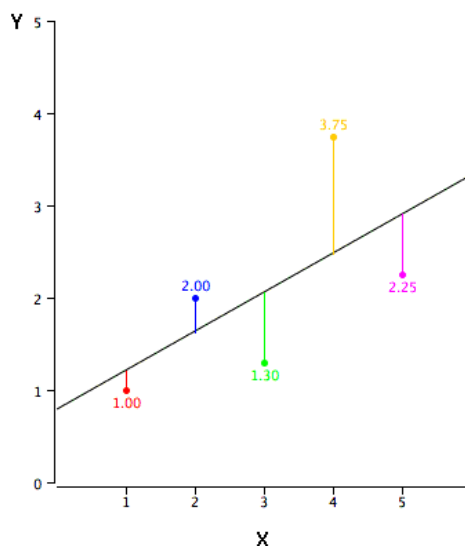


Рисунок 1 – Расчет отклонений точек данных от линии

Особенности линейной регрессии:

- 1) должна быть линейная связь между независимыми и зависимыми переменными;
- 2) возможна автокорреляция, гетероскедастичность (множественная регрессия может обладать мультиколлинеарностью), что приводит к нестабильным оценкам коэффициента;
- 3) чувствительна к выбросам.

Логистическая регрессия используется, чтобы найти вероятность успешного события или неудачи или когда зависимая переменная имеет двоичный характер (например: 0/1, True/False, Yes/No).

Особенности:

- 1) часто используется для задач классификации;

2) не требует линейных отношений между зависимыми и независимыми переменными (применяет нелинейное логарифмическое преобразование);

3) необходимо включить все значимые переменные;

4) требует больших размеров выборки;

5) независимые переменные не должны коррелировать друг с другом (т.е. не иметь мультиколлинеарность);

б) существуют разновидности – порядковая и многочленная логистическая регрессия.

Полиномиальная регрессия. Уравнение регрессии является уравнением полиномиальной регрессии, если степень независимой переменной больше 1. Уравнение, приведенное ниже, представляет полином:

$$Y = \alpha + \beta X^2.$$

В этом методе регрессии лучшая линия соответствия – кривая.

Особенности: использование полинома более высокой степени может привести к переопределению (переобучению).

Пошаговая регрессия используется при нескольких независимых переменных. В этом методе выбор независимых переменных осуществляется на основе наблюдения статистических значений: R-квадрат, t-stats и AIC метрики. Пошаговая регрессия добавляет/отбрасывает ковариации по одному на основе заданного критерия. Распространены следующие методы пошаговой регрессии: стандартная ступенчатая регрессия (добавляет и удаляет предикторы по мере необходимости для каждого шага); прямой выбор начинается с наиболее значимого предиктора в модели и добавляет переменную на каждом шаге; обратное исключение начинается со всех предикторов в модели и удаляет наименее значимую переменную для каждого шага [3].

Конечная цель этого метода – максимизировать мощность прогнозирования с минимальным количеством переменных предиктора. Это один из методов обработки набора данных большей размерности.

Регрессия хребта используется, когда данные мультиколлинеарные. Это приводит к большим значениям дисперсии, что может отклонить наблюдаемое от истинного значения. Регрессия хребта добавляет степень смещения к оценкам регрессии, что уменьшает стандартные ошибки.

Регрессия лассо способна снизить изменчивость и повысить точность моделей линейной регрессии. Однако она отличается от регрессии хребта тем, что использует абсолютные значения в корректи-

рующей функции вместо квадратов. Это приводит к тому, что некоторые оценки параметров оказываются равными нулю.

Особенности:

- 1) допущения аналогичны методу наименьших квадратов;
- 2) сокращает коэффициенты до нуля (облегчает выбор функции);
- 3) если группа предикторов сильно коррелирует, лассо выбирает только один из них и сокращает остальные до нуля.

Регрессия эластичных сетей – это гибрид методов регрессии лассо и хребта. Эластичная сеть используется, когда есть несколько взаимосвязанных функций.

Особенности:

- 1) добавляет групповой эффект в случае сильно коррелированных переменных;
- 2) нет ограничений на количество переменных.

Помимо семи рассмотренных, наиболее часто используемых методов регрессии, можно использовать другие модели, такие как байесовская, экологическая и робастная регрессия.

Основная проблема с регрессионными моделями заключается в том, что чем больше вариантов, тем сложнее становится выбрать правильный.

Чтобы сравнить степень соответствия моделей, анализируются различные метрики: статистическая значимость параметров, R-квадрат, скорректированный R-квадрат, AIC, BIC и др. Важным показателем точности прогнозирования является среднее квадратичное между наблюдаемыми и прогнозируемыми значениями. Установлено, что регрессионные методы лассо, хребта и эластичных сетей хорошо работают в случае высокой размерности и мультиколлинеарности среди переменных в наборе данных.

ЛИТЕРАТУРА

1. Гайдышев И. Анализ и обработка данных. — СПб.: Питер, 2001. — 750 с.
2. Анализ данных и регрессия, Мостеллер Ф., Тьюки Д. Серия: Математико-статистические методы за рубежом. Мю- 1986. – 570 с.
3. Дрейпер Н., Смит Г. Прикладной регрессионный анализ В 2-х кн. М.: Финансы и статистика, 1986. — 366 с.

УДК 517.982.43, 515.127

Студ. И. В. Кебец, З. В. Рекунов
 Науч. рук. зав. кафедрой, доц. О. Н. Пыжкова
 (кафедра высшей математики, БГТУ)

МНОГООБРАЗИЕ ФРАКТАЛОВ: ИХ ПРИМЕНЕНИЕ И ПОСТРОЕНИЕ

В каждом человеке заложена природная любознательность, стремление познавать окружающий его мир. И в этом стремлении человек старается придерживаться логики в суждениях. Анализируя процессы, происходящие вокруг него, он пытается найти логичность происходящего и вывести некоторую закономерность. Даже в хаосе ученые находят связь между событиями. И эта связь — фракталы.

Целью работы является знакомство с понятием фрактала, историей его возникновения, изучение практического применения фракталов в различных науках и их построение.

Фрактал (лат. fractus — дроблённый, сломанный, разбитый) — множество, обладающее свойством самоподобия (то есть целое имеет ту же форму, что и одна или более частей).

Первые примеры самоподобных множеств с необычными свойствами появились в XIX веке в результате изучения непрерывных недифференцируемых функций. Термин «фрактал» введён Бенуа Мандельбротом в 1975 году. Особую популярность фракталы обрели с развитием компьютерных технологий, позволивших эффектно визуализировать эти структуры. Самоподобные объекты с необычными с точки зрения классического анализа свойствами классифицируют по трём основным группам: 1) геометрические фракталы; 2) алгебраические фракталы; 3) стохастические фракталы.

Основными представителями первой группы фракталов являются такие объекты, как: пыль Кантора, снежинка Коха, треугольник Серпинского, губка Менгера (рисунок 1). Именно с них и начиналась история фракталов. Этот тип фракталов получается путем простых геометрических построений. Фракталы этого класса самые наглядные, потому что в них сразу видна самоподобность при любых масштабах наблюдения.

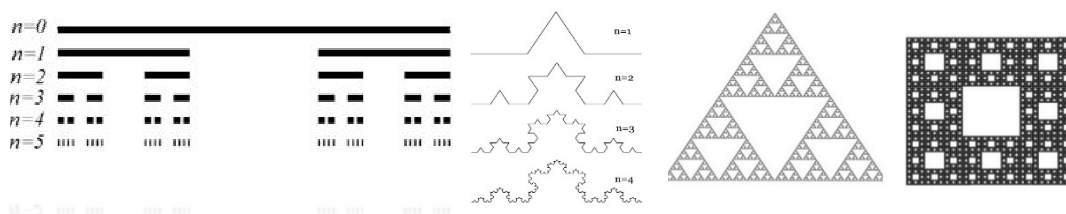


Рисунок 1 – Множество Кантора, кривая Коха, треугольник и ковер Серпинского

Кантор с помощью простой рекурсивной процедуры «превратил» линию в набор несвязных точек: брал линию и выносил её центральную треть на определенное расстояние, затем повторял эту процедуру с остальными отрезками. Множество Кантора — нигде не плотное несчётное множество.

Кривая Коха — не самопересекающаяся непрерывная кривая бесконечной длины, не имеющая касательной ни в одной точке. Она может быть использована для описания движения броуновских частиц.

Треугольник Серпинского («скатерть») и ковёр Серпинского — аналоги множества Кантора на плоскости. Этот фрактал описал в 1915 году польский математик Вацлав Серпинский. Чтобы его получить, нужно взять (равносторонний) треугольник с внутренностью, провести в нём средние линии и выкинуть центральный из четырех образовавшихся маленьких треугольников. Дальше эти же действия нужно повторить с каждым из оставшихся трех треугольников, и т. д.

Губка Менгера — аналог множества Кантора в трёхмерном пространстве

Примеры Вейерштрасса и Ван дер Вардена нигде не дифференцируемой непрерывной функции (рисунок 2).

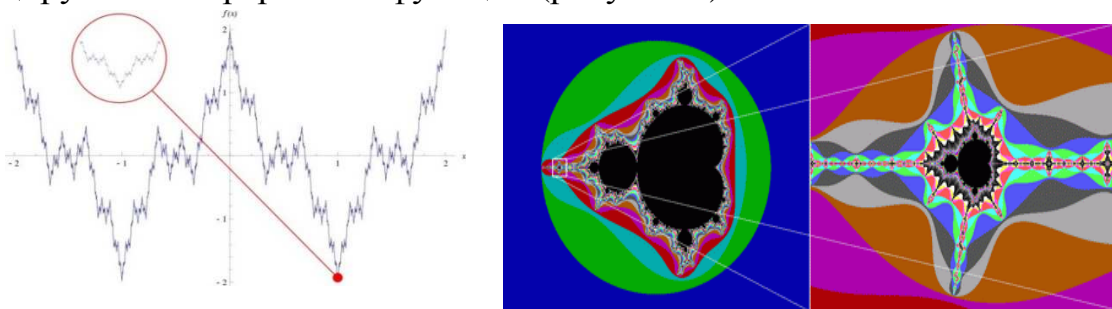


Рисунок 2 – Пример Вейерштрасса и Ван дер Вардена, множество Мандельброта визуальное построенное с применением ЭВМ

Алгебраические фракталы получили свое название за то, что их строят, используя простые алгебраические формулы с помощью нелинейных процессов в n -мерных пространствах. Примеры алгебраических фракталов: множество Мандельброта; множество Жюлиа; бассейны Ньютона; биоморфы (рисунок 2).

Множество Мандельброта впервые было построено (визуально с применением ЭВМ) Бенуа Мандельбротом весной 1980 г. в исследовательском центре фирмы IBM им. Томаса Дж. Уотсона. И хотя исследования подобных объектов начались ещё в прошлом веке, именно открытие этого множества и совершенствование аппаратных средств

машинной графики в решающей степени повлияли на развитие фрактальной геометрии и теории хаоса.

Черный цвет в середине показывает, что в этих точках функция стремится к нулю – это и есть множество Мандельброта. За пределами этого множества функция стремится к бесконечности. А самое интересное это границы множества. Они-то и являются фрактальными.

Многие объекты в природе обладают свойствами фрактала. Например, побережья, облака, кроны деревьев, снежинки, система кровообращения, альвеолы, суккуленты. Лучше всего фракталы можно разглядеть в сорте капусты под названием Романеско — гибриде цветной капусты и брокколи.

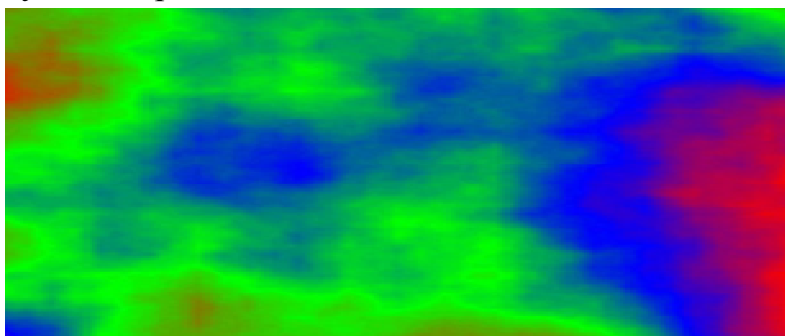


Рисунок 3 - Плазма

Типичный представитель третьего класса фракталов «Плазма» (рисунок 3). Для ее построения возьмем прямоугольник и для каждого его угла определим цвет. Далее находим центральную точку прямоугольника и раскрашиваем ее в цвет равный среднему арифметическому цветов по углам прямоугольника плюс некоторое случайное число. Чем больше случайное число, тем более «рваным» будет рисунок.

Для генерации фракталов существуют программы, например: FractalExplorer, UltraFRACTAL, ChaosPro и др., а также онлайн генераторы NadinDesign, Newart.ru и др. Фрактальное изображение строится на основе выбранной заготовки, определяемой системой параметрических уравнений, в которых параметры несложно изменить по своему желанию. Меняя функцию, условия выхода из цикла, можно получать разные фракталы. Один из способов получить фрактальный узор — воспользоваться онлайн-векторным редактором Тоби Ша мана. В основе инструментария этого графического редактора лежит все тот же принцип самоподобия.

Мы взяли четырехугольник и круг. Их можно добавлять на холст, масштабировать и вращать. Перекрываясь по принципу сложения, эти элементы образуют новые, менее тривиальные формы. Далее эти новые формы можно добавлять в проект, а программа будет по-

вторять генерирование этих изображений до бесконечности. Нам удалось получить следующие фракталы – рисунок 4.

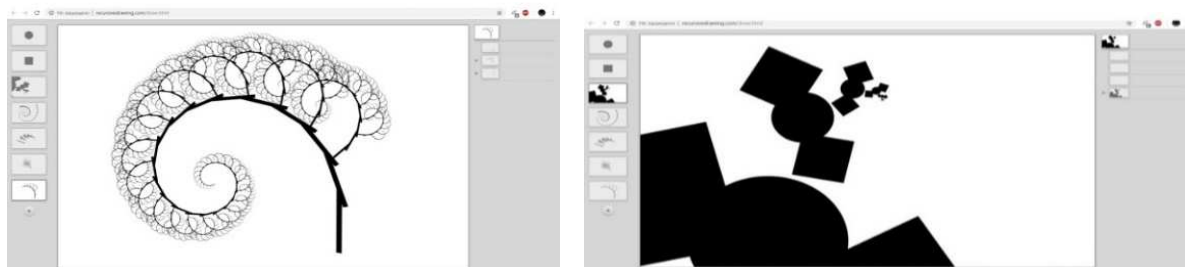


Рисунок 4 – Полученные фракталы

Существуют программы, которые озвучивают фрактальные множества, превращая их в электронную музыку.

Фракталы находят всё большее применение в науке. Наиболее полезным использованием фракталов в компьютерной науке является фрактальное сжатие данных, например, можно смоделировать языки пламени; пористые материалы хорошо представляются во фрактальной форме в связи с тем, что они имеют очень сложную геометрию. Земля – классический пример фрактального объекта.

В настоящее время фракталы стремительно вторгаются во многие области физики, биологии, медицины, социологии, экономики. Методы обработки изображений и распознавания образов, использующие новые понятия, дают возможность исследователям применить этот математический аппарат для количественного описания огромного количества природных объектов и структур.

ЛИТЕРАТУРА

1. Онлайнный векторный редактор Тоби Ша'мана. (<http://recursivedrawing.com/>)
2. А. А. Кириллов. Повесть о двух фракталах. — Летняя школа «Современная математика». — Дубна, 2007. (<https://www.litres.ru/aleksandr-aleksandrovich-kirillov/povest-o-dvuh-fraktalah-uchebnoe-posobie-11647920/>)
3. Мандельброт Б. Фрактальная геометрия природы. — М.: «Институт компьютерных исследований», 2002. (http://sernam.ru/book_fract.php)
4. Пайтген Х.- О., Рихтер П. Х. Красота фракталов. Образы комплексных динамических систем. — М.: «Мир», 1993. (<https://www.twirpx.com/file/222381/>)
5. Федер Е. Фракталы. — М: «Мир», 1991. (<https://www.twirpx.com/file/18004/>)

УДК 519.254

Студ. А. Т. Черенков
 Науч. рук. доц. Е. И. Ловенецкая
 (кафедра высшей математики, БГТУ)

МЕТОД ШТРАССЕНА

Фолькен Штрассен — немецкий математик, почетный профессор кафедры математики и статистики Констанцского университета. Родился 29 апреля 1936 г. в Дюссельдорфе (Германия). Занимался анализом и разработкой быстрых алгоритмов, один из создателей теории алгебраической сложности, в которой ему принадлежат многие классические теоремы.

мощностей не было, существовала проблема скорости и возможности выполнения подсчётов больших объёмов чисел. Чтобы решить эту проблему, учёные старались оптимизировать счёт при помощи упрощения, ускорения, замены алгоритмов. Алгоритм Штрассена стал альтернативой для обыкновенного умножения квадратных матриц, давая преимущество в скорости для матриц большого размера. Если обычное умножение двух матриц размера $N \times N$ производится за время $O(N^3)$, то метод Штрассена требует $O(N^{2,81})$. Поэтому метод применяется в случаях, когда вычислительные мощности могут быть ограничены, существуют большие матрицы, требуется высокая скорость вычислений.

Чтобы произвести обыкновенное умножение матриц A и B размера 2×2 , необходимо выполнить 8 умножений. Метод Штрассена позволяет выполнить данную операцию за 7 умножений.

Переходя к особенностям метода, стоит отметить размер матриц. Метод Штрассена работает с квадратными матрицами, порядок которых можно представить в виде числа, равного степени двойки. В случае, когда это не так, матрица дополняется нулевыми элементами до квадратной матрицы ближайшей корректного размера:

$$\begin{pmatrix} 1 & -9 & 2 \\ 3 & 7 & 4 \\ 8 & -2 & 5 \end{pmatrix}, N = 3; \quad \begin{pmatrix} 1 & -9 & 2 & 0 \\ 3 & 7 & 4 & 0 \\ 8 & -2 & 5 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, N = 4;$$

Теперь можно приступить к самому алгоритму. Как было указано ранее, метод имеет 7 умножений. Ценой упразднения одного умножения является подсчёт дополнительных сумм и разностей.

Алгоритм действий [1]:

1. Разделить входные матрицы A и B и выходную матрицу C на подматрицы размера $N/2 \times N/2$.

2. Создать 10 матриц S , каждая из которых имеет размер $N/2 \times N/2$ и представляет собой сумму или разность двух матриц, созданных на шаге 1.

3. Используя подматрицы, созданные на шаге 1, и 10 матриц S , рекурсивно вычислить 7 матричных произведений P .

4. Вычислить подматрицы $C_{11}, C_{12}, C_{21}, C_{22}$ результирующей матрицы C .

Рассмотрим работу алгоритма на примере квадратных матриц размера 4×4 . Исходные матрицы

$$A = \begin{pmatrix} 4 & 1 & 8 & 3 \\ 6 & 2 & 1 & 9 \\ 3 & 2 & 1 & 3 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 & 3 & 2 \\ 4 & 6 & 8 & 0 \\ 2 & 0 & 9 & 1 \\ 3 & 4 & 1 & 6 \end{pmatrix}$$

и их произведение $C = A \cdot B$ разделим на четыре блока — подматрицы размера $N/2 \times N/2$, т. е. в данном случае размера 2×2 :

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}; B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}; C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}.$$

В нашем примере:

$$A_{11} = \begin{pmatrix} 4 & 1 \\ 6 & 2 \end{pmatrix}; A_{12} = \begin{pmatrix} 8 & 3 \\ 1 & 9 \end{pmatrix}; A_{21} = \begin{pmatrix} 3 & 2 \\ 0 & 0 \end{pmatrix}; A_{22} = \begin{pmatrix} 1 & 3 \\ 1 & 0 \end{pmatrix};$$

$$B_{11} = \begin{pmatrix} 1 & 0 \\ 4 & 6 \end{pmatrix}; B_{12} = \begin{pmatrix} 3 & 2 \\ 8 & 0 \end{pmatrix}; B_{21} = \begin{pmatrix} 2 & 0 \\ 3 & 4 \end{pmatrix}; B_{22} = \begin{pmatrix} 9 & 1 \\ 1 & 6 \end{pmatrix}.$$

На втором шаге алгоритма создадим 10 матриц S размера $N/2 \times N/2$. Каждая матрица S находится путём сложения или вычитания матриц, полученных ранее. Матрицы S вычислим по формулам:

$$S_1 = B_{12} - B_{22}; S_2 = A_{11} + A_{12};$$

$$S_3 = A_{21} + A_{22}; S_4 = B_{21} - B_{11}; S_5 = A_{11} + A_{22}; S_6 = B_{11} + B_{22}; S_7 = A_{12} - A_{22}; S_8 = B_{21} + B_{22}$$

$$; S_9 = A_{11} - A_{21}; S_{10} = B_{11} + B_{12}.$$

Подставляя необходимые подматрицы в формулы, получим:

$$S_1 = \begin{pmatrix} -6 & 1 \\ 7 & -6 \end{pmatrix}; S_2 = \begin{pmatrix} 12 & 4 \\ 7 & 11 \end{pmatrix}; S_3 = \begin{pmatrix} 4 & 5 \\ 1 & 0 \end{pmatrix}; S_4 = \begin{pmatrix} 1 & 0 \\ -1 & -2 \end{pmatrix};$$

$$S_5 = \begin{pmatrix} -5 & 4 \\ 7 & 2 \end{pmatrix}; S_6 = \begin{pmatrix} 10 & 1 \\ 5 & 12 \end{pmatrix}; S_7 = \begin{pmatrix} 7 & 0 \\ 0 & 9 \end{pmatrix}; S_8 = \begin{pmatrix} 11 & 1 \\ 4 & 10 \end{pmatrix};$$

$$S_9 = \begin{pmatrix} 1 & -1 \\ 6 & 2 \end{pmatrix}; S_{10} = \begin{pmatrix} 4 & 2 \\ 12 & 6 \end{pmatrix}.$$

Далее, используя подматрицы, созданные на шаге 1, и 10 матриц S , рекурсивно вычислим 7 матричных произведений P с помощью следующих формул:

$$\begin{aligned} P_1 &= A_{11} \cdot S_1 = A_{11} \cdot B_{12} - A_{11} \cdot B_{22}; \\ P_2 &= S_2 \cdot B_{22} = A_{11} \cdot B_{22} + A_{12} \cdot B_{22}; \\ P_3 &= S_3 \cdot B_{11} = A_{21} \cdot B_{11} - A_{22} \cdot B_{11}; \\ P_4 &= A_{22} \cdot S_4 = A_{22} \cdot B_{21} - A_{22} \cdot B_{11}; \\ P_5 &= S_5 \cdot S_6 = A_{11} \cdot B_{11} - A_{11} \cdot B_{22} + A_{22} \cdot B_{11} + A_{22} \cdot B_{22}; \\ P_6 &= S_7 \cdot S_8 = A_{12} \cdot B_{21} - A_{12} \cdot B_{22} - A_{22} \cdot B_{21} - A_{22} \cdot B_{22}; \\ P_7 &= S_9 \cdot S_{10} = A_{11} \cdot B_{11} - A_{11} \cdot B_{12} - A_{21} \cdot B_{11} - A_{21} \cdot B_{12}. \end{aligned}$$

В результате получаем:

$$\begin{aligned} P_1 &= \begin{pmatrix} -17 & -2 \\ -22 & -6 \end{pmatrix}; \quad P_2 = \begin{pmatrix} 112 & 36 \\ 74 & 73 \end{pmatrix}; \quad P_3 = \begin{pmatrix} 24 & 30 \\ 1 & 0 \end{pmatrix}; \quad P_4 = \begin{pmatrix} -2 & -6 \\ 1 & 0 \end{pmatrix}; \\ P_5 &= \begin{pmatrix} 70 & 53 \\ 80 & 31 \end{pmatrix}; \quad P_6 = \begin{pmatrix} -8 & -4 \\ 48 & 24 \end{pmatrix}; \quad P_7 = \begin{pmatrix} 77 & 7 \\ 36 & 90 \end{pmatrix}. \end{aligned}$$

Вычислим подматрицы $C_{11}, C_{12}, C_{21}, C_{22}$ результирующей матрицы C , используя формулы:

$$\begin{aligned} C_{11} &= P_5 + P_4 - P_2 + P_6; \\ C_{12} &= P_1 + P_2; \\ C_{21} &= P_3 + P_4; \\ C_{22} &= P_5 + P_1 - P_3 - P_7. \end{aligned}$$

В итоге получим искомую матрицу $C = A \cdot B$, составив ее из блоков $C_{11}, C_{12}, C_{21}, C_{22}$:

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} = \begin{pmatrix} 33 & 12 & 95 & 34 \\ 43 & 48 & 52 & 67 \\ 22 & 24 & 37 & 25 \\ 2 & 0 & 9 & 1 \end{pmatrix}.$$

В программной реализации [1] данного метода используется рекурсия. Метод позволяет рекурсивно делить матрицу до того момента, пока её порядок не станет равным 2. Далее происходит обыкновенное умножение. Все подматрицы, в отличие от исходных матриц, хранятся в одномерном массиве, где определённый диапазон массива соответствует значениям подматриц, а позиция начала этого диапазона представима в виде номера подматрицы, умноженного на порядок входных матриц.

Публикация метода Штрассена инициировала активные поиски быстрых алгоритмов, и уже через несколько лет учёные заметно уско-

рили подсчёты. В 1990 г. американские учёные Дон Копперсмит и Шмуэль Виноград, развившие идеи Штрассена, добились оценки $O(N^{2,376})$, после чего ускорение прекратилось. На протяжении двадцати лет алгоритм Копперсмита – Винограда оставался теоретически самым быстрым, однако в реальных вычислительных задачах не использовался, так как он эффективен только на чрезвычайно больших матрицах[2].

В 2011 г. Вирджиния Уильямс продолжила рассмотрение метода Копперсмита – Винограда и нашла более быстрый способ расчёта. Ей удалось добиться оценки $O(N^{2,3728642})$ [3]. В 2014 г. Франсуа Ле Галль упростил метод Уильямс и получил новую улучшенную оценку $O(N^{2,3728639})$. Но все перечисленные методы имеют чрезвычайно трудную реализацию в программировании, поэтому чаще всего вместо доработанных методов применяется алгоритм Штрассена.

Не смотря на рост мощностей вычислительной техники, метод Штрассена остаётся актуальным и сегодня, поскольку параллельно с усовершенствованием средств вычисления значительно возрастают размеры обрабатываемых данных.

ЛИТЕРАТУРА

1. Кормен Т. Алгоритмы: построение и анализ, 3-е изд. : Пер. с англ. / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. — М.: ООО «И. Д. Вильямс», 2013. — 1328 с.

2. Найден максимально быстрый способ умножения матриц [Электронный ресурс]. Режимдоступа:

<http://www.nanonewsnet.ru/news/2012/naiden-maksimalno-bystryi-sposob-umnozheniya-matritys>. Дата доступа: 23.04.2019.

3. Алгоритм Копперсмита – Винограда [Электронный ресурс].
— Режим доступа: https://ru.wikipedia.org/wiki/Алгоритм_Копперсмита_–_Винограда. Дата доступа: 22.04.2019.

УДК 378.6:519.83

Студ. М. Е. Алексеев
 Науч. рук. доц. И. К. Асмыкович
 (кафедра высшей математики, БГТУ)

ШИФРОВАНИЕ МЕТОДОМ ГАММИРОВАНИЯ

Цель данной работы: рассмотреть алгоритм шифрования методом гаммирования по модулю 2 и его преимущества по сравнению с другими алгоритмами.

Гаммирование – метод последовательного симметричного шифрования, суть которого состоит в том, что символы шифруемого текста последовательно складываются с символами некоторой специальной последовательности, которая называется *гаммой*. Такой метод чаще всего представляют как наложение гаммы на исходный текст, поэтому он получил название "*гаммирование*".

Процедуру наложения гаммы на исходный текст можно осуществить двумя способами. При первом способе символы исходного текста и гаммы заменяются цифровыми эквивалентами, которые складываются по модулю k :

$$R_i = (S_i + G_i) \bmod(k),$$

где k – число символов в алфавите, а R_i , S_i и G_i – символы, соответственно, зашифрованного, исходного текста и гаммы.

При втором методе символы исходного текста и гаммы представляются в виде двоичного кода, затем соответствующие разряды складываются по модулю 2:

$$R_i = (S_i + G_i) \bmod(2) = S_i \oplus G_i.$$

Можно дополнительно использовать и другие логические операции. Это будет равносильно введению еще одного ключа, которым является выбор правила формирования зашифрованного сообщения из символов исходного текста и гаммы.

Процесс дешифрования данных сводится к повторной генерации гаммы шифра при известном ключе и наложении такой гаммы на зашифрованные данные.

Операция сложения по модулю 2 в алгебре логики называется также "исключающее ИЛИ". Данная операция работает следующим образом (рис.1): при сложении двух двоичных знаков получается 0, если исходные двоичные цифры одинаковы, и 1, если цифры разные.

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Рисунок 1 – Операция сложения по модулю 2

Таким образом, при гаммировании по модулю 2 нужно использовать одну и ту же операцию как для шифрования, так и для расшифровки. Это позволяет использовать один и тот же алгоритм, а соответственно и одну и ту же программу при программной реализации, как для шифрования, так и для расшифровки.

Операция сложения по модулю 2 очень быстро выполняется на вычислительных устройствах (в отличие от многих других арифметических операций), поэтому наложение гаммы даже на очень большой открытый текст выполняется практически мгновенно.

Благодаря указанным достоинствам метод гаммирования широко применяется в современных технических системах сам по себе, а также как элемент комбинированных алгоритмов шифрования.

Гаммирование по модулю 2 в общем случае производится следующим образом:

- символы исходного текста и гамма представляются в двоичном коде исходя из выбранной кодировки и располагаются один под другим, при этом ключ (гамма) записывается столько раз, сколько необходимо;
- каждая пара двоичных знаков складывается по модулю два;
- полученная последовательность двоичных знаков кодируется символами алфавита в соответствии с выбранной ранее кодировкой.

Разберём пример шифрования слова. Символы слова будем представлять в соответствии с кодировкой Windows-1251. Ниже представлена таблица 1, которая показывает как записываются прописные буквы кириллицы в соответствии с кодировкой.

Таблица 1 – Кодировки прописных букв

Буква	Bin-код	Буква	Bin-код	Буква	Bin-код
А	1100 0000	Л	1100 1011	Ц	1101 0110
Б	1100 0001	М	1100 1100	Ч	1101 0111
В	1100 0010	Н	1100 1101	Ш	1101 1000
Г	1100 0011	О	1100 1110	Щ	1101 1001
Д	1100 0100	П	1100 1111	Ъ	1101 1010
Е	1100 0101	Р	1101 0000	Ы	1101 1011
Ж	1100 0110	С	1101 0001	Ь	1101 1100
З	1100 0111	Т	1101 0010	Э	1101 1101
И	1100 1000	У	1101 0011	Ю	1101 1110
Й	1100 1001	Ф	1101 0100	Я	1101 1111
К	1100 1010	Х	1101 0101		

Исходный текст: КОНФЕРЕНЦИЯ

Гамма (ключ): СИРЕНЬ

Исходный текст в шестнадцатеричном виде:

CA CE CD D4 C5 D0 C5 CD D6 C8 DF

Гамма в шестнадцатеричном виде: D1 C8 D0 C5 CD DC

Если гамма короче, чем сообщение, предназначенное для шифрования, гамма повторяется требуемое число раз. Так в примере длина исходного сообщения равна 11-и байтам, а длина ключа – 6-и байтам. Следовательно, для шифрования гамма должна быть повторена 1 раз полностью и еще один раз частично.

Исх. биты	1100	1010	1100	1110	1100	1101	1101	0100	1100	0101	1101	0000
Гамма	1101	0001	1100	1000	1101	0000	1100	0101	1100	1101	1101	1100
Результат	0001	1011	0000	0110	0001	1101	0001	0001	0000	1000	0000	1100

Исх. биты	1100	0101	1100	1101	1101	0110	1100	1000	1101	1111
Гамма	1101	0001	1100	1000	1101	0000	1100	0101	1100	1101
Результат	0001	0100	0000	0101	0000	0110	0000	1101	0001	0010

Закодированный текст в шестнадцатеричном виде:

1B 06 1D 11 08 0C 16 05 06 0D 12

Исх. биты	0001	1011	0000	0110	0001	1101	0001	0001	0000	1000	0000	1100
Гамма	1101	0001	1100	1000	1101	0000	1100	0101	1100	1101	1101	1100
Результат	1100	1010	1100	1110	1100	1101	1101	0100	1100	0101	1101	0000

Исх. биты	0001	0100	0000	0101	0000	0110	0000	1101	0001	0010
Гамма	1101	0001	1100	1000	1101	0000	1100	0101	1100	1101
Результат	1100	0101	1100	1101	1101	0110	1100	1000	1101	1111

Расшифрованный текст в шестнадцатеричном виде:

CA CE CD D4 C5 D0 C5 CD D6 C8 DF

Стойкость шифрования методом гаммирования определяется главным образом свойствами гаммы – длительностью периода и равномерностью статистических характеристик. Последнее свойство обеспечивает отсутствие закономерностей в появлении различных символов в пределах периода. Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, если гамма шифра не содержит повторяющихся битовых последовательностей.

Обычно разделяют две разновидности гаммирования – с конечной и бесконечной гаммами. При хороших статистических свойствах гаммы стойкость шифрования определяется только длиной периода гаммы. При этом, если длина периода гаммы превышает длину шиф-

руемого текста, то такой шифр теоретически является абсолютно стойким, т.е. его нельзя вскрыть при помощи статистической обработки зашифрованного текста, а можно раскрыть только прямым перебором (bruteforce). Криптостойкость в этом случае определяется размером ключа. Это, однако, не означает, что дешифрование такого текста вообще невозможно: при наличии некоторой дополнительной информации исходный текст может быть частично или полностью восстановлен даже при использовании бесконечной гаммы.

Кроме того, для использования ключа вначале необходимо каким-либо надежным способом доставить его обеим сторонам, обменивающимся сообщениями. Это приводит к возникновению проблемы распределения ключей, сложность решения которой возрастает с увеличением длины ключа и количества абонентов в сети передачи сообщений.

В качестве гаммы может быть использована любая последовательность случайных символов, например, последовательность цифр числа π , числа e и т.п. При шифровании с помощью ЭВМ последовательность гаммы формируется с помощью *генератора псевдослучайных чисел* (ГПСЧ). В настоящее время разработано несколько алгоритмов работы таких генераторов, которые обеспечивают удовлетворительные характеристики гаммы.

ЛИТЕРАТУРА

1. Методы гаммирования [Электронный ресурс] – Режим доступа: <https://www.intuit.ru/studies/courses/691/547/lecture/12373?page=4>. – Дата обращения: 05.04.2019.

2. Шифрование методом гаммирования [Электронный ресурс] – Режим доступа: <https://studfiles.net/preview/5470123/page:9/> – Дата обращения: 08.04.2019.

УДК 519.176

Студ. Р. Ю. Злобин
Науч. рук. доц. И. К. Асмыкович
(кафедра высшей математики, БГТУ)

НЕКОТОРЫЕ ПРИМЕНЕНИЯ ТЕОРИИ ГРАФОВ

Графы и теория графов представляют собой один из наиболее важных и интересных, но в тоже время один из сложных разделов в математике и информатике.

Многие структуры, представляющие практический интерес в логике, информатике, математике и других науках, могут быть представлены графами.

Теория графов находит применение, например, в геоинформационных системах (ГИС), химии, экономике, логистике, схемотехнике. Применение различных вычислений, производимых на графах, позволяет, например, найти кратчайший объездной путь или спланировать оптимальный маршрут.

Цель работы: описать некоторые задачи теории графов и алгоритмы их решения.

Граф G – это упорядоченная пара $G := (V, E)$, где V – это непустое множество вершин или узлов, а E – множество пар вершин, называемых рёбрами [1].

Вершины и рёбра графа называются также элементами графа, число вершин в графе — порядком, число рёбер — размером графа.

Граф может быть ориентированным или неориентированным. В ориентированном графе, связи являются направленными (то есть пары в множестве E являются упорядоченными, например, пары (2,5) и (5,2) это разные связи). В неориентированном графе, связи ненаправленные, и поэтому если существует связь (2,5), то значит, что и связь (5,2) существует.

При изображении графов на рисунках чаще всего используется следующая система обозначений: вершины графа изображаются точками или, при конкретизации смысла вершины, прямоугольниками, овалами и др., где внутри фигуры раскрывается смысл вершины (графы блок-схем алгоритмов). Если между вершинами существует ребро, то соответствующие точки (фигуры) соединяются линией или дугой. В случае ориентированного графа дуги заменяют стрелками, они явно указывают направленность ребра. Иногда рядом с ребром размещают поясняющие надписи, раскрывающие смысл ребра, например, в графах переходов конечных автоматов.

При создании программ для работы с графами большую значимость имеет метод их представления в памяти компьютера. Есть два основных метода представления графа: в виде списке смежности и в виде матрицы смежности.

Матрица смежности представляет собой квадратную матрицу A размером $|V| \times |V|$, где V – это число вершин в графе. Каждый элемент матрицы A несёт в себе информацию о рёбрах графа, например, наличие или отсутствие ребра из вершины i в j (обычно обозначается 1 или 0).

При списках смежности используется массив A , содержащий $|V|$ списков, где V – это число вершин в графе. Список $A[i]$ содержит такие вершины u , что между i и u есть ребро.

Каждый из данных способов имеет свои преимущества и недостатки. Выбор каждого из них зависит от типа задачи, вида исходных данных, предполагаемого решения и т.д.

В интернете, а также в специализированной литературе существует большое множество задач, решение которых представляется в виде графа.

Одной из таких задач является поиск кратчайших расстояний от одной вершины ко всем остальным.

Имеется оргграф. Необходимо найти кратчайшие расстояния от вершины 1 до остальных, построить дерево кратчайших путей.

Для решения задачи воспользуемся алгоритмом Дейкстры [3]. Он заключается в том, что вершинам графа присваиваются временные метки d_i , которые меняются по определённым правилам. У всех меток вершин, кроме стартовой, принимают значение ∞ . У стартовой метки значение 0.

Затем, до тех пор, пока минимальное значение меток не равно ∞ или не прошло n итераций (n – число вершин графа), производим релаксации из вершины с минимальной меткой, т.е. если i – индекс вершины с минимальной меткой, а j – индекс сопряжённой вершины с вершиной i , то $d_j = \min(d_j, d_i + a_{i,j})$, где $a_{i,j}$ – вес ребра между вершинами i и j .

Для построения дерева кратчайших путей, запоминаем для каждой вершины индекс вершины p_i , из которой проводилась её последняя релаксация. После в исходном графе для каждой вершины оставляем только то ребро, которые идёт в неё из p_i .

Ещё одной типовой задачей является, получение дерева минимального веса.

Дан взвешенный связный неориентированный граф. Необходимо удалить рёбра графа, так что бы получилось дерево, причём сумма весов его рёбер должна быть минимальна. Называется минимальным остовом.

Обычно данную задачу решают с помощью алгоритмов Прима и Краскала.

Алгоритм Прима. Искомое дерево строится постепенно, добавляют в него рёбра по одному. Изначально остов состоит из одной вершины (её можно выбрать произвольно). Затем выбирается ребро минимального веса, исходящее из этой вершины, и добавляется в остов. После этого остов содержит уже две вершины, и теперь ищут и добавляют ребро минимального веса, имеющее один конец в одной из двух выбранных вершин, а другой — наоборот, во всех остальных, кроме этих двух. И так далее, т.е. всякий раз ищется минимальное по

весу ребро, один конец которого — уже взятая в остов вершина, а другой конец — ещё не взятая, и это ребро добавляется в остов (если таких рёбер несколько, можно взять любое). Этот процесс повторяется до тех пор, пока остов не станет содержать все вершины (или, что то же самое, $n - 1$ ребро).

В итоге будет построен остов, являющийся минимальным. Если граф был изначально не связан, то остов найден не будет.

Алгоритм Краскала. Изначально помещают каждую вершину в своё дерево, а затем постепенно объединяют эти деревья, объединяя на каждой итерации два некоторых дерева ребром. Перед началом выполнения алгоритма, все рёбра сортируются по весу (в порядке возрастания). Затем начинается процесс объединения: перебираются все рёбра от первого до последнего (в порядке сортировки), и если у текущего ребра его концы принадлежат разным поддеревьям, то эти поддеревья объединяются, а ребро добавляется к ответу.

В итоге все вершины окажутся в одном поддереве, которое и образует минимальный остов.

Графы являются универсальным средством, которое помогает нам как при решении задач в различных практических сферах. Умение оперировать ими, необходимых навык востребованного специалиста.

ЛИТЕРАТУРА

1. Граф (математика) Wikipedia [Электронный ресурс]. – Режим доступа:
[https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_\(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0\)](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)). – Дата доступа: 29.03.2019.
2. Оре О. Теория графов. — 2-е изд. — М.: Наука, 1980. — С. 336.
3. Иллюстративное введение в теорию графов и её применение Proglib [Электронный ресурс]. – Режим доступа:
<https://proglib.io/p/graph-theory/>– Дата доступа: 29.03.2019.

Студ. А.Д. Радчиков
 Науч. рук. доц. И.К. Асмыкович
 (кафедра высшей математики)

СРАВНЕНИЕ СКОРОСТИ СХОДИМОСТИ ЧАСТИЧНЫХ СУММ РЯДА ФУРЬЕ

Ряд Фурье позволяет представить непрерывную функцию $f(x)$, определенную на некотором отрезке $[0, T]$, в виде суммы бесконечного числа (бесконечного ряда) гармонических функций (синусоид и/или косинусоид) с определёнными амплитудами и фазами, также рассматриваемых на отрезке $[0, T]$. То есть в следующем виде:

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} A_k \cos\left(2\pi \frac{k}{\tau} x + \theta_k\right)$$

где A_k — амплитуда k -го гармонического колебания (функции \cos);
 $2\pi \frac{k}{\tau} x = k\omega$ — круговая частота гармонического колебания; θ_k — начальная фаза k -го колебания.

Ряд Фурье позволяет изучать периодические (непериодические) функции [1]. Например, если дана функция, интегрируемая на отрезке $[-l, l]$, то её можно разложить в тригонометрический ряд Фурье:

$$f(x) \sim \frac{a_0}{2} + \sum_{n=1}^{+\infty} a_n \cos \frac{\pi n x}{l} + b_n \sin \frac{\pi n x}{l}$$

где a_0, a_n, b_n — коэффициенты ряда Фурье, которые находятся по следующим формулам:

$$a_0 = \frac{1}{l} \cdot \int_{-l}^l f(x) dx, \quad a_n = \frac{1}{l} \cdot \int_{-l}^l f(x) \cdot \cos \frac{\pi n x}{l} dx, \quad b_n = \frac{1}{l} \cdot \int_{-l}^l f(x) \cdot \sin \frac{\pi n x}{l} dx.$$

Цель работы: изучить зависимость точности приближения периодических функций частичными суммами ряда Фурье.

Например, для функции $f(x) = x + 1$, заданной на отрезке $[-\pi, \pi]$, тригонометрический ряд Фурье будет иметь следующий вид:

$$f(x) \sim 1 - 2 \sum_{n=1}^{+\infty} \left(\frac{(-1)^n \cdot \sin nx}{n} \right).$$

Разложение функций в сходящийся ряд Фурье нашло практическое применение во многих инженерных расчётах: переменные токи и напряжения, смещения, скорость и ускорение кривошипно-шатунных механизмов и акустические волны. Оно также является математической основой спектрального анализа периодических сигналов.

Частичная сумма ряда Фурье $S_N(x)$ функции $f(x)$, заданной на отрезке $[-l, l]$ определяется выражением:

$$S_N(x) = \frac{a_0}{2} + \sum_{k=1}^N a_n \cos \frac{\pi kx}{l} + b_n \sin \frac{\pi kx}{l}.$$

Так для периодической функции с периодом 2π , заданной на отрезке $[-\pi, \pi]$ $f(x) = x + 1$, графики частичных сумм S_2 и S_{10} имеют следующий вид:

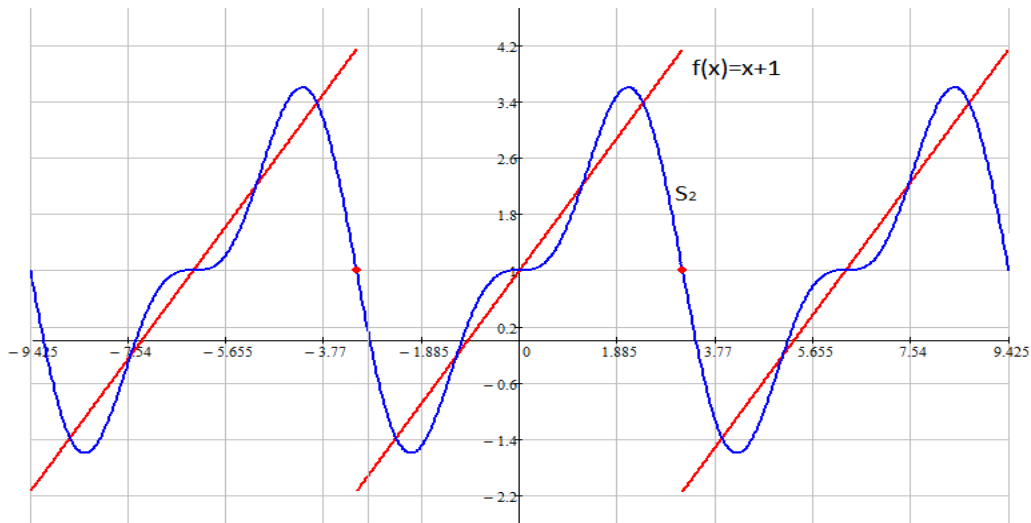


Рисунок 1 – Графики функции и частичной суммы S_2

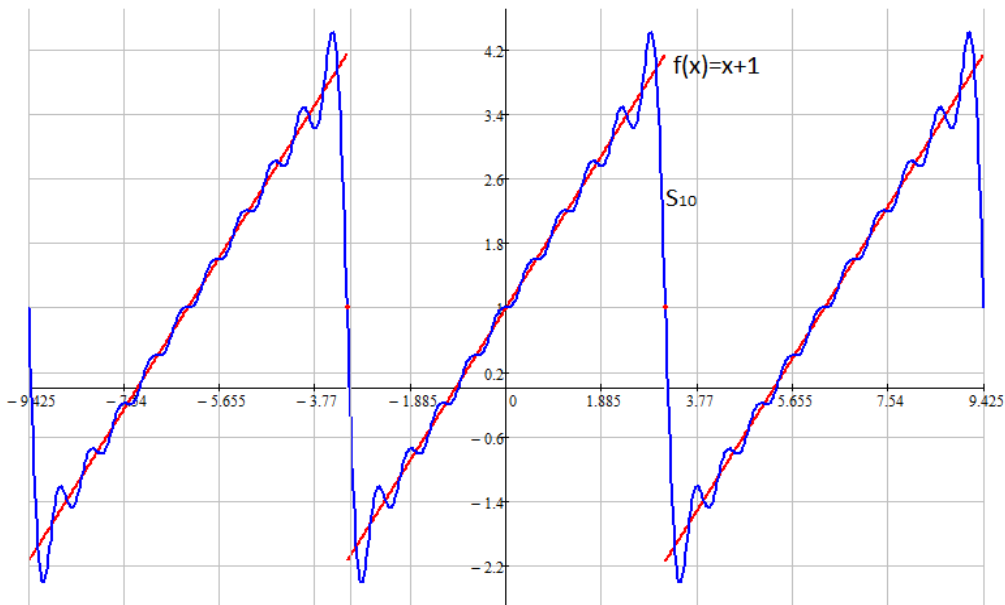


Рисунок 2 – Графики функции и частичной суммы S_{10}

Анализируя графики, можно заметить, что периоды гармонических составляющих кратны длине отрезка $[0, T]$, на котором определена исходная функция $f(x)$ и что при устремлении числа слагаемых

частичной суммы к бесконечности, среднеквадратическое отклонение от функции стремиться к нулю:

$$\lim_{N \rightarrow \infty} \frac{1}{2\pi} \int_{-l}^l (f(x) - S_N(x))^2 dx = 0.$$

Поэтому стоит обратить внимание на исследование частичных сумм. А именно, при каком порядке частичной суммы, значение в определенной точке будет отклонено от значения функции с определённой точностью. И наоборот, как будет отклонено значение частичной суммы порядка k от функции в определённой точке, при случайно заданном k .

Ниже приведены расчеты определения порядка частичной суммы ряда Фурье, при котором значение частичной суммы отклоняется от значения функции $f(x) = x$, определенной на отрезке $[-\pi, \pi]$, в точке $x = 2.5$ с определённой точностью.

Таблица – Порядки частичных сумм

Точность	Количество слагаемых, n
0.1	7
0.01	41
0.001	95
0.0001	237
1E-5	3219
1E-6	20969
1E-7	45251
1E-8	51215
1E-9	51925
1E-10	364756

В данной работе на основании ряда численных методов [2] разработаны программы, позволяющие дать точечную оценку и оценку среднеквадратического отклонения между частичной суммой произвольного порядка и раскладываемой функции. А также алгоритмы нахождения порядка частичных сумм, по заданному соответствующему отклонению. Проведены численные эксперименты для непрерывно дифференцируемых функций, кусочно-непрерывных функций и кусочно-монотонных функций. Получено, что число слагаемых, обеспечивающих требуемую точность значения частичной суммы, существенно зависит от гладкости разлагаемой функции.

ЛИТЕРАТУРА

1. Письменный Д. Конспект лекций по высшей математике. – Москва. – 2007. – 608с.
2. Колдаев В.Д. Численные методы и программирование. – Москва. – 2009. – 335с.

Студ. К.С. Марчук
Науч. рук. доц. И.К. Асмыкович
(кафедра высшей математики)

ИСПОЛЬЗОВАНИЕ ТЕОРИИ ГРУПП ТОЧЕК НА ЭЛЛИПТИЧЕСКОЙ КРИВОЙ ДЛЯ СОЗДАНИЯ ЭЛЕКТРОННОЙ ПОДПИСИ

Долгое время люди копили информацию. Сначала это были рисунки на стенах, после – печатный или письменный текст, а в наше время мы имеем огромное количество информации в электронном виде. Ни для кого не секрет, что большая часть информации имеет ценность, особенно для юридических лиц. В нашем примере, информация нуждается в проверке принадлежности – подписи.

ECDSA (Elliptic Curve Digital Signature Algorithm) – криптографический алгоритм с открытым ключом для создания цифровой подписи, определённый в группе точек эллиптической кривой [1]. Подпись создается секретно, но может быть публично проверена. Это означает, что только один субъект может создать подпись сообщения, но любой может проверить её корректность.

Цель работы: на конкретных примерах подробно рассмотреть алгоритм создания электронной подписи и способов ее публичной проверки.

До того, как ECC стала популярной, почти все алгоритмы с открытым ключом основывались на RSA, DSA и DH [2], альтернативных криптосистемах на основе модулярной арифметики. Они по-прежнему популярны, и часто используются вместе с ECC. Однако основы ECC всё ещё являются для большинства людей загадкой.

Секретные ключи и открытые ключи. К примеру, в Биткойне [3] используется вариант эллиптической криптографии **secp256k1**. Покажем, откуда берутся секретные и открытые ключи и как они связаны друг с другом. В ECDSA секретный ключ – это случайное целое число между 1 и значением порядка – количеством элементов группы. Открытый же ключ получается из секретного при помощи операции скалярного умножения базовой точки на значение секретного ключа. В виде уравнения:

Открытый ключ = секретный ключ * базовая точка.

Это показывает, что максимально возможное количество секретных ключей - **конечно**, и равно порядку. Так может, они когда-то закончатся? Вряд ли, потому что порядок – это **действительно большое** число.

Вычисление открытого ключа разбивается на ряд операций удвоения и сложения точек, начиная с базовой точки [1]. Сложение точек $p + q$ определяется покомпонентно следующим образом:

$$c = (q_y - p_y) / (q_x - p_x), \quad r_x = c^2 - p_x - q_x, \quad r_y = c(p_x - r_x) - p_y.$$

А операция «удвоения» точки p выглядит следующим образом:

$$c = (3p_x^2 + a) / 2p_y, \quad r_x = c^2 - 2p_x, \quad r_y = c(p_x - r_x) - p_y.$$

Объем вычислений на реальном примере был бы невероятно сложным, но мы можем попробовать на примере с небольшими числами, чтобы увидеть, как это работает.

Итак, возьмем уравнение кривой Биткойна, будем использовать маленькие числа. Уравнение кривой: $y^2 = x^3 + 7$. Модуль: 67. Базовая точка: (16, 4). Порядок: 79.

Требуется выбрать секретный ключ из 79 доступных. Возьмём 3. Найдём к нему публичный ключ. Нам для этого потребуется операция удвоения и сложения:

$$c = (3 * 16^2 + 0) / (2 * 4) \bmod 67 \rightarrow c = 768 / 8 \bmod 67 \rightarrow 31 / 8 \bmod 67.$$

Но как нам выполнить операцию деления в контексте конечного поля, где результат должен быть целочисленным? Для этого мы должны умножить 31 на величину, обратную к 8. Это должно быть такое число, при умножении на которое, мы получаем остаток 1 по заданному модулю, т. е. $8^{-1} = 42$.

Далее все довольно просто:

$$c = 31 * 42 \bmod 67 = 29, \quad r_x = (29^2 - 2 * 16) \bmod 67 = 809 \bmod 67 = 5,$$

$$r_y = (29 * (16 - 5) - 4) \bmod 67 = 315 \bmod 67 = 47.$$

Далее сложим эту точку с базовой $p = (16, 4)$, $q = (5, 47)$:

$$c = (47 - 4) / (5 - 16) \bmod 67 = 43 / 56 \bmod 67 = 43 * 6 \bmod 67 = 57,$$

$$r_x = 3249 - 16 - 5 \bmod 67 = 12,$$

$$r_y = 57 * (16 - 12) - 4 \bmod 67 = 224 \bmod 67 = 23.$$

Итак, мы определили, что для секретного ключа 3 публичным ключом будет **точка** (12, 23).

Получить публичный ключ, имея секретный в разы проще, чем получить секретный, имея публичный. Хотя получить секретный ключ из публичного **теоретически** и возможно, это является вычислительно невозможным из-за огромных чисел-параметров, используемых в эллиптической криптографии.

Как и секретный ключ, публичный ключ тоже представляется в виде шестнадцатеричного числа. Однако только что мы получили точку, а не число. Так вот, публичный ключ – это просто записанные вместе два 256-битных числа, являющиеся его x и y координатами.

Подпись данных секретным ключом. Теперь у нас есть секретный и публичный ключ, которые мы можем использовать для подписи данных. Сами данные могут иметь любую длину. Для начала данные хэшируются, чтобы получить уникальное число, содержащее такое же количество битов (256), как и порядок кривой. Упрощая нашу задачу, мы пропустим шаг хеширования и подпишем данные z . Обозначим через G базовую точку, через n – порядок, а d – закрытый ключ. Алгоритм создания подписи выглядит следующим образом:

1. Выбрать некоторое целое k в пределах от 1 до $n-1$.
2. Рассчитать точку $(x, y) = k * G$, используя скалярное умножение.
3. Найти $r = x \bmod n$. Если $r = 0$, вернуться к шагу 1.
4. Найти $s = (z + r * D) / k \bmod n$. Если $S = 0$, вернуться к шагу 1.
5. Пара (r, s) является искомой подписью.

Напомним, вместо деления числителя на знаменатель, мы числитель умножаем на обратную знаменателю величину. На шаге 1 важно, чтобы k не повторялось в разных подписях, и чтобы его не могла угадать третья сторона [2]. То есть k должен быть либо случайным, либо создан детерминированным процессом, который хранится в тайне. Иначе третья сторона получает возможность найти секретный ключ, начиная с шага 4, так как s, z, r, k и n всем известны.

Давайте выберем в качестве наших данных число 21 и подпишем его секретным ключом 2.

$$z = 17 \text{ (данные)}, n = 79 \text{ (порядок)}, G = (16, 4) \text{ (базовая точка)}, \\ d = 3 \text{ (секретный ключ)}.$$

1. Выберем случайное число:

$$k = \text{rand}(1, n - 1) \rightarrow k = \text{rand}(1, 79 - 1) \rightarrow k = 2.$$

2. Рассчитаем точку. Это делается таким же образом, как ранее при вычислении публичного ключа – для краткости опустим подробную арифметику сложения и удвоения.

$$(x, y) = 2G = (16, 4) + (16, 4) = (5, 47) \rightarrow x = 5, y = 47.$$

3. Находим $r: r = x \bmod n \rightarrow r = 5 \bmod 79 = 5$.
4. Находим s :

$$s = (z + r * d) / k \bmod n \rightarrow s = (21 + 5 * 3) / 2 \bmod 79 = 36 / 2 \bmod 79 = 18 \bmod 79 = 18.$$

Обратите внимание, что выше мы смогли разделить на 2, так как результат был целым числом. Если искать обратную величину, результат будет таким же $36 * 40 \bmod 79 = 1440 \bmod 79 = 18$.

5. Теперь наша подпись – это пара $(r, s) = (5, 18)$. Как и секретные и публичные ключи, подпись обычно представляется в виде шестнадцатеричной строки.

Проверка подписи публичным ключом. Теперь у нас есть данные и подпись. Третья сторона, которая знает наш публичный ключ, может получить наши данные и подпись, и убедиться, что именно мы являемся отправителями. Обозначив наш открытый ключ Q , шаги для проверки подписи будут следующими:

1. Убедиться, что r и s находятся в диапазоне от 1 до $n-1$.
2. Рассчитать $w = s^{-1} \bmod n$.
3. Рассчитать $u = (z * w) \bmod n$.
4. Рассчитать $v = (r * w) \bmod n$.
5. Рассчитать точку $(x, y) = uG + vQ$.
6. Убедиться, что $r = x \bmod n$. Если это не так, то подпись недействительна.

На сторонних ресурсах можно найти довольно большое доказательство того, что данный алгоритм проверки работает. Но мы просто прогоним алгоритм самостоятельно и покажем, что он работает. Напомним, наши переменные:

$$z = 21 \text{ (данные)}, (r, s) = (5, 18) \text{ (подпись)}, n = 79 \text{ (порядок)}, \\ G = (16, 4) \text{ (базовая точка)}, Q = (12, 23) \text{ (публичный ключ)}.$$

1. Убедимся, что r и s находятся в диапазоне от 1 до $n-1$.

$$r: 1 \leq 5 < 79, \quad s: 1 \leq 18 < 79.$$

2. Рассчитаем w :

$$w = s^{-1} \bmod n \rightarrow w = 18^{-1} \bmod 79 = 22.$$

3. Рассчитаем u :

$$u = (z * w) \bmod n \rightarrow u = (21 * 22) \bmod 79 = 462 \bmod 79 = 67.$$

4. Рассчитаем v :

$$v = (r * w) \bmod n \rightarrow v = (5 * 22) \bmod 79 = 110 \bmod 79 = 31.$$

5. Рассчитаем точку (x, y) :

$$(x, y) = uG + vQ.$$

Разберем операции удвоения и сложения в uG и vQ отдельно.

$$uG = 67G = 2(2(16G)+G)+G = 2(2(2(2(2G))))+G)+G$$

Таким разложением мы заменяем 67 сложений точек шестью удвоениями и двумя сложениями. Данная группировка делает наши вычисления намного проще.

$$\begin{aligned} uG &= 2(2(2(2(2(5,47))))+G)+G = 2(2(2(2(49,2))) +G)+G = \\ &2(2(2(58,22))+G)+G = 2(2(34,7)+G)+G = 2(2(34,7)+G)+G = \\ &2((55,17)+G)+G = 2(11,20)+G = (14,65)+(16,4) = (46,40). \end{aligned}$$

И теперь, все то же для vQ :

$$\begin{aligned} vQ &= 31Q = 2(2(2(2Q+Q)+Q)+Q)+Q = 2(2(2((30,41)+Q)+Q)+Q)+Q = \\ &2(2(2(26,30)+Q)+Q)+Q = 2(2((64,39)+Q)+Q)+Q = 2(2(24,30)+Q)+Q = \\ &2((23,39)+Q)+Q = 2(2,22)+Q = (52,7)+(12,23) = (38,41) \end{aligned}$$

Посчитав, складываем их вместе:

$$(x, y) = uG + vQ \rightarrow (x, y) = (46, 40) + (38, 41) = (5, 47).$$

Сразу видно, что больше всего работы приходится на 5 шаг.

6. Наконец, убедимся, что $r = x \bmod n$

$$r = x \bmod n \rightarrow 5 \bmod 79 = 5.$$

Таким образом, мы рассмотрели достаточно новый, сложный и надёжный алгоритм для создания и проверки цифровой подписи, ныне используемый повсеместно.

ЛИТЕРАТУРА

1. Острик В. В., Цфасман М. А. Алгебраическая геометрия и теория чисел: рациональные и эллиптические кривые. – М.: МЦНМО, 2001. – С. 48.
2. Доступно о криптографии на эллиптических кривых [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/335906/>
3. Википедия, свободная энциклопедия [Электронный ресурс]. – Режим доступа: <https://www.wikipedia.org>
4. Биткойн [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Биткойн>
5. «Что такое биткойн?» [Электронный ресурс]. – Режим доступа: <https://coinspot.io/beginners/что-такое-bitcoin/>

УДК 519.6

Студ. А.В. Жерноклёва
 Науч. рук. ассист. О.А. Архипенко
 (кафедра высшей математики, БГТУ)

НЕКОТОРЫЕ ПРЕДСТАВЛЕНИЯ НАТУРАЛЬНЫХ ЧИСЕЛ

В современной теории чисел значительное место занимает проблема представления некоторой последовательности натуральных чисел в виде суммы ограниченного количества слагаемых заданного вида.

Цель нашей работы: представить некоторые натуральные числа в виде последовательностей любой одной из 9 цифр, а также операций сложения, вычитания, умножения, деления, возведения в степень, расстановки скобок. Кроме того, описать представление натуральных чисел с помощью цифр от 1 до 9 и вышеперечисленных операций двумя различными способами: в порядке возрастания от 1 до 9 и в порядке убывания от 9 до 1. Установить разложение числа 10958 в возрастающем случае.

Представление натуральных чисел с помощью одной цифры.
 Пусть $a \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Очевидно, что

$$0 = a - a, 1 = \frac{a}{a}, 2 = \frac{a+a}{a}, 3 = \frac{a+a+a}{a} \text{ и т. д.}$$

Мы наблюдаем, что по мере увеличения числа для записи требуется больше цифр. Но это не всегда так, например, число 10 можно записать как $10 = 11 - 1 = 2 \times 2 \times 2 + 2$. Здесь нужно только 3 цифры для единицы и четыре цифры для двойки.

Рассмотрим ряд

$$f^n(10) = 10^n + 10^{n-1} + \dots + 10^2 + 10 + 1.$$

Тогда любое положительное натуральное число, в записи которого используется только одна цифра, можно записать в виде

$$af^n(10) = \underbrace{aaa\dots a}_{(n+1)\text{-раз}}, \text{ где } a \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}.$$

В частности, для $n = 3$, имеем

$$af^3(10) = a10^3 + a10^2 + a10^1 + a = aaaa.$$

Заметим, что

$$5 = \frac{af^1(10)-a}{a+a} = \frac{aa-a}{a+a}, 55 = \frac{af^2(10)-a}{a+a} = \frac{aaa-a}{a+a},$$

и

$$6 = \frac{af^1(10) + a}{a + a} = \frac{aa + a}{a + a}, 56 = \frac{af^2(10) + a}{a + a} = \frac{aaa + a}{a + a}.$$

Тогда можно записать следующие представления чисел:

$$5 = \frac{11-1}{1+1} = \frac{22-2}{2+2} = \frac{33-3}{3+3} = \frac{44-4}{4+4} = \frac{55-5}{5+5} = \frac{66-6}{6+6} = \frac{77-7}{7+7} = \frac{88-8}{8+8} = \frac{99-9}{9+9},$$

$$55 = \frac{111-1}{1+1} = \frac{222-2}{2+2} = \frac{333-3}{3+3} = \dots = \frac{666-6}{6+6} = \frac{777-7}{7+7} = \frac{888-8}{8+8} = \frac{999-9}{9+9},$$

$$6 = \frac{11+1}{1+1} = \frac{22+2}{2+2} = \frac{33+3}{3+3} = \frac{44+4}{4+4} = \frac{55+5}{5+5} = \frac{66+6}{6+6} = \frac{77+7}{7+7} = \frac{88+8}{8+8} = \frac{99+9}{9+9},$$

$$56 = \frac{111+1}{1+1} = \frac{222+2}{2+2} = \frac{333+3}{3+3} = \dots = \frac{666+6}{6+6} = \frac{777+7}{7+7} = \frac{888+8}{8+8} = \frac{999+9}{9+9}.$$

По данному алгоритму выписаны разложения вплоть до 925 в работе [2].

Представление натуральных чисел цифрами от 1 до 9 по возрастанию (убыванию). Для двух разных положительных натуральных числа a и b можем выполнить следующие действия: $a + b$, $a \times b$, a^b и ab , где ab – двузначное число. Например, если $a = 2$ и $b = 3$, тогда есть только 4 варианта записи $2 + 3$, 2×3 , 2^3 и 23 в порядке возрастания, и $3 + 2$, 3×2 , 3^2 и 32 в порядке убывания. Эти правила можно распространить на случай, когда имеется 3 цифры $a < b < c$ ($a > b > c$) и более.

Такие разложения от 0 до 11111 опубликованы в [1], где было также доказано, что представления являются наикратчайшими. Однако для числа 10958 разложение в порядке возрастания отсутствовало.

Но мы нашли решение этой проблемы

$$10958 = 1 \times 23 + ((4 \times 5 \times 6)7 + 8) \times 9.$$

Таким образом, нами было получено разложение натурального числа 10958 с помощью цифр от 1 до 9 по возрастанию.

ЛИТЕРАТУРА

1. Taneja I. J. Crazy Sequential Representation: Numbers from 0 to 11111 in terms of Increasing and Decreasing Orders of 1 to 9 //arXiv preprint arXiv:1302.1479. – 2013.

2. Taneja I. J. Crazy Representations of Natural Numbers, Selfie Numbers, Fibonacci Sequence, and Selfie Fractions. – RGMIA Research Report Collection, 19, 2016. – P. 1-37.

УДК 634.0.30

Студ. М.Д. Фирьян

Науч. рук. доц. Игнатенко В.В. (кафедра высшей математики БГТУ);
доц. Леонов Е.А. (кафедра ЛМДиТЛП БГТУ)

ИСПОЛЬЗОВАНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ДЛЯ АНАЛИЗА РАБОТЫ ОДНОМАШИННЫХ ЛЕСОПРОМЫШЛЕННЫХ СИСТЕМ БЕЗ ЗАПАСА

Ряд лесопромышленных систем функционируют без запаса. К ним могут относиться: сортировочные лесотранспортеры, окорочные станки, лесопильные рамы и др.

Пусть лесопромышленная система состоит только из одного станка и к нему поступает на обработку пуассоновский поток предметов труда с интенсивностью λ_1 , зависящий, в общем случае, от времени $\lambda_1 = \lambda_1(t)$.

Обработка предмета труда осуществляется с изменяющейся продолжительностью цикла $t_{ц}$, распределенного по показательному закону с параметром $\mu_1 = \mu_1(t)$.

Запишем математическую модель работы системы.

Функционирование системы будет определено, если установлены ее возможные состояния. С учетом изложенного, она может находиться в одном из следующих состояний:

S_0 – оборудование исправно и простаивает из-за отсутствия сырья по организационным причинам;

S_1 – оборудование осуществляет обработку предмета труда.

Функционирование рассматриваемой системы можно представить следующей схемой (графом) состояний:

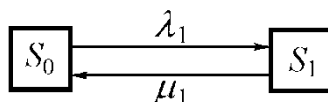


Рисунок 1 – Размеченный граф состояний рассматриваемой системы

При поступлении единицы предмета труда с интенсивностью λ_1 оборудование перейдет в состояние обработки S_1 , а после окончания операции вернется в состояние S_0 с интенсивностью обработки μ_1 .

Обозначим вероятности состояний S_0 как $P_0(t)$, а S_1 как $P_1(t)$. Для любого времени функционирования системы t

$$P_0(t) + P_1(t) = 1. \quad (1)$$

Математическая модель функционирования системы записывается системой дифференциальных уравнений Колмогорова [1]:

$$\begin{cases} \frac{dP_0}{dt} = -\lambda_1 P_0 + \mu_1 P_1, \\ \frac{dP_1}{dt} = -\mu_1 P_1 + \lambda_1 P_0. \end{cases} \quad (2)$$

В первое уравнение системы (2) подставим вместо P_1 его выражение $P_1 = 1 - P_0$ из (1), тогда

$$\begin{aligned} \frac{dP_0}{dt} &= -\lambda_1 P_0 + \mu_1(1 - P_0), \\ \frac{dP_0}{dt} &= -(\lambda_1 + \mu_1)P_0 + \mu_1. \end{aligned} \quad (3)$$

При $\lambda_1 = \text{const}$, что обосновано ранее, и при начальных условиях $P_0(0) = 1$; $P_1(0) = 0$ получим

$$P_0 = \frac{\mu_1}{\lambda_1 + \mu_1} + \frac{\lambda_1}{\lambda_1 + \mu_1} e^{-(\lambda_1 + \mu_1)t}. \quad (4)$$

Зависимости вероятностей простоя P_0 и работы системы P_1 приведены на рис. 2.

В начале работы машина свободна (не работает) и $P_0 = 1$. По мере вступления в работу вероятность P_0 уменьшается и в пределе (при $t \rightarrow \infty$) достигает значения $\mu_1 / (\lambda_1 + \mu_1)$.

Вероятность работы машины соответственно растет и достигает значения $\lambda_1 / (\lambda_1 + \mu_1)$.

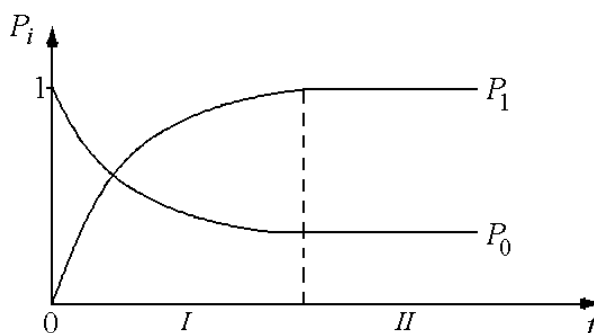


Рисунок 2 – Зависимости вероятностей функционирования системы при постоянной интенсивности поступления предмета труда на обработку

Зона I представляет собой период пуска системы с отработкой режимов эксплуатации.

В установившемся режиме эксплуатации ($t \rightarrow \infty$) при $\lambda_1 \approx \text{const}$, $P_0 \approx \text{const}$, $P_1 \approx \text{const}$ (зона II):

$$\frac{dP_0}{dt} = 0; \quad \frac{dP_1}{dt} = 0.$$

Система дифференциальных уравнений трансформируется в систему линейных алгебраических уравнений:

$$\begin{cases} 0 = -\lambda_1 P_0 + \mu_1 P_1, \\ 0 = -\mu_1 P_1 + \lambda_1 P_0, \\ P_0 + P_1 = 1. \end{cases}$$

Тогда расчетные формулы будут иметь следующий вид:

$$\begin{aligned} P_0 &= \frac{\mu_1}{\lambda_1 + \mu_1}; & P_1 &= \frac{\lambda_1}{\lambda_1 + \mu_1}; \\ \lambda_1 &= \frac{1}{t_{\text{п}}}; & \mu_1 &= \frac{1}{t_{\text{ц}}}, \end{aligned} \quad (5)$$

где $t_{\text{п}}$ – среднее значение времени между поступлениями предметов труда на обработку; $t_{\text{ц}}$ – средняя продолжительность цикла обработки предмета труда. Вероятность P_1 представляет собой коэффициент использования рабочего времени машины.

Пример. Система раскряжевки работает с циклом обработки $t_{\text{ц}} = 1$ мин. Интенсивность подачи можно изменять. Необходимо установить рациональную интенсивность подачи и цикл подачи хлыста. Интенсивность обработки составит $\mu_1 = 1 / t_{\text{ц}} = 1 / 1 = 1$ хлыст/мин.

Задаваясь различными значениями λ_1 , по формуле (5) построим зависимости для P_0 и P_1 (рис. 3).

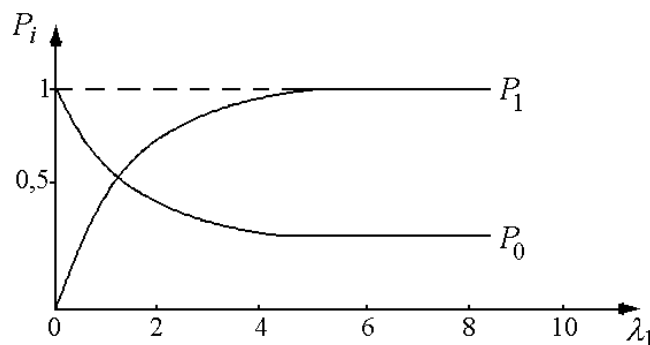


Рисунок 3 – Зависимости вероятностей работы раскряжевочной установки от интенсивности поступления предмета труда на обработку

При интенсивности подачи 1 хлыст/мин вероятность работы установки составит 0,5. Начиная с $\lambda_1 = 5-6$, дальнейшее увеличение па-

раметра существенно не повысит вероятность рабочего состояния. Так, увеличение λ_1 с 5 до 7 повысит P_1 лишь на 2%.

Рациональный цикл подачи хлыстов составит

$$t_{\pi} = 1 / \lambda_1 = 1 / 5 = 0,2 \text{ мин.}$$

Полученное значение цикла подачи хлыста позволяет выбирать подающий механизм: растаскиватель, манипулятор или др.

В анализируемых вариантах $\lambda_1 / \mu_1 < 1$, а если система работает в режиме $\lambda_1 / \mu_1 > 1$, то предыдущий механизм вынужден простаивать либо предметы труда накапливаются перед обрабатывающей установкой. Последний случай может иметь место в течение кратковременного периода работы установки.

ЛИТЕРАТУРА

1. Игнатенко В. В. Моделирование и оптимизация процессов лесозаготовок: учеб. пособие для студентов специальности «Лесоинженерное дело» / В. В. Игнатенко, И. В. Турлай, А. С. Федоренчик. – Минск: БГТУ, 2004. 180 с.

2. Игнатенко В. В., Леонов Е. А. Установление рациональных параметров многооперационных машин в лесозаготовительной промышленности // Актуальные направления научных исследований XXI века: теория и практика. 2015. Т. 3. № 5–4. С. 291–295.

УДК 634.0.30

Студ. М.М. Цмак
Науч. рук. доцент Игнатенко В.В.,
(кафедра высшей математики БГТУ)
доцент Леонов Е.А.
(кафедра ЛМДиТЛП БГТУ)

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ РАБОТЫ ОДНОМАШИННЫХ ЛЕСОПРОМЫШЛЕННЫХ СИСТЕМ С ЗАПАСОМ

Большинство лесопромышленных машин и оборудования, как правило, работают с запасом сырья, обеспечивающим ритмичность производства, уменьшение влияния негативных факторов. Это трелевочные, сучкорезные, раскряжевочные, сортировочные, окорочные, погрузочные и другие системы [1].

Рассмотрим одномашинную систему, в которую поступает поток предметов труда с интенсивностью λ_1 , интенсивностью обработки μ_1 (т. е. в среднем непрерывно занятая машина будет выдавать μ_1 обрабо-

танных предметов труда в единицу времени). Предмет труда, поступивший в момент, когда система занята, попадает в запас и ожидает обработки.

Пусть размер запаса ограничен и равняется m . Составим математическую модель работы системы, используя систему дифференциальных уравнений Колмогорова. Для этого выделим все возможные состояния системы и нарисуем размеченный граф состояний (рис. 1) [2]. Все возможные состояния системы следующие:

- S_0 – машина свободна, исправна, нет предмета труда;
- S_1 – машина обрабатывает предмет труда, запас пуст;
- S_2 – машина обрабатывает предмет труда, в запасе одна единица;
-
- S_k – машина обрабатывает предмет труда, в запасе $k-1$ предметов труда;
- S_{m+1} – машина обрабатывает предмет труда, в запасе m предметов труда.

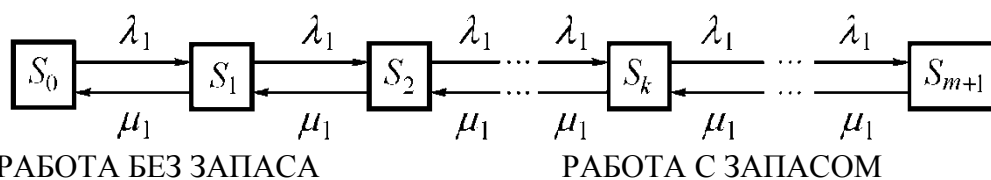


Рисунок 1 – Размеченный граф состояний одномашинной системы с ограниченным запасом

По стрелкам слева направо систему переводит поток предметов на обработку, а справа налево – поток обработанных предметов труда с интенсивностью μ_1 .

Математическая модель функционирования системы представляет систему уравнений Колмогорова

$$\begin{cases} \frac{dP_0}{dt} = -\lambda_1 P_0 + \mu_1 P_1, \\ \frac{dP_1}{dt} = -(\lambda_1 + \mu_1) P_1 + \lambda_1 P_0 + \mu_1 P_2, \\ \frac{dP_2}{dt} = -(\lambda_1 + \mu_1) P_2 + \lambda_1 P_1 + \mu_1 P_3, \\ \dots \\ \frac{dP_{m+1}}{dt} = -\mu_1 P_{m+1} + \lambda_1 P_m, \\ P_0 + P_1 + P_2 + \dots + P_{m+1} = 1. \end{cases}$$

Здесь $P_i(t)$ означает вероятность того, что в момент времени

тсистема находится в состоянии S_i . При длительной работе системы $t \rightarrow \infty$ (установившийся режим) вероятности состояний $P_i(t)$ (предельные или финальные вероятности) будут постоянными [1].

Формулы для предельных вероятностей состояний ($t \rightarrow \infty$; $P_0 \approx \text{const}$; $P_1 \approx \text{const}$; ...; $P_{m+1} \approx \text{const}$) будут иметь вид

$$P_0 = \left[1 + \rho_1 + \rho_1^2 + \dots + \rho_1^k + \dots + \rho_1^{m+1} \right]^{-1};$$

$$P_1 = \rho_1 P_0; P_2 = \rho_1^2 P_0; \dots; P_{m+1} = \rho_1^m P_0.$$

В знаменателе формулы для P_0 стоит геометрическая прогрессия с первым членом 1 и знаменателем ρ_1 . Суммируя эту прогрессию, находим

$$P_0 = \left[\frac{1 - P_1^{m+2}}{1 - P_1} \right]^{-1} = \frac{1 - P_1}{1 - P_1^{m+2}}. \quad (1)$$

Таким образом, формулы для P_1, P_2, \dots, P_{m+2} примут вид с учетом найденного выражения P_0 .

Формула (1) справедлива только при $\rho_1 \neq 1$, т. к. при $\rho_1 = 1$ формула для определения P_0 дает неопределенность $0/0$. Сумма геометрической прогрессии со знаменателем $\rho_1 = 1$ составит $m+2$.

Тогда $P_0 = P_1 = P_2 = \dots = P_{m+1} = 1 / (m+2)$.

Характеристики работы системы будут следующие. Вероятность простоя машины из-за отсутствия предмета труда $P_{\text{пр}} = P_0$. Вероятность работы машины с запасом

$$P_{\text{р.з}} = \sum_{i=2}^{m+1} P_i = 1 - P_0 - P_1. \quad (2)$$

Среднее число предметов труда, находящихся в запасе, определится как математическое ожидание дискретной случайной величины R — числа предметов труда в запасе:

$$m = M[R].$$

С вероятностью P_2 в запасе находится один предмет труда, с вероятностью P_3 — два предмета труда и т. д.

Тогда среднее число предметов труда получим, умножив число предметов труда в запасе на соответствующую вероятность и сложив результаты:

$$m = 1P_2 + 2P_3 + \dots + (k-1)P_n + \dots + m P_{m+1} =$$

$$= 1\rho_1^2 P_0 + 2\rho_1^3 P_0 + \dots + (k-1)\rho_1^k P_0 + \dots + m\rho_1^{m+1} P_0$$

После преобразований получается:

$$m = \frac{\rho_1^2 [1 - \rho_1^m (m + 1 - m \rho_1)]}{(1 - \rho_1^{m+2})(1 - \rho_1)}. \quad (3)$$

Пример. Окорочный станок ОК-40 осуществляет окорку лесоматериалов с интенсивностью $\mu_1 = 1$. Интенсивность подачи $\lambda_1 = 0,9$. Необходимо установить влияние запаса сортиментов на работу системы. Используя формулы (1), (2) и (3), построим зависимости (рис. 2, рис 3).

С увеличением количества сортиментов в запасе вероятность работы возрастает, соответственно, вероятность простоя уменьшается. Основной прирост производительности достигается созданием запаса размером 10–11 сортиментов. При этом производительность возрастает на 35% по сравнению с состоянием системы без запаса ($m = 0$). Дальнейшее увеличение запаса, например до 30 сортиментов, приведет к росту производительности всего на 4% по отношению к достигнутой при $m = 10$ –11.

До сих пор рассматривалась работа одномашинной системы с запасом при его ограниченном размере. Ряд лесопромышленных систем: очистка стволов от сучьев, отгрузка хлыстов (дереьев) на лесном складе и др. выполняются с практически неограниченными запасами предметов труда ($m \rightarrow \infty$).

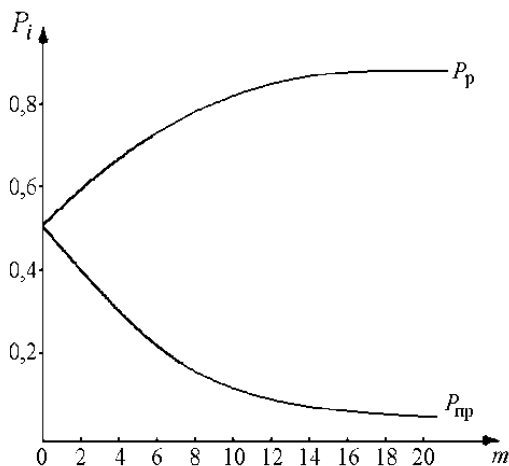


Рисунок 2 – Зависимости вероятности работы (P_p) и простоя ($P_{пр}$) окорочного станка от размера запаса/

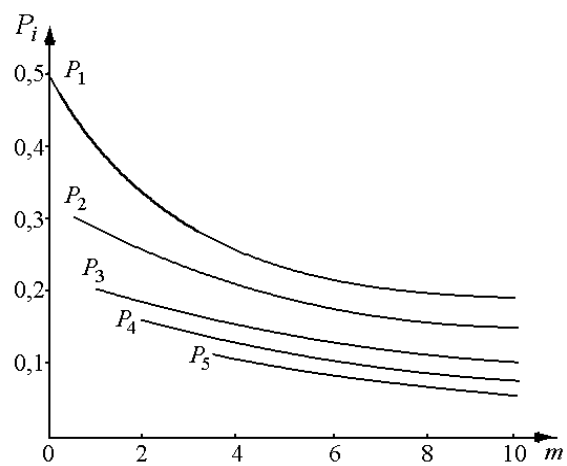


Рисунок 3 – Зависимости вероятностей состояний системы от размера запаса/

Исходя из этого, число состояний системы станет бесконечным, и схема состояний примет вид, показанный на рисунке 4.

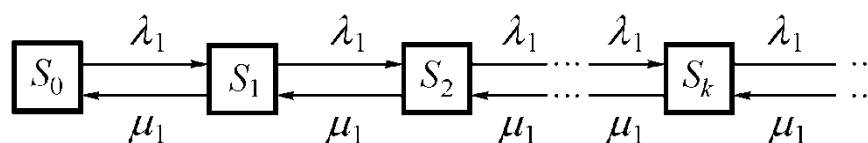


Рисунок 4 – Размеченный граф состояний одномашинной системы с неограниченным размером запаса

Функционирование подобной системы предполагает два режима, первый $\rho_1 \geq 1$. В этом случае запас предметов труда растет в принципе до бесконечности.

Следует отметить, что лесопромышленные системы в течение продолжительного периода работать в таком режиме не могут. Тогда, естественно, сохраняется условие $\rho_1 \leq 1$. Если в формуле (1) принять $m \rightarrow \infty$, то формулы для предельных вероятностей состояний системы примут вид

$$P_0 = 1 - \rho_1; P_1 = \rho_1(1 - \rho_1); P_2 = \rho_1^2(1 - \rho_1); \dots; P_k = \rho_1^k(1 - \rho_1).$$

Среднее число предметов труда в запасе будет $\bar{m} = \frac{\rho_1^2}{1 - \rho_1}$.

Среднее время ожидания на обработку составит $t_{\text{ож}} = \frac{1}{\mu_1} \frac{\rho_1}{1 - \rho_1}$.

ЛИТЕРАТУРА

1. Игнатенко, В. В. Моделирование и оптимизация процессов лесозаготовок: учеб. пособие для студентов специальности «Лесоинженерное дело» / В. В. Игнатенко, И. В. Турлай, А. С. Федоренчик. – Минск: БГТУ, 2004. 180 с.

2. Игнатенко В. В., Леонов Е. А. Установление рациональных параметров многооперационных машин в лесозаготовительной промышленности // Актуальные направления научных исследований XXI века: теория и практика. 2015. Т. 3. № 5–4. С. 291–295.

УДК 535.37+541.65+543.4

Студ. А. А. Минченя, П. Г. Климович
Науч. рук. зав. кафедрой физики,
док. физ.-мат. наук Н.Н. Крук
(кафедра физики, БГТУ)

ДЕПРОТОНИРОВАНИЕ МОЛЕКУЛ СВОБОДНЫХ ОСНОВАНИЙ КОРРОЛА В РАСТВОРАХ

Известно, что кислотно-основное равновесие между свободным основанием и депротонированной формой коррола в ядре тетрапиррольного макроцикла легко смещается при варьировании растворителя. Попытки установить корреляцию между смещением кислотно-основного равновесия в ядре макроцикла и каким-либо из эмпирических параметров, описывающих основность либо полярность растворителя, не привели к успеху [1]. Для объяснения наблюдаемых превращений была предложена двухпараметрическая модель протонирования в различных растворителях [2]. В основу модели заложена многоцентровость межмолекулярных взаимодействий в ядре тетрапиррольного макроцикла при сольватации молекул свободных оснований корролов в растворах. Поскольку молекула состоит из трех пиррольных колец ($-NH-$ центры) и одного пирроленинового кольца ($-N=$ центры), то одновременно может происходить депротонирование и протонирование ядра тетрапиррольного макроцикла, а весовые коэффициенты в двухпараметрической зависимости растворителя должны относиться как 3:1. Первый параметр — α -кислотность, описывающий способность растворителя донировать растворенной молекуле протон для образования межмолекулярной водородной связи; Второй параметр — β -основность, описывающий способность растворителя акцептировать протон, донируемый растворенной молекулой для межмолекулярной водородной связи [3].

Настоящая работа посвящена применению предложенной модели для описания кислотно-основных равновесий в ядре макроцикла корролов с различными арильными заместителями в бинарных растворителях MeCN-EtOH.

Асимметричность молекулярной структуры макроцикла корролов приводит к тому, что константы кислотности либо основности для любого из пиррольных (пирролениновых) колец могут различаться. При этом молекула свободного основания коррола характеризуется высокой скоростью таутомеризации, так что в растворе одновременно присутствуют два NH-таутомера. Поэтому межмолекулярное взаимодействие при сольватации молекулами одного и того же растворителя будет существенно зависеть от того, какова структура данного таутомера, поскольку любая

молекула растворителя обладает как основными, так и кислотными свойствами (рис.1).

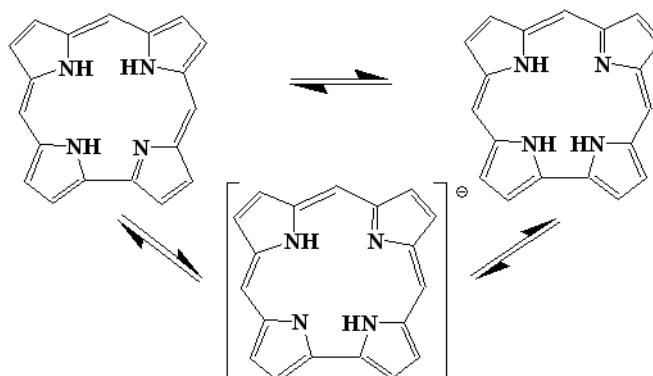


Рисунок 1 –Схема кислотно-основных равновесий корролов в растворах

На рис. 2 приведена зависимость доли депротонированных молекул коррола от объемной концентрации этанола в бинарном растворителе MeCN-EtOH. Протонирование в бинарных растворах при переходе от ацетонитрилу к этанолу анализировалось в допущении, что параметры α и β для смеси будут представлять собой взвешенную сумму с вкладом, пропорциональными объемным долям растворителей. Очевидно, что полученная функциональная зависимость для корролов с различными периферическими заместителями также является однозначной, однако в этаноле наблюдается различная доля протонированных молекул.

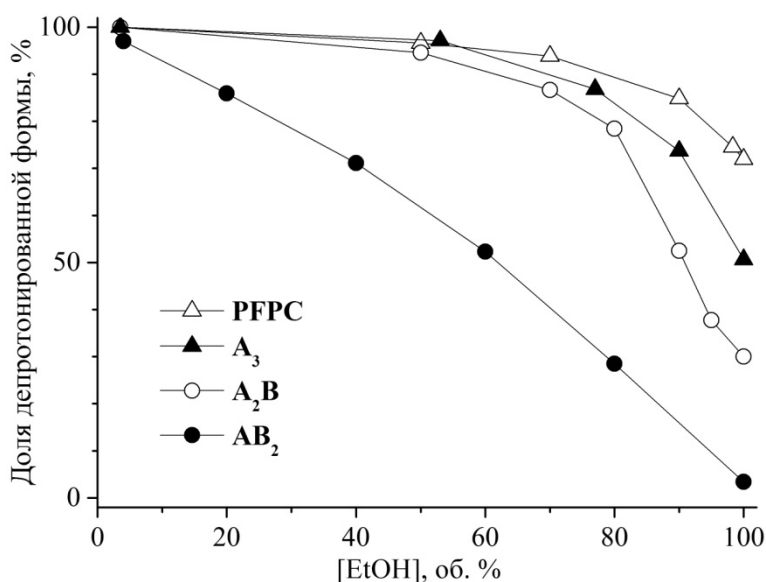


Рисунок 2 –Зависимость доли депротонированной формы корролов от объемной концентрации этанола в бинарном растворителе MeCN-EtOH.

Для определения термодинамических характеристик кислотно-основных равновесий в ядре макроцикла коррола в этаноле изучена температурная зависимость спектров поглощений исследованных соединений в температурном диапазоне 293-328 К. Обнаружено, что доля депротонированной формы увеличивается с ростом температуры для всех исследованных соединений. Анализ изменения концентраций депротонированной формы с помощью уравнения Вант-Гоффа показал, что диссоциация протона требует преодоления активационного барьера, высота которого $E_a=2,0$ ккал/моль оказалась одинаковой в пределах ошибки измерений. Предложено, что температура влияет на процесс депротонирования опосредованно, определяя величину диэлектрической проницаемости, которая падает с ростом температуры. Диэлектрическая проницаемость смещает кислотно-основное равновесие в сторону образования депротонированной формы, так как при уменьшении диэлектрической проницаемости сольватационная способность этанола уменьшается.

Таким образом, предложенная ранее двухпараметрическая модель протонирования ядра макроцикла корролов адекватно описывает процессы протонирования корролов с различным строением периферических заместителей в бинарных растворах MeCN-EtOH. Установлено, что уменьшение величины диэлектрической проницаемости растворителя с ростом температуры приводит к изменениям в сольватной оболочке, которые благоприятствуют стабилизации депротонированной формы.

ЛИТЕРАТУРА

1. Kruk, Mikalai M. Solvent-Dependent Deprotonation of meso-Pyrimidinylcorroles: Absorption and Fluorescence Studies / Mikalai M. Kruk [et al.]// Journal of Physical Chemistry, A. – 2012. - Vol. 116, № 44. - P. 10704 - 10711.
2. Mikalai M. Kruk, Wouter Maes and Roberto Paolesse, On the Shift of Equilibrium Between the Free Base Corroles and Their Deprotonated Forms in Solution// International Workshop on Photochemistry of Organic Molecules dedicated to the 85-th anniversary of academician G.P. Gurinovich, September 18 - 20, 2018, Minsk, Belarus: Program and Book of Abstracts. – Minsk: BSTU, 2018. – P.43.
3. Katritzky, A. R. Quantitative measures of solvent polarity / A. R. Katritzky [et al.] // Chem. Rev. - 2014. V. 104, №1. - P. 175 – 198.

УДК 621.37:543.42

Студ. М.С. Пилипчик
Науч. рук. ассист. кафедры А.В. Буцень¹
В.В. Кирис², Е.А. Невар²

(¹Кафедра физики, БГТУ; ²Институт физики НАН Беларуси)

ЭЛЕМЕНТНЫЙ АНАЛИЗ ВЕЩЕСТВА С ПРИМЕНЕНИЕМ МЕТОДА ЛАЗЕРНОЙ ЭМИССИОННОЙ СПЕКТРОСКОПИИ

Использование для атомизации вещества и получения плазмы импульсного лазерного излучения широко применяется в настоящее время для аналитических целей. По соотношению интенсивностей спектральных линий в лазерно-абляционной плазме, соответствующих различным элементам, можно судить о содержании их в плазме и, соответственно, в образце. Этот способ в литературе известен как лазерная атомно-эмиссионная спектроскопия (ЛАЭС).

Экспрессность, возможность одновременного определения многих элементов (включая такие легкие элементы, например H, Li, Be) как в проводящих, так и в непроводящих матрицах, высокая локальность, низкая деструктивность, возможность послойного анализа с минимальной пробоподготовкой и отсутствие ионизирующего излучения являются наиболее значимыми преимуществами метода. По данным авторов работы [1], большая часть экспресс-анализов элементного состава в мире производится с помощью атомно-эмиссионного спектрального анализа (АЭС), в число которых входит и вышеуказанный метод.

Целью настоящей работы является, в первую очередь, освоить физические основы метода, познакомиться с экспериментальным оборудованием, а также провести модельный эксперимент по качественному определению элементов, входящих в пробу, с заранее известным содержанием компонентов в образце.

Схема экспериментальной установки приведена на рис.1. Абляция и возбуждение вещества анализируемых образцов осуществлялись излучением двухимпульсного Nd:YAG лазера при следующих основных параметрах: длина волны излучения 1064 нм, энергия и длительность каждого импульса генерации (на полувысоте) 30 мДж и 12 нс, соответственно, расходимость излучения <1.5 мрад. Межимпульсная задержка (временной сдвиг между импульсами) составляла 3 мкс, что в наших условиях обеспечивало наибольшее усиление интенсивности линий. Излучение лазера проходило через оптический телескоп для расширения пучка и фокусировалось линзой ($F = 50$ мм) на поверхность мишени.

Изображение факела 1:1 проецировалось на входную щель спектрографа (ширина щели 30 мкм) при помощи кварцевой линзы (фокусное расстояние 70 мм) со светосилой, достаточной для заполнения входной апертуры спектрального аппарата. Анализируемые образцы располагались под углом $\sim 15^\circ$ относительно оси пучка лазерного излучения. Такой режим регистрации позволяет выбирать зону лазерного факела, из которой производится отбор излучения.

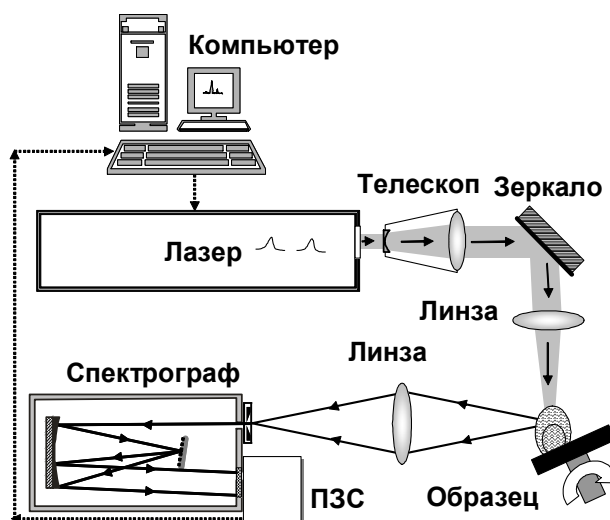


Рисунок 1 - Схема экспериментальной установки для эмиссионной диагностики плазмы

В качестве объекта исследования были выбраны монеты различных номиналов Национального банка Беларуси, которые в настоящее время являются действующим платежным средством на территории Республики Беларусь. Процедура регистрации спектров состояла из пяти сдвоенных импульсов предварительного обжига для удаления поверхностных загрязнений, окислов и микронеровностей, и пяти импульсов, спектры которых регистрировались и усреднялись. Такой режим анализа приводил к формированию кратера диаметром ~ 250 мкм и глубиной ~ 4 мкм, и при небольшой полировке образца после проведения измерений пятно поражения визуально практически незаметно.

К недостаткам ЛАЭС относят относительно невысокую чувствительность, что частично нивелируется применением сдвоенных лазерных импульсов [2]. В данном режиме работы два лазерных импульса одной длины волны воздействуют на мишень последовательно с задержкой в несколько микросекунд. При этом второй лазерный луч, воздействуя на плазму, сформированную предыдущим импульсом, позволяет существенно повысить интенсивности спектральных линий (рис. 2).

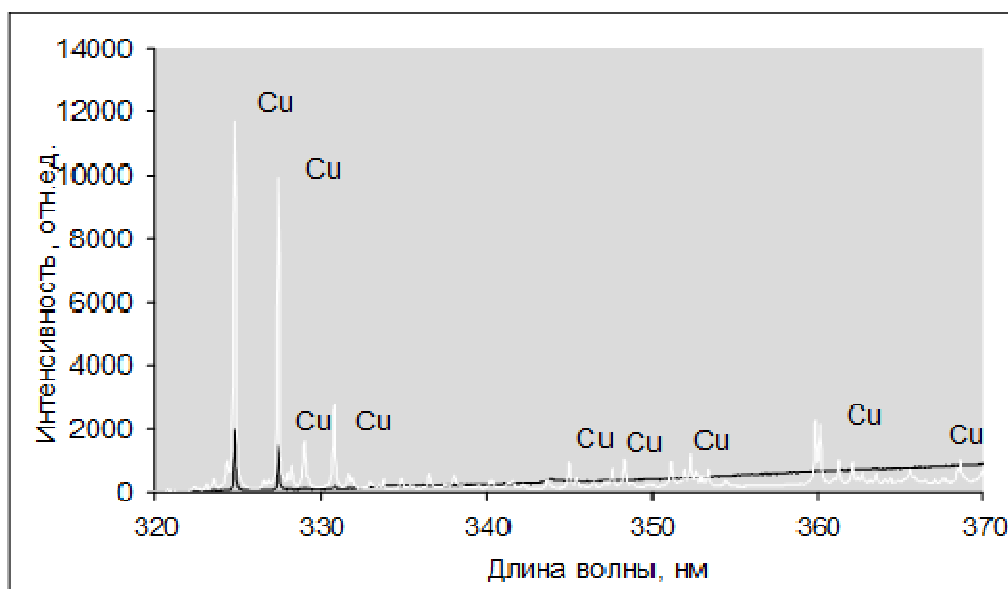


Рисунок 2 - Спектры испускания плазмы, сформированной при одно (черный цвет) и двухимпульсном (белый) лазерном воздействии на медную мишень в воздушной атмосфере

Типичные спектры испускания плазмы в диапазоне 320 – 370 нм, полученные при воздействии излучения на монеты, приведены на рис.3. По результатам идентификации спектральных линий оказалось, что для монет номиналом 1, 2 и 5 копеек все проявившиеся линии принадлежат меди. В монетах номиналом 10, 20 и 50 копеек, а также во внешней части 2-х рублевой монеты, помимо линий меди, наблюдаются также спектральные линии цинка. Монета номиналом 1 рубль визуально имеет такой же спектр, как и белая часть двухрублевой монеты, и спектр содержит линии меди и никеля. Эти заключения подтверждаются данными Нацбанка (табл.4) о составе монет [3]. Стоит отметить, что толщина медно-цинкового покрытия монет превышает по крайней мере 4 мкм (максимальная толщина слоя, из которого происходит испарение вещества мишени в плазму), поскольку интенсивных спектральных линий железа обнаружено не было.

Таблица 1 - Данные Нацбанка РБ о составе монет

	1 копейка	20 копеек	1 рубль	2 рубля
Элементный состав	Сталь, покрытая медью	Сталь, покрытая медью и латунью	Сталь, покрытая медью и никелем	Сталь, покрытая медью и латунью; Сталь, покрытая медью и никелем

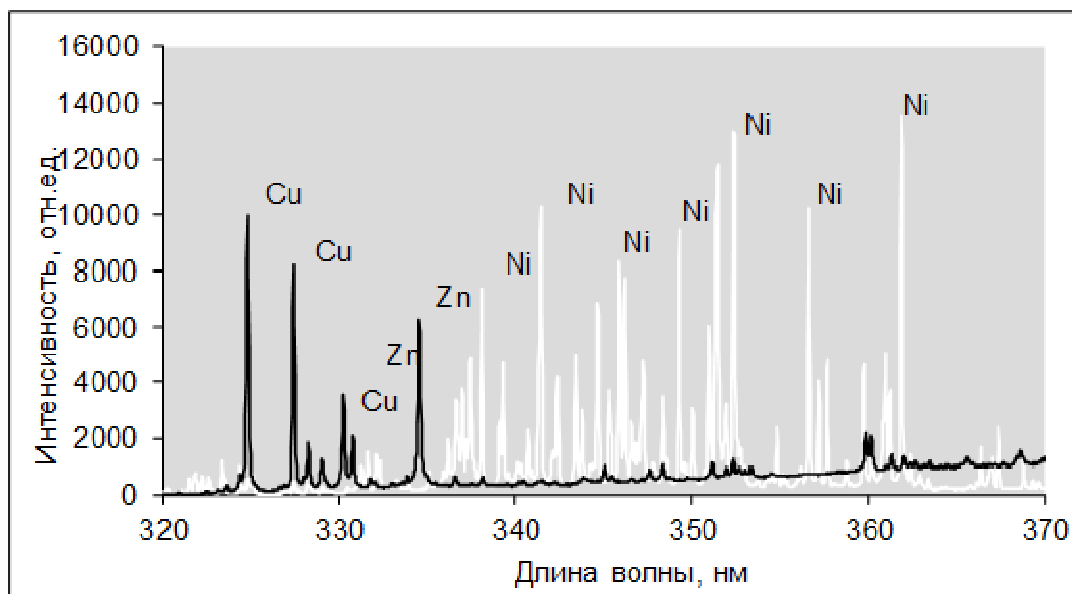


Рисунок 3 - Спектры монет номиналом 20 копеек (черный цвет) и 1 рубль (белый цвет)

Таким образом, в результате выполнения работы изучены принципы элементного спектрального анализа, наработан опыт регистрации эмиссионных сигналов и освоена методика расшифровки спектров и выбора спектрального диапазона.

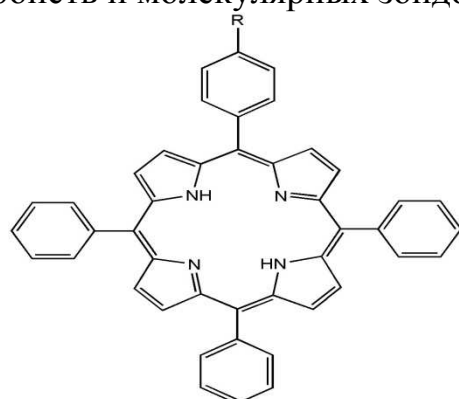
Проведенный качественный спектральный анализ позволил установить основные элементы (цинк, медь), входящие в состав покрытия денег (монет) Республики Беларусь, что подтверждают данные Нацбанка о составе монет.

ЛИТЕРАТУРА

1. Атомно-эмиссионный многоканальный спектральный анализ: научное и практическое применение / Е.С. Воропай, К.Ф. Ермалицкая, А.П. Зажогин, М.П. Патапович, А.Р. Фадаиян. – Иран: Вестник БГУ. 2009. - сер. 1. № 1,14-19 с.
2. Yingchao Li A review of laser-induced breakdown spectroscopy signal enhancement / Yingchao Li, Di Tian, Yu Ding, Guang Yang, Ke Liu, Chenghao Wang, Xu Han. - Applied Spectroscopy Reviews, 2018. - сер.53 № 1, 1-35 с.
3. Национальный банк Республики Беларусь [Электронный ресурс]. – Режим доступа: <http://www.nbrb.by/CoinsBanknotes/Coins/>– Дата доступа: 10.04.2019.

СПЕКТРАЛЬНО-ЛЮМИНЕСЦЕНТНЫЕ СВОЙСТВА АСИММЕТРИЧНО ЗАМЕЩЕННЫХ МОЛЕКУЛ ТЕТРАФЕНИЛПОРФИРИНА

Известно, что тетрапиррольные молекулы способны к специфическим межмолекулярным взаимодействиям, которые включают хелатирование ионов металлов свободными основаниями, аксиальное лигандирование хелатированного макроциклом иона металла, процессы протонирования и депротонирования макроциклического ядра, кулоновские взаимодействия протонированного или депротонированного ядра макроцикла, кулоновские взаимодействия ионизированных периферических заместителей. Эти свойства тетрапиррольных соединений лежат в основе создания на их базе молекулярных сенсорных устройств и молекулярных зондов [1,2].



ТФП R= — H

1 R= — NH₂

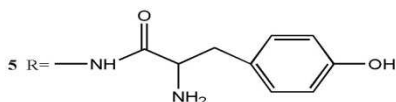
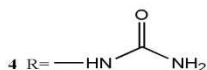
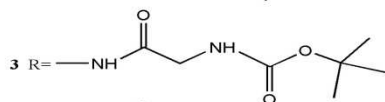
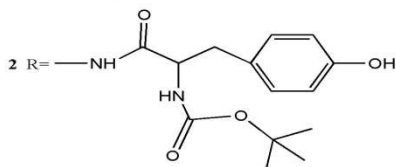


Рисунок 1 – Структура исследованных соединений

в электронных спектрах поглощения по сравнению с молекулой ТФП

Вместе с тем структурная модификация молекулы всегда сопровождается электронными эффектами замещения, обусловленными перераспределением электронной плотности между макроциклом и вновь присоединенными молекулярными фрагментами. В настоящей работе методами абсорбционной и люминесцентной спектроскопии исследовано семейство асимметрично замещенных производных молекулы 5,10,15,20-тетрафенилпорфирина **1–5** (рис. 1), которые получены путем присоединения заместителей в *пара*-положение одного из фенильных фрагментов [3].

Анализ представленных экспериментальных данных показывает, что положение полос в спектрах поглощения и флуоресценции для изученных соединений изменяется слабо. Наибольшие спектральные сдвиги

обнаруживает соединение **1**, для которого максимумы поглощения чисто электронных переходов $Q_x(0,0)$ и $Q_y(0,0)$ сдвигаются на 2 и 6 нм соответственно. Для остальных соединений эти сдвиги не превышают 0,5 и 1,5 нм. Показано, что конформационная гетерогенность, обусловленная большим количеством степеней свободы периферических заместителей, в спектрах поглощения и флуоресценции не проявляется. Так, например, полуширины полос в спектрах поглощения и флуоресценции соединения **5** в пределах ошибки измерения остаются такими же, как у молекулы **ТФП**, что указывает на отсутствие дополнительного источника неоднородного уширения (рис.2). Для других производных полуширина полос также оказалась равной полуширине полос молекулы **ТФП**.

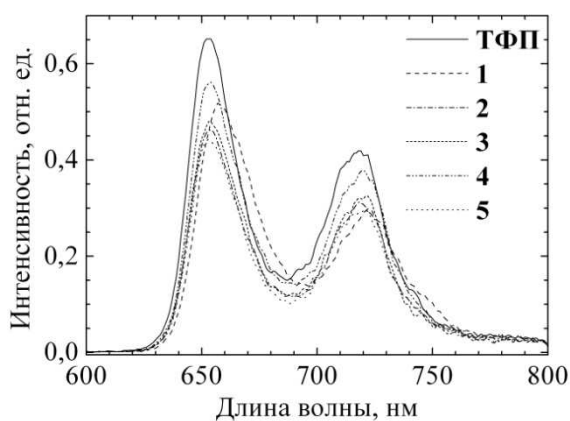


Рисунок 2 – Спектры флуоресценции соединений 1–5 и молекулы **ТФП**. Длина волны возбуждения 490 нм

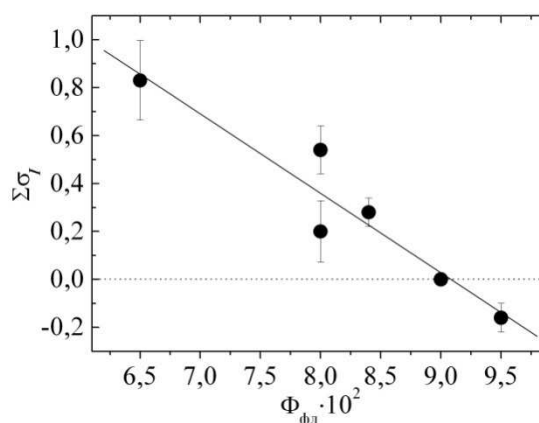


Рисунок 3 – Зависимость рассчитанной константы Гаммета σ_p *para*-заместителя от величины $\Phi_{\text{фл}}$

На рис. 3 приведен график зависимости рассчитанной суммарной константы Гаммета заместителя в *para*-положении фенильного кольца от квантового выхода флуоресценции $\Phi_{\text{фл}}$ для исследованных соединений. Очевидно, что между двумя величинами существует однозначная корреляция. Известно, что присоединение электроноакцепторных заместителей к тетрапиррольному макроциклу приводит к тушению флуоресценции посредством формирования нового канала безызлучательной дезактивации – переноса электрона. В ряду исследованных соединений порфирин **3** характеризуется максимальной суммарной константой Гаммета $\sigma_p = 0,83$. Соответственно, квантовый выход флуоресценции порфирина **3** оказался почти на 30% меньше, чем у **ТФП**.

Для NH₂-замещенного порфирина **1** вклад резонансной составляющей существенно больше, отношение A(0,0)/A(1,0) увеличивается до 0,63 по сравнению со значением 0,52 для молекулы **ТФП**. В то же время линейная корреляция индуктивной константы Гаммета $\sigma_{\text{И}}$ квантового выхода флуоресценции $\Phi_{\text{фл}}$ (рис. 5), при расчете которой для NH₂-группы использовалось значение $\sigma_{\text{I}} = -0,16$, дает основание предположить, что индуктивный эффект играет определяющую роль в тушении флуоресценции порфирина по каналу переноса заряда с макроцикла на заместитель. В то же время спектральные характеристики, определяемые положением π -орбиталей макроцикла, зависят от резонансных взаимодействий.

Таким образом, анализ результатов исследования спектрально-люминесцентных характеристик семейства асимметрично замещенных производных молекулы 5,10,15,20-тетрафенилпорфирина, полученных путем присоединения заместителей в *пара*-положение одного из фенильных фрагментов, показывает, что экспериментальные данные могут быть объяснены электронными эффектами замещения. Молекулярная конформация заместителей на спектрально-люминесцентные свойства изученных соединений не оказывает заметного влияния. Предложено, что тушение флуоресценции молекул порфиринов посредством переноса электрона с тетрапиррольного макроцикла на электроноакцепторные периферические заместители контролируется индуктивными эффектами. Снижение (с учетом архитектуры периферического замещения) молекулярной симметрии молекулы при переходе от молекулы **ТФП** (точечная группа симметрии D_{2h}) к соединениям **1–5** (точечная группа симметрии C_s) не сказывается на спектрально-люминесцентных характеристиках порфиринов.

ЛИТЕРАТУРА

- 1 Xie, Y. Porphyrin colorimetric indicators in molecular and nano-architectures / Y. Xie [et al.] // J. Nanosci. Nanotechn. – 2007. – Vol. 7, no. 9. – P. 2969–2993.
- 2 Эггинс, Б. Химические и биологические сенсоры; пер. с англ. / Б. Эггинс. – М.: Техносфера, 2005. – 336 с.
- 3 Lyubimtsev, A. Synthesis and photophysical properties of low symmetrical porphyrin-amino acid conjugates and their Zn complexes / A. Lyubimtsev [et al.] // *Macroheterocycles*. – 2018. – Vol. 11, no. 1, – P. 103–110.

Студ. А.В. Байдук
Науч. рук. доц. Г. И. Касперов
(кафедра инженерной графики, БГТУ)

ОЦЕНКА БЕЗОПАСНОСТИ ВОДОЕМОВ ТЕХНИЧЕСКОГО НАЗНАЧЕНИЯ ОЧИСТНЫХ

Очистные сооружения представляют собой специализированное оборудование для очистки сточных вод, которое может быть локального типа, то есть устанавливаться на небольших объектах, так и промышленного. Промышленный комплекс очистных сооружений – это и есть та линия, проходя через которую загрязнённые стоки (пищевые, химические, сельскохозяйственные) очищаются от вредных примесей, способных неблагоприятно повлиять на водоёмы, куда их сбрасывают. Несомненно, **аварии на очистных сооружениях** являются наиболее опасными и могут повлечь за собой самые негативные последствия, ведь продуктом производственного процесса нередко становятся агрессивные стоки, содержащие примеси тяжёлых металлов и других токсичных веществ [1].

В условиях Беларуси, как показали натурные обследования, очистные сооружения районных и областных центров это сочетание механических и биологических методов очистки сточных вод. Наличие большого количества площадных водных объектов, к которым относятся пруды биологической очистки, включающие в себя различные гидротехнические сооружения: водосбросы, перепуски, затворные механизмы, дамбы обвалования и др. имеют различный срок эксплуатации, износ и являются потенциально опасными. В этой связи, оценка состояния такого типа водных объектов и сооружений на них, с выявлением и классификацией локальных разрушений и деформаций – является актуальной задачей.

Для определения качественных показателей состояния гидротехнических сооружений (ГТС) водоемов технического назначения (очистных) по подходам, изложенным в [2], были предложены категории безопасности ГТС: I – нормальный уровень безопасности, II – неудовлетворительный уровень безопасности и III – опасный уровень безопасности

Результаты проведенных натурных обследований [3] в период с 2016 по 2019 годы по качественной оценке технического состояния водоемов технического назначения (очистных) приведены в таблице.

**Таблица – Качественная оценка технического состояния водоемов
технического назначения (очистных)**

Показатели	Кол-во объектов по категориям безопасности (табл.2) по областям								
	Витебская			Минская			Могилевская		
	I	II	III	I	II	III	I	II	III
Просадки, подвижки, оползни, трещины откосов и гребня плотины	1	2	1	5	6	3	7	8	-
Разрушение креплений, просадка, оползание, раскрытие швов и деформации плит креплений откосов и гребня плотины*	2	2	-	4	3	3	1	5	-
Повреждение облицовок, заиление, зарастание, перемерзание водосборных кюветов	-	2	2	5	6	3	13	2	-
Целостность, проточность, перемерзание водовыпусков из закрытых дренажных устройств*	1	-	-	7	4	3	13	2	-
Появление выходов воды в примыкании плотины к бетонным сооружениям*	2	2	-	6	3	2	9	3	-
Оползание откосов плотины и береговых примыканий	3	1	-	5	5	4	10	5	-
Примечание * - для отдельных объектов исследований категории безопасности не определялись, ввиду отсутствия показателей									

ЛИТЕРАТУРА

1. Ляпичев, Ю. П. Гидрологическая и техническая безопасность гидросооружений/ Ю. П. Ляпичев. — М.: РУДН, 2008. — 222 с.
2. Методика определения критериев безопасности гидротехнических сооружений. РД 153-34.2-21.342-00. М.: 2001 – 24 с.
3. Разработать научно-методические основы ведения мониторинга состояния сооружений на водоемах технического назначения для оценки последствий и ущербов от чрезвычайных ситуаций: отчет о НИР (окончат.) /БГТУ; рук.темы Г.И.Касперов. – Минск, 2018. – 254 с. – № ГР 20160782.

Студ. В. П. Невар
Науч. рук. доц. А. М. Волк
(кафедра высшей математики, БГТУ)

ОСАЖДЕНИЕ ТВЁРДЫХ ЧАСТИЦ В ОТСТОЙНИКАХ ТЕХНИЧЕСКОГО НАЗНАЧЕНИЯ

Отстойники – резервуары или бассейны для выделения из жидкости взвешенных примесей осаждением их под действием силы тяжести при пониженной скорости потока. Отстойники применяются для очистки воды в системах гидроузлов и ирригационных сооружений, водоснабжения, канализации.

Отстойники гидросиловых и ирригационных систем служат для предотвращения проникновения взвесей в турбины ГЭС и в оросительные каналы. На ГЭС и насосных станциях отстойники предохраняют лопасти гидротурбин и насосов от истирания их твёрдыми примесями (с размером частиц от 0,25 мм и более), которое может привести к понижению КПД турбин и насосов.

В ирригационных системах отстойники предохраняют каналы от заиления; в оросительную сеть с водой попадают только мелкие частицы, которые могут служить удобрением. Такие отстойники различают: по характеру работы – непрерывного или периодического действия; по способу удаления наносов – с гидравлическим промывом, с механической очисткой и комбинированные; по числу камер – однокамерные и многокамерные. Скорость течения воды в камерах отстойников (в зависимости от характера и количества наносов) – от 0,25 до 0,5 м/сек.

По направлению движения основного потока суспензии отстойники делятся на радиальные, горизонтальные, вертикальные и наклонные. В радиальных отстойниках суспензия подается в центр аппарата и движется к периферии. Радиальные отстойники обычно применяют на крупных водоочистных станциях для предварительного осветления очень мутных вод (мутность более 2 г/л), а также для очистки воды в системах оборотного промышленного водоснабжения; они оборудуются скребковыми механизмами для непрерывного удаления выпавшей взвеси.

В горизонтальных – вода загружается с одного конца аппарата и передвигается вдоль него. Их применяют для удаления взвеси: коагулированной – на водопроводных очистных станциях производительностью 30 – 50 тысяч м³/сут и некоагулированной – на станциях любой производительности. Удаление осадка из горизонтальных отстой-

ников обычно осуществляется с помощью перфорированных коробов или труб, укладываемых по дну отстойника.

В вертикальных – суспензия подается снизу и поднимается вверх, причем скорость восходящего потока должна быть меньше скорости оседания твердых частиц. Вертикальные отстойники служат для осаждения коагулированной взвеси на очистных станциях производительностью до 3 тысяч $m^3/сут$. В наклонных – осаждение осуществляется в пакетах пластин (или труб), наклоненных под углом 45-60°.

На рис. 1. Приведена схема радиального отстойника с механическим удалением осадка

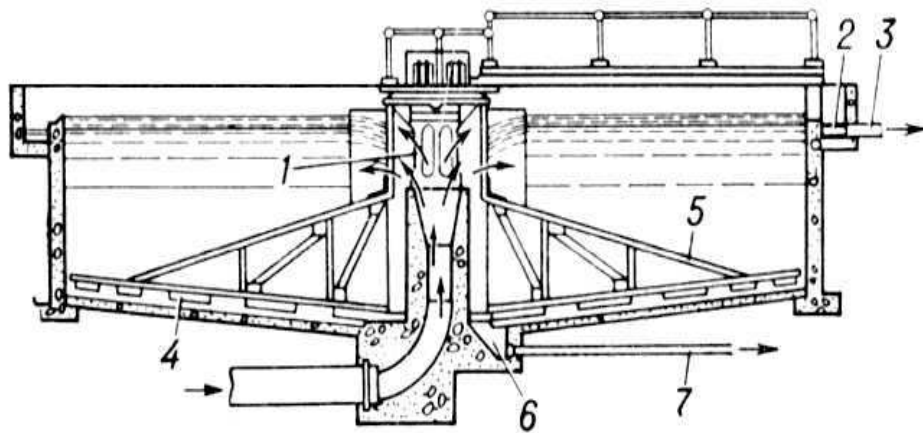


Рисунок 1 – Радиальный отстойник: 1 – центральная распределительная труба; 2 – круговой желоб; 3 – труба для отвода жидкости; 4 – скребки; 5 – движущая ферма; 6 – приемок; 7 – иловая труба.

Рассмотрим процесс осаждения твердой частицы в канале под воздействием силы тяжести, которая является движущей силой процесса. Принимаем, что частицы имеют сферическую форму диаметра d , плотность ρ_T , объем V_T и массу m . Скорость частицы \vec{W} по направлению совпадает с силой тяжести \vec{G} . При движении частицы на нее действует выталкивающая сила Архимеда \vec{F}_A и сила сопротивления среды, сила Стокса \vec{F}_C .

Запишем в декартовой системе координат уравнения движения твердой частицы в жидкости под воздействием силы тяжести:

$$m \frac{d\vec{W}}{dt} = \vec{G} - \vec{F}_C - \vec{F}_A, \quad (1)$$

Направления скорости осаждения и действующих сил изображены на рис. 2.

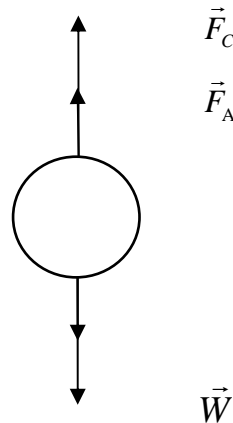


Рисунок 2 – Скорость осаждения и действующие силы

Рассмотрим силы, входящие в уравнение движения (1).

1. Сила тяжести

$$\vec{G} = mg = \frac{\pi d^3}{6} \rho_T \vec{g}; \quad (2)$$

2. Сила Архимеда

$$\vec{F}_A = \rho_{\text{ж}} V_T \vec{g} = \frac{\pi d^3}{6} \rho_{\text{ж}} \vec{g}; \quad (3)$$

3. Сила гидродинамического сопротивления среды, отнесенная к поперечному сечению, сила Стокса

$$\vec{F}_C = \varphi \frac{\pi d^2}{4} \cdot \frac{W^2}{2} \rho_{\text{ж}}; \quad (4)$$

При условии постоянства скорости осаждения уравнение (1) примет вид:

$$G - F_C - F_A = 0. \quad (5)$$

Подставляя выражения для действующих сил, получим:

$$\varphi \frac{\pi d^2}{4} \cdot \frac{W^2}{2} \rho_{\text{ж}} = \frac{\pi d^3}{6} \rho_T \vec{g} - \frac{\pi d^3}{6} \rho_{\text{ж}} \vec{g}; \quad (6)$$

Отсюда получаем скорость осаждения:

$$W = \sqrt{\frac{4d(\rho_T - \rho_{\text{ж}})g}{3\varphi\rho_{\text{ж}}}}. \quad (7)$$

Коэффициент сопротивления φ зависит от числа Рейнольдса

$$Re = \frac{Wd\rho_{ж}}{\mu}$$

Выполненные расчеты показывают, что режим является переходным ($2 < Re < 500$). В данном случае коэффициент сопротивления будет [2]

$$\varphi = \frac{18,5}{Re^{0,5}}$$

График зависимости скорости в воде W от диаметра частицы частиц песка d приведен на рис. 2.

Расчеты выполнены при следующих значениях физических величин: плотность воды $\rho_{ж} = 1000 \text{ кг/м}^3$; плотность песка $\rho_{т} = 2500 \text{ кг/м}^3$; коэффициент динамической вязкости воды $\mu = 0,0015 \text{ Н}\cdot\text{с/м}^2$.

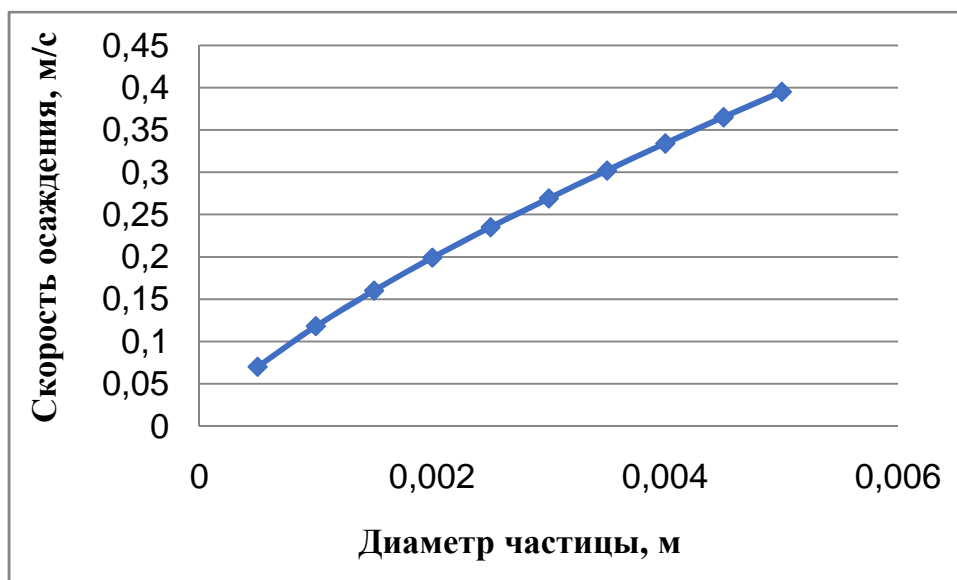


Рисунок 3 – Зависимость скорости оседания в воде W от диаметра частицы песка d

ЛИТЕРАТУРА

1. Клячко В. А., Апелцин И. Э. Очистка природных вод, М., 1971. – 579 с.
2. Соу С. Гидродинамика многофазных систем. – М.: Мир, 1971. – 536 с.

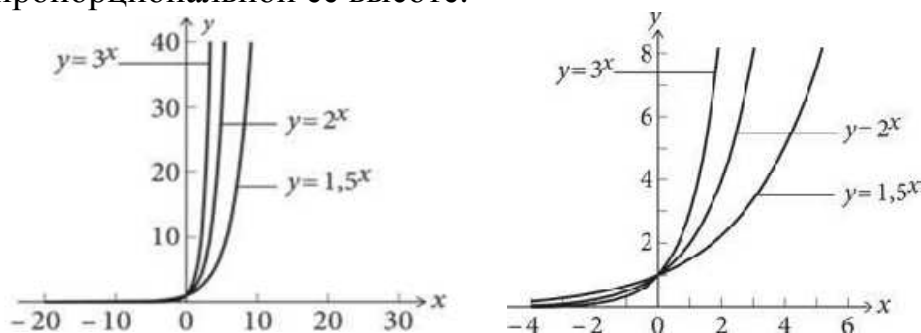
ЧИСЛО E В МАТЕМАТИЧЕСКИХ МОДЕЛЯХ*«Математика есть прообраз красоты мира»**/ И. Кеплер /*

Всем известно, что математика играет огромную роль во всех дисциплинах и их приложениях. Нет ни одной науки, в которой бы она не участвовала. Как говорится, математика нужна для изучения многих наук, но сама она не нуждается ни в одной из них. Каждое ее понятие интересно и многогранно.

Цель работы: изучить возникновение числа e ; применение его в математических моделях, в нашей жизни; способы его определения; отличия и сходства числа e с числом π .

Известно, что экспоненциальный рост имеет место в случае, если какая-то величина постоянно увеличивается пропорционально ее значению, например, путем удвоения: 1, 2, 4, 8, 16, 32, 64... или посредством умножения на три: 1, 3, 9, 27, 81, 243..., или даже посредством увеличения всего лишь на один процент.

Экспоненциальный рост может быть либо пошаговым, либо непрерывным. На представленных рисунках кривые растут экспоненциально и непрерывно. В каждой точке кривая повышается со скоростью, пропорциональной ее высоте.

**Рис.1. Экспоненциальные кривые**

Когда уравнение представлено в виде $y=a^x$, где a -положительное число, кривая демонстрирует непрерывный экспоненциальный рост. Чем дальше мы перемещаемся по экспоненциальной кривой, тем круче она становится, т.е., тем больше будет градиент. Значит, для всех экспоненциальных кривых градиент неизменно представляет собой определенный процент от высоты. Кривая, для которой значения градиента и высоты всегда равны, называется «кривая Златовласки» и выражается уравнением:

$$y=(2,7182818284\dots)^x \quad \text{или} \quad y = e^x .$$

В данной кривой высота точки на кривой всегда равна градиенту в этой точке. Однако геометрическая красота этой кривой вступает в противоречие с ее хаотической совокупностью цифр десятичного числа, начинающейся с 2,718 и продолжающейся до бесконечности без повторений. Данное число обозначили латинской буквой e и назвали экспоненциальной константой. Открытие числа принадлежит Бернулли. Оно связано с понятием сложного процента.

Сложный процент – это сумма процентных платежей, рассчитываемых за каждый очередной период, с учетом начисленных процентов.

Пример. Допустим, мы взяли кредит в размере 100 фунтов. Банк назначил сложный процент в размере 20 процентов годовых. Проследим, сколько фунтов нам придется платить ежегодно.

$$\text{Первый год: долг+проценты} = \pounds 100 + (\pounds 100 \times \frac{20}{100}) = \pounds 120;$$

$$\begin{array}{lll} \text{Второй} & \text{год:} & \text{накопленный} \\ \text{долг+проценты} = \pounds 120 + (\pounds 120 \times \frac{20}{100}) = \pounds 144; & & \end{array}$$

$$\begin{array}{lll} \text{Третий} & \text{год:} & \text{накопленный} \\ \text{долг+проценты} = \pounds 144 + (\pounds 120 \times \frac{20}{100}) = \pounds 172,8. & & \end{array}$$

Сложный процент растет гораздо быстрее, чем простой, поскольку он увеличивается по экспоненте. Прибавление x процентов к основной сумме долга равносильно умножению на $\frac{x}{100}$. Рассчитаем подробно результаты платежей первых трех лет:

$$\text{Первый год: } \pounds 100 + (1 + \frac{20}{100});$$

$$\text{Второй год: } \pounds 100(1 + \frac{20}{100})(1 + \frac{20}{100}) = \pounds 100(1 + \frac{20}{100})^2;$$

$$\text{Третий год: } \pounds 100(1 + \frac{20}{100})^2(1 + \frac{20}{100}) = \pounds 100(1 + \frac{20}{100})^3 \text{ и т. д.}$$

Эта последовательность подчиняется экспоненциальному закону.

Бернулли показал, что если частоту начисления процентов бесконечно увеличивать, то процентный доход в случае сложного процента имеет предел:

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n. \quad \left\{ \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n \right\}$$

Этот предел равен числу e . Символу для обозначения экспоненциального роста ввел Леонард Эйлер.

$e \approx 2,71828$ В настоящее время число e – это важнейший элемент математики роста и выгодно используется в банковском деле.

Число e красиво выражается в виде факториалов. Вычислим значение $\frac{1}{n!}$ для каждого числа, начиная с 0, а затем подсчитаем сумму всех членов этого ряда. В результате получится число e . В виде равенства это можно записать вот так:

$$e = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} + \dots$$

Подсчитывая сумму, получим:
1, 2, 2.5, 2.666, ..., 2.7083, ..., 2.7166, ...

Этот ряд приближается к истинному значению числа e с большой скоростью. Всего после десяти членов ряда значения совпадают с точностью до шести тысячных знаков, что очень хорошо практически для всех научных целей.

Как и в случае со сложным процентом, оно представляет собой предел $\left(1 + \frac{1}{n}\right)^n$, когда n приближается к бесконечности. Красиво представлено число e в виде второго замечательного предела: $\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e$. Доказательство очень простое: подставляем вместо n числа 1, 2, 3, ... и, высчитывая полученные значения, приблизимся к числу $e = 2,718281828\dots$

С числом e связаны гиперболические функции: $sh x = \frac{e^x - e^{-x}}{2}$,
 $ch x = \frac{e^x + e^{-x}}{2}$, $th x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, $cth x = \frac{e^x + e^{-x}}{e^x - e^{-x}}$. Их приложение огромно.

Например, большой интерес представляет цепная линия. Ее уравнение: $y = \frac{e^{ax} + e^{-ax}}{2a}$, где a – постоянная, от которой зависит масштаб кривой. Иначе, цепная линия – это среднее значение кривой экспоненциального роста и спада. Перевернутую цепную линию называют также математическим украшением. Возможно, именно благодаря ей, придумали бусы. Перевернутую цепную линию также использовал в своем творчестве известный каталонский архитектор А. Гауди при построении храма Святого семейства в Барселоне. Известные сооружения в форме цепной линии – это арка в Сант-Луисе под названием «Врата на Запад» и мегааэропорт в Кувейте.

Число e — основание натурального логарифма, иррациональное число. Оно используется также в биологии при исследовании роста клеток в организме, в формуле вычисления скорости ракеты [1], и даже – в математике выбора жены или мужа.

Одним из примеров применения гиперболических функций является граничная задача, лежащая в основе физической модели, описывающей процесс ограничения столба плазмы: $y'' = k \operatorname{sh} k y$, $y(0)=0$, $y(1)=1$. Эта модель подробно изучалась, ее оценивают в вычислительном отношении, как достаточно сложную и трудную, и проверяют на ней новые численные методы решения прикладных задач. Эта задача решалась сеточными методами, методом прогонки, методом множественной двусторонней пристрелки и т. д.

К числу e приводит и французская карточная игра в совпадения. Если в игре в совпадения колода карт каждого из игроков состоит из одной карты, то вероятность ее выпадения будет 100 процентов. Если в колоде две карты, то вероятность равна 50 процентам. Эйлер составил таблицы перестановок для игр с колодами из трёх и четырёх карт и вывел закономерность. Вероятность совпадения карт при n картах в колоде рассчитывается по формуле: $1 - \frac{1}{2!} + \frac{1}{3!} - \frac{1}{4!} + \dots + (-1)^n \frac{1}{n!}$.

Сумма этого ряда в точности равна $\left(1 - \frac{1}{e}\right)$ в случае, если n стремится к бесконечности, но приближение уже хорошее после первых членов ряда. Когда $n=52$, то есть количеству карт в колоде, сумма ряда равна числу $\left(1 - \frac{1}{e}\right)$ с точностью до 70 десятичных знаков.

Интересны соотношения между числами e и π . Они отличаются друг от друга всего на 0,5. Это иррациональные числа, десятичная часть которых содержит бесконечное число не повторяющихся цифр.

Существует ряд удивительных выражений, составленных из данных констант: $\pi^4 + \pi^5 = e^6$, что верно до семи значимых цифр; $e^\pi - \pi = 19.999099979 \dots$, что очень близко к 20; $e^{i\pi} + 1 = 0$. и самое впечатляющее уравнение: $e^{\pi\sqrt{163}} = 262537412640768743.999999999$, что всего на одну триллионную меньше целого числа.

Вывод: Вспомним слова знаменитого энтомолога Ж.Фарба: «Рассмотрим паутину. Усеянные крохотными капельками, ее липкие нити провисают, образуя цепные линии, и вся сеть похожа на множество ожерелий. При появлении солнца паутину начинает переливаться всеми цветами радуги, превращаясь в сверкающую гроздь бриллиантов, и число e предстает во всем своем великолепии».

ЛИТЕРАТУРА

1. Перельман Я. И. Занимательная алгебра. Изд. М., 1967, 200 с.

УДК 51:542.9

Студ. Д. С. Шелегова
 Науч. рук. ассист. Е. В. Терешко
 (кафедра высшей математики, БГТУ)

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ СКОРОСТИ ХИМИЧЕСКОЙ РЕАКЦИИ

Мы пользуемся математикой, чтобы описать закономерности явлений вокруг нас. И если мы изучаем иностранные языки, чтобы иметь возможность общаться с любым человеком, то владение языком математики даст нам больше. Мы сможем вести диалог с природой.

Применение и важность математики будет показана на примере решения задач в химической кинетике – разделе физической химии, изучающей закономерности протекания химических реакций во времени, зависимости этих закономерностей от внешних условий, а также механизмы химических превращений.

Цель работы: проанализировать и выбрать задачи из химической кинетики, вывести формулы для их решения, продемонстрировать возможности математики в описании закономерностей химических процессов.

Для того, чтобы реакция произошла, должны встретиться и столкнуться две молекулы. Согласно основному постулату химической кинетики, называемому также законом действующих масс, скорость реакции пропорционально произведению текущих концентраций реагирующих веществ, возведенных в некоторые степени [1]:

$$\omega = kC_1^{n_1}C_2^{n_2} \dots C_i^{n_i},$$

где k – коэффициент пропорциональности, называемый константой скорости реакции, C_i – концентрация вещества, $n_1, n_2 \dots n_i$ – порядок реакции, определяемый экспериментально.

Для элементарных реакций (те, что происходят в результате парного столкновения или распада одной частицы) порядок можно принять равным коэффициенту, стоящему перед веществом.

Реакции первого порядка. В общем виде реакции первого порядка можно записать $A \rightarrow B$. Тогда уравнение изменения исходной концентрации вещества A выглядит $-\frac{dC_A}{dt} = kC_A$ [2].

Решим дифференциальное уравнение, проинтегрировав выражение в пределах от C_0 до C_A и выразим ответ через C_A , $C_A = C_0 e^{-kt}$ [3].

Реакции второго порядка. Реакции второго порядка имеют два возможных варианта протекания: $2A \rightarrow B$ и $A + B \rightarrow D + E$.

В первом случае уравнение скорости реакции выглядит $-\frac{dC_A}{dt} = kC_A^2$. Решив уравнение, получим $C_A = \frac{C_0}{C_0kt+1}$.

Во втором случае возможны 3 варианта:

а) концентрация вещества А равна концентрации В. Тогда мы можем записать уравнение в виде $-\frac{dC_A}{dt} = kC_A^2$, решение которого уже было рассмотрено выше.

б) концентрация одного из веществ значительно превосходит концентрацию второго: $C_A \gg C_B$. В данном случае изменением концентрации преобладающего вещества мы можем пренебречь, а саму концентрацию считать постоянной и внести в k . Тогда скорость будем считать по концентрации только одного вещества. Порядок таких реакций принято называть псевдопорядком.

$$-\frac{dC_A}{dt} = kC_A,$$

$$C_A = C_0 e^{-kt}.$$

в) Концентрации А и В различны.

$$-\frac{dC_A}{dt} = kC_A C_B.$$

Сократим количество неизвестных. Пусть x – текущая концентрация образовавшихся веществ. Тогда $C_A = C_{OA} - x$, а $C_B = C_{OB} - x$.

$$\int_0^x \frac{dx}{(C_{OA} - x)(C_{OB} - x)} = k \int_0^t dt.$$

Проинтегрируем левую часть методом неопределенных коэффициентов[3]:

$$\begin{aligned} \int_0^x \frac{dx}{(C_{OA} - x)(C_{OB} - x)} &= -\frac{1}{C_{OA} - C_{OB}} \int_0^x \frac{dx}{(C_{OA} - x)} + \\ &+ \frac{1}{C_{OA} - C_{OB}} \int_0^x \frac{dx}{(C_{OB} - x)} = \frac{1}{C_{OA} - C_{OB}} \left(-\ln \frac{C_{OA}}{C_{OA} - x} + \ln \frac{C_{OB}}{C_{OB} - x} \right) = \\ &= \frac{1}{C_{OA} - C_{OB}} \ln \frac{C_{OB}(C_{OA} - x)}{C_{OA}(C_{OB} - x)}. \end{aligned}$$

Тогда решение дифференциального уравнения примет вид:

$$C_A = C_{OA} - \frac{C_{OA}C_{OB}(e^{kt(C_{OA}+C_{OB})}-1)}{C_{OB}-C_{OA}e^{kt(C_{OA}-C_{OB})}},$$

$$C_B = C_{OB} - \frac{C_{OA}C_{OB}(e^{kt(C_{OA}+C_{OB})}-1)}{C_{OB}-C_{OA}e^{kt(C_{OA}-C_{OB})}}.$$

Реакции третьего порядка. Реакции вида $3A \rightarrow E$ могут быть осуществимы в процессе перекombинации атомов и простых радика-

лов и описываются уравнением $-\frac{dC_A}{dt} = kC_A^3$.

Наиболее проблематичным случаем являются реакции, где концентрации веществ не одинаковы: $-\frac{dC_A}{dt} = kC_A^2C_B$. Примем $C_A = C_{OA} - x$, а $C_B = C_{OB} - x$. Тогда $\int_0^x \frac{dx}{(C_{OA}-x)^2(C_{OB}-x)} = k \int_0^t dt$.

Распишем левую часть в виде суммы простых дробей

$$\frac{1}{(C_{OA} - x)^2(C_{OB} - x)} = \frac{A}{(C_{OA} - x)} + \frac{B}{(C_{OA} - x)^2} + \frac{C}{(C_{OB} - x)};$$

$$A = \frac{1}{2C_{OA}C_{OB} - C_{OB}^2 - C_{OA}^2}, B = -\frac{1}{(C_{OA} - x) - (C_{OB} - x)}, C =$$

$$= -\frac{1}{2C_{OA}C_{OB} - C_{OB}^2 - C_{OA}^2}.$$

После подстановки и преобразований получим:

$$\int_0^x \frac{dx}{(C_{OA} - x)^2(C_{OB} - x)} = \frac{1}{2C_{OA}C_{OB} - C_{OA}^2 - C_{OB}^2} \ln \frac{C_{OB}(C_{OA} - x)}{C_{OA}(C_{OB} - x)} -$$

$$- \frac{1}{C_{OA} - C_{OB}} \left(\frac{1}{C_{OA} - x} - \frac{1}{C_{OA}} \right).$$

Так как выражение концентрации из полученного выражения слишком громоздко, то приведено решение для расчета значения k .

$$k = \frac{1}{t} \left(\frac{1}{2C_{OA}C_{OB} - C_{OA}^2 - C_{OB}^2} \ln \frac{C_{OB}(C_{OA} - x)}{C_{OA}(C_{OB} - x)} - \right.$$

$$\left. - \frac{1}{C_{OA} - C_{OB}} \left(\frac{1}{C_{OA} - x} - \frac{1}{C_{OA}} \right) \right).$$

В ходе работы были выведены формулы, используя которые, можно находить концентрацию вещества в любой момент времени, что является важным в любых отраслях химической, фармакологической, пищевой промышленности.

ЛИТЕРАТУРА

1. А. И. Клындюк, Физическая химия : учебное пособие для студентов учреждений высшего образования по химико-технологическим специальностям / А. И. Клындюк, Г. С. Петров, Е. А. Чижова. – Минск : БГТУ, 2013. - 299 с.

2. Семиохин И. А., Страхов Б. В., Осипов А. И. Кинетика химических реакций: Учеб. пособие. – М.: Изд-во МГУ 1995. – 351 с.

3. Высшая математика : учебное пособие для студентов высших учебных заведений по техническим специальностям : в 2 ч. Ч. 1 / В. М. Марченко [и др.]. – Минск : БГТУ, 2010. – 204 с.

**Секция
ПЕРВЫЙ ШАГ В НАУКУ**

УДК 666.913

Учащаяся А.А. Телешова
Науч. рук. преп. М.Н. Буель
(филиал БГТУ «Белорусский государственный колледж
промышленности строительных материалов»)

ИЗУЧЕНИЕ ВЛИЯНИЯ НА ИЗМЕНЕНИЕ СВОЙСТВ ПРИ ВВЕДЕНИИ В ГИПСОВУЮ МАССУ ГРАНИТНОГО ОТСЕВА

*Качество – это делать что-либо
правильно, даже когда никто не смотрит.
Генри Форд*

Гипотеза: при подборе рационального состава массы производится выпуск продукции с заданными свойствами и наименьшими затратами.

Целью данной работы является разработка составов массы для выпуска продукции с заданными свойствами; Найти новые отрасли применения гранитного отсева; Практически доказать эффективность составов на основе гранитного отсева.

Поставленная цель обусловила необходимость решения следующих задач:

- разработать новый состав на основе гипса и гранитного отсева;
- изготовить образцы для проведения основных испытаний на прочность, водопоглощение и морозостойкость;
- на основе полученных результатов сделать вывод о пригодности состава.

Одной из серьезных проблем в сфере производства нерудных строительных материалов является образование большого количества отсевов камнедробления. Так при производстве щебня из изверженных пород выход отсева дробления (фракция 0—5 мм или 0—10 мм) составляет в среднем 25 %, а из карбонатных пород 35 %.

В связи с возросшим спросом на щебень, неизбежно увеличивается выход отсева дробления, который не реализуется и складировается.

В сложившейся ситуации необходимо производить поиск рынков сбыта отсевов дробления или продуктов их переработки.

Гранитный отсев – материал, который получается путем дробления и дальнейшего просеивания гранитной породы. Такой способ считается основной технологией в изготовлении материала. Кроме этого, он может получиться и как побочный результат при изготовлении гранитного щебня. Если получается мелкая фракция, имеющая

размеры 2-5 мм, то ее именуют отсевом. Все, что представлено в большем размере, является щебнем.

По качеству отсев такой же прочный, как и щебень, поскольку из него же он и производится. Материал имеет достаточно тяжелый вес и большую плотность. Среди гранитного отсева можно встретить 2 вида: красный и серый. Производится материал согласно требованиям ГОСТа, благодаря чему сохраняются механические и физические свойства. Вес отсева может быть разным, так как это зависит от его фракций. Его физико-химические свойства аналогичны щебню. Кроме этого, гранитный отсев имеет антигололедные свойства, а также стойкость к механическим воздействиям, прочность, износостойкость и влагостойкость. Благодаря этому изделия из него обладают высоким качеством. Он не ведет к таянию снега из-за отсутствия химических веществ. Именно поэтому материал не может негативно сказаться на обуви человека, а также на его здоровье.

На сегодняшний день гранитный отсев применяется:

- отсыпки водоемов технического назначения,
- при строительстве железнодорожных путей,
- строительстве дорог,
- при изготовлении железобетонных изделий,
- он входит в состав тротуарной плитки,
- применяется декоративного материала в ландшафтном дизайне,
- в изготовления кирпича.

Основные преимущества гранитного отсева:

1. Отсев гранитный невосприимчив к неблагоприятному воздействию внешней среды, а также маслянистых материалов.
2. Отсев обладает высоким показателем прочности строительной смеси.
3. На отсев практически не влияют воздействия природных осадков и резких перепадов рабочей температуры.
4. Отсев гранитный имеет широкое применение в различных сферах строительства, а также в уходе за дорожным покрытием в зимнее время.

Теперь мы понимаем, почему это материал не теряет свою популярность с годами. Всё это позволяет использовать отсев во время производства строительных смесей.

С помощью гранитного отсева получается улучшить коэффициент прочности готового гипсового изделия и предать ему новые свойства.

Гипс. По химическому составу — это водный раствор сульфата кальция. Его химическая формула — $\text{CaSO}_4 \cdot 2\text{H}_2\text{O}$. При нагревании до 140°C из кристаллической решетки гипса выделяется вода, в результате чего образуется так называемый полуводный гипс.

Поскольку гипс состоит из довольно хрупкого вещества — кальция, то для улучшения качества получаемого материала, к нему добавляют различные вещества и примеси.

Таблица 1 - Результаты испытаний

№	Состав (Г/ГО)	Прочн. на изгиб		Описание внешнего вида	Водопоглощение	Выводы, наблюдения
		1 исп.	2 исп.			
1	100/0	42,1	39,2		2	
2	90/10	43,3	37,0	появляется серый оттенок, пористая структура	0,8	образец прочнее
3	85/15	45,09	39,3	форма не соблюдена, наблюдаются впадины и поры, серый цвет, пористая структура	0,65	образец прочнее
4	80/20	46,7	42,3	имеются впадины, темно-серый цвет, пористая структура	0,4	образец прочнее
5	75/25	49,4	44,4	неравномерный темно-серый цвет, пористая структура	0,2	Прочность увеличивается, образец не теряет полезных свойств

На базе колледжа были проведены исследования. А именно, изучено влияние гранитного отсева на качество строительного гипса. Согласно проведенным исследованиям можно сделать вывод о том, что благодаря добавке «гранитный отсев», водопоглощение стремится к нулю, прочность повышается, следовательно, изделия из такого состава могут использоваться на улице, в качестве декорирования внешних фасадов зданий и для ландшафтного дизайна. В практической деятельности этот состав был опробован в стенах колледжа. Результаты приведены ниже.

Проведенная исследовательская работа показала, что состав на основе гранитного отсева и гипса является перспективным и изделия из него могут использоваться для оформления домов и улиц.

В ходе работы были сделаны выводы: состав гранитный отсев + гипс позволяет делать изделия, которые могут применяться на улице в качестве оформления внешних фасадов и улиц.

УДК 821.161.3-14

Навуч. П.В. Крук
Навук. кіраўнік Н.М. Мяркулава
(філіял БДТУ «Бабруйскі дзяржаўны лесатэхнічны каледж»)

«ЖЫЦЦЁ» ЎНУТРЫ ВЕРША

Кожны паэт, пісьменнік праз свой твор імкнецца выразіць тые думкі і пачуцці, якія хвалююць яго больш за ўсё. Але любая думка павінна быць змешчана ў нейкую форму і мець змест, інакш яна не будзе мець той мастацкай прывабнасці, якая адрознівае мастацкі твор ад набору слоў.

Наколькі вядома, Караткевіч добра маляваў. Не выпадкова па контурах яго верша можна ўбачыць вобраз ужо немаладога чалавека, якім з'ўляўся паэт пад час напісання верша. Магчыма, аўтар хацеў занатаваць самага сябе і свой стан у гэты перыяд часу, а магчыма, гэты партрэт і ёсць знешняя форма верша.

Я паспрабавала даказаць, што верш – гэта не акамянелая форма, а нейкі «жывыарганізм».

З'яўляецца актуальным даследаванне вершаў як «жывых арганізмаў», якія могуць самастойна жыць пасля іх напісання.

Аб'ект даследавання – вершы У.С.Караткевіча «Стаю ўначы ў завіруху» і «О каханне маё бясконцае».

Прадмет даследавання- гукавы склад вершаў У.С.Караткевіча «Стаю ўначы ў завіруху» і «О каханне маё бясконцае».

Навізна работызключаецца ў разглядзе вершаў праз частотныя гукавыя ваганні.

Пры разглядзе дадзенай тэмы выкарыстаны наступныя метады работы: параўнальна-супастаўляльны, аналіз і сінтэз. У ходзе майго даследавання на прыкладзе супастаўляльнага аналізу вершаў У.С.Караткевіча «Стаю ўначы ў завіруху» і «О каханне маё бясконцае» былі зроблены спробы пераасэнсавання верша як «жывой формы». У прыродзе, як і ў вершы, абсалютна ўсё ўзаемазвязана ў адзіны арганізм, у якім цесна пераплятаюцца між сабой форма і змест.

Па графіку частотных гукавых і мысленчых ваганняў выявілася, што рытм і гук цесна ўзаемазвязаны паміж сабой, вызначаюць мысленчы працэс, які адбываецца ў вершах, які спрабуе выйсці з пачатковай аўтарскай формы.

Па «фізічных» гуках вызначыла, што форма існуе не толькі знешне, але і унутрана. Яна дапамагае ўтрымліваць змест.

Можна меркаваць, што пачатковы «фізічны» гук супадае з перажываннямі і пачуццямі У.С.Караткевіча ў момант напісання гэтых вершаў. Гэта цалкам тлумачыць партрэтны контур, падобны да ўзроставых асаблівасцей паэта. Так аўтар першасна ўстанаўлівае знешнюю форму (цела), у якую затым укладвае нейкае напаўненне – змест. Колер – гэта гукавыя адбіткі пачуццяў, эмоцый, а значыць, з’яўляецца тым напаўненнем формы, без якога немагчыма разуменне верша.

УДК 316.774

Навуч. А.А. Емяльянава
Навук. кіраўнік Н.М. Мяркулава
(філіял БДТУ «Бабруйскідзяржаўнылесатэхнічны каледж»)

СРОДКІ ПРЫЦЯГНЕННЯ ЎВАГІ ЧЫТАЧОЎ Ў ЗАГАЛОЎКАХ ГАЗЕТНЫХ ПУБЛІКАЦЫЙ

Газета – адно з важнейшых сучасных сродкаў масавай інфармацыі. Яна захоўвае сваё асобае месца ў сістэме іншых СМІ як старэйшае і незаменімае перыядычнае выданне. Сёння перыядычныя выданні вымушаны шукаць спосабы прыцягнення ўвагі чытачоў[2]. Адным з такіх спосабаў з’яўляецца загаловак.

Загалавак засяроджвае ўвагу на найбольш важных і цікавых момантах, заахвочваючы чытача азнаёміцца з тэкстам. Загалоўкі дапамагаюць яму хутка азнаёміцца са зместам нумара, зразумець, пра што паведамляюць яго публікацыі, што важна ў інфармацыі, якую яму прапануюць, што выклікае ў яго асаблівую цікавасць [3].

Актуальнасць даследавання абумоўлена тым, што ў наш імклівы час у людзей усё менш і менш часу чытаць газеты што называецца “ад скарынкі да скарынкі”, і роля загалоўкаў шматразова ўзрастае – менавіта ад іх залежыць, будзе прачытаны матэрыял ці не.

Мэтай дадзенага даследавання з’яўляецца выяўленне розных сродкаў прыцягнення ўвагі чытачоў у газетных загалоўках; выявіць, артыкулы з якімі загалоўкамі больш прыцягваюць увагу чытачоў. Для дасягнення дадзенай мэты былі выкарыстаны наступныя метады:

- 1.Метад суцэльнага выбару (сабраць газетныя загалоўкі);
- 2.Метад класіфікацыі (класіфікаваць газетныя загалоўкі па выкарыстаных у іх сродках прыцягнення ўвагі);
3. Імпірычны метады даследавання (праанкеціраваць навучэнцаў з мэтай выяўлення цікавасці да артыкула).

На аснове разгледжанага матэрыялу зроблены наступныя вывады:

1. У загалюўках газеты “Звязда” найчасцей сустракаюцца такія сродкі прыцягнення ўвагі як лексічныя .

2. Нямнога радзей сустракаліся загалюўкі, утвораныя шляхам трансфармацыі назваў мастацкіх твораў, назваў мастацкіх фільмаў і радкоў з песень і загалюўкі, утвораныя пры дапамозе стntaxічных сродкаў.

3. Найбольш рэдка сустракаліся назвы загалюўкаў, утвораныя пры дапамозе фанетычных, словаўтваральных сродкаў.

Што датычыць рэгіянальнай газеты “Бабруйскае жыццё”, то ў даследаваным матэрыяле найбольш пашыранае выкарыстанне апавядальных загалюўкаў, якія нясуць кароткую інфармацыю пра тэкст; загалюўкі, утвораныя шляхам пашырэння лексічнай спалучальнасці; загалюўкі, утвораныя пры дапамозе сінтаксічных сродкаў (прыём змоўчання, пыталыныя канструкцыі, клічныя канструкцыі); іншыя.

Праведзенае сярод навучэнцаў апытанне даказала, што найбольш цікавымі з’яўляюцца загалюўкі, якія ў сваёй аснове маюць назвы мастацкіх твораў і цытаты з іх, фразеалагічныя звароты, знаёмыя чытачу выразы і фразы.

ЛІТАРАТУРА

1. Гуревич, С.М. Газета: вчера, сегодня, завтра / С.М. Гуревич. – М., 2004. – 288 с.

2. Іўчанкаў, В.І. Дыскурс беларускіх СМІ. Арганізацыя публіцыстычнага тэксту / В.І. Іўчанкаў. – Мінск: БДУ, 2003. – 257 с.

3. Лазарева, Э.А. Заголовок в газете / Э.А. Лазарева. – Свердловск, 1989. – 96 с.

4. Скоробогатько, В. Комикс вместо заголовка / В. Скоробогатько // Журналист. – 2004. – № 1. – С. 73–75 .

5. Тертычный, А.А. Заголовок – слово главное / А.А. Тертычный // Журналист. – 2004. – № 1. – С. 80.

УДК 615.322

Учащийся К.М. Силич
Научн. рук. преп. Л.В. Саливон
(филиал БГТУ «Бобруйский государственный лесотехнический колледж)

ИВАН-ЧАЙ – БЕЛОРУССКИЙ ЧАЙ

Целью исследовательской работы является узнать можно ли чай из кипрея считать заменителем традиционного чая.

В качестве объекта исследования выступает иван-чай. Основные методы исследования - эмпирический (изучение разнообразных источников информации), теоретический (анализ, синтез). Основная часть работы включает в себя следующие главы:

1. История появления чая в Европе. Кратко изложен материал об истории распространения культуры чая на Востоке и в Западной Европе Развитие чайной промышленности.

2. История появления появления чая на Руси.

3. Чай из кипрея на Руси. История копорского чая. Экспорт иван-чая в страны Западной Европы. Петр Бадмаев - открыл клинику, где готовили настойки и отвары на основе иван-чая.

4. Ботаническое описание Иван-чая узколистного (*Chamaenerion angustifolium*).

5. Основные этапы приготвление чая из кипрея в домашних условиях: сбор сырья, завяливание листьев, ферментация, сушка, сухая ферментация.

6. Полезные свойства иван-чая.

7. Сравнительный анализ традиционного чач (китайского) и чая из кипрея.

В заключении сделан вывод о том, что чай из кипрея может стать прекрасной альтернативой в отсутствие традиционного (китайского) чая.

УДК 336.7

Учащиеся И.С. Воронец, С.А. Крутолевич
Науч. рук. преп. Н.В. Казакова
(филиал БГТУ «Витебский государственный технологический колледж»)

КРАУДФАНДИНГ – БУДУЩЕЕ ВАШИХ ИДЕЙ

Что такое краудфандинг. По определению, краудфандинг - это совместное сотрудничество людей, которые объединяют свои ресурсы вместе, чтобы поддержать других людей. Актуальность краудфандинга заключается в том, что проекты реализуются через интернет. Для реализации краудфандинга используют крауд-платформы, например, Талака, Улей, Kikstarter.

Кому полезны крауд-технологии.

Абсолютно всем! Думаю, именно это вы хотели услышать от нас, но, к сожалению, это не так, т.к. есть люди, которым краудфандинг противопоказан. Крауд-технологии могут помочь в поддержке инновационных проектов, финансированию благотворительных акций, поддержке молодых художников, инвестициям в начальный капитал малого бизнеса, поддержке со стороны фанатов и поклонников, созданию свободного программного обеспечения, поддержке политических компаний. Разберем это все на примерах.

- Поддержка инновационных проектов: проект Бояна Слэта по очистке океана. Реализован с помощью Kikstarter.

- Поддержка политических компаний: Барак Обама только на этапе предварительных выборов собрал около \$272 млн. при помощи краудфандинга.

- Создание свободного программного обеспечения: возьмем успешно реализованный проект на базе платформы улей. Выпускник нашего колледжа (Ерлан) вместе со своей командой придумал и успешно реализовал польско-белорусский проект «ForestSpace». Они создали интерактивную карту Беловежской пуши на основе мифов и легенд наших предков. Этот проект содержит в себе видео и аудио ряд про мифических существ. Проект сохраняет главную историко-культурную ценность Беловежской пуши.

- Поддержка со стороны фанатов и поклонников: БИ-2 при помощи краудфандинга собрали 20 тыс. долларов на один из своих альбомов.

- Возможное финансирование благотворительных акций: Профи-Помощь в формате благотворительных акций и мероприятий, реализуемых учащимися филиала БГТУ "ВГТК", воспитанникам Детского дома города Витебска.

Чтобы более точно узнать о распространённости краудфандинга, мы провели опрос среди преподавателей колледжа. Результат подтвердил, что крауд-технологии малоизвестны. Чтобы понять почему краудфандинг малоизвестен, мы выявили плюсы и минусы крауд-технологий.

Плюсы и минусы

Для начала плюсы:

- независимость;
- отсутствие посредников;
- равные правила игры;
- открытость сборов;
- подотчетность.

Минусы:

- требуется знание маркетинга, платформ для раскрутки;
- требуются аналитические способности;
- возможность мошенничества.

Рассмотрим пошаговые инструкции реализации крауд-проекта. Он состоит из трех этапов, последовательность которых нельзя менять.

Пошаговые инструкции реализации краудфандинга.

1. Готовится автор: кратко о самом важном, что должен иметь в своем арсенале автор-движок проекта.

2. Подготовка проекта: контент(упаковка), описание (правильное формулирование идеи), видео (средство завлечения), бюджет (как обоснованно составить бюджет) и список наград (как составить и оформить).

3. Продвижение проекта: где следует продвигать свой проект, план продвижения.

Заключение

В данном докладе мы постарались показать потенциал краудфандинга. Это достаточно понятный и оптимальный способ развить свой стартап. Прогресс не стоит на месте, и чтобы реализовать свою мечту, нужно искать что-то новое, стараться идти к своему успеху, мыслить материально. Также не забывайте, именно вы – автор, вы – движок проекта и именно от вас зависит успешность реализации проекта. Построить сообщество сторонников вокруг идеи – это удивительный способ создать что-то новое, пробуйте - и у вас всё получится! Двигаться и развиваться или ждать удобного случая, идеальных условий - тоже каждый решает сам. Мы – за развитие!

УДК 364.614.2-055.76

Учащаяся П.А. Овчинникова
Науч. рук. преп. Е.Н Кандыбо
(филиал БГТУ «Витебский государственный технологический колледж»)

ГОСУДАРСТВО ДЛЯ БЛИЗНЕЦОВ: ПРОГРЕСС ИЛИ СОЦИАЛЬНОЕ НЕРАВЕНСТВО?

Сегодня встретить двухместную, а иногда и трехместную коляску можно практически в каждом дворе. По данным Национального статистического комитета Республики Беларусь в нашей стране ежегодно увеличивается число рождений близнецов. Так в 2016 году близнецы родились у 1348 женщин (в 2010 году - у 1027). Например, в 2016 году 328 минчанок родили двойни, у 9 женщин были тройни. Это на 22 двойни и 2 тройни больше, чем в 2015 году. А 10 лет назад в Минске появилось на свет всего 194 двойни. Таким образом, рождение близнецов теперь далеко не редкость.

«Счастье вдвойне» или «счастье в квадрате»... Так говорят о рождении близнецов. Между тем, на счастливую семью ложатся двойные заботы и проблемы по уходу и воспитанию двойняшек. Не на последнем месте оказываются и финансовые затраты. Это и две кровати, и два стульчика для кормления, и двукратный объем средств по уходу и средств гигиены. Даже коляска для двойни и тройни стоит на порядок дороже обычной. Именно по этой причине система государственной поддержки семей с детьми в Республике Беларусь предусматривает ряд социальных выплат и пособий при рождении детей и дальнейшему уходу за ними.

Целью исследовательской работы является получение ответа на вопрос: «Прогрессом или социальным неравенством является поддержка со стороны государства семей с близнецами?». Работа посвящена исследованию системы государственной поддержки семей с близнецами в соответствии с законодательством Республики Беларусь в настоящее и советское время, а также в соответствии с законодательством отдельных зарубежных стран; сравнительному анализу государственной поддержки, оказываемой рассматриваемой социальной группе (семьям с близнецами) и семьям, находящимся в подобном положении (семьям, имеющим детей-погодок).

Кто же такие близнецы? Близнецы - это дети одной матери, развившиеся и появившиеся на свет в результате одних родов через непродолжительное время друг за другом.

В ходе исследования системы государственной поддержки семей с детьми в Республики Беларусь было установлено, что в нашем

обществе существуют социальные группы, положение которых схоже с положением семей с детьми близнецами, но на законодательном уровне не имеющих тех льгот, которые имеют указанные семьи. Это семьи, имеющие детей-погодок.

Дети-погодки - это дети, родившиеся в ситуации, когда вторая беременность наступила, а первому ребёнку ещё не исполнилось 12 месяцев, т.е. максимальная разница между детьми может быть 1 год и 9 месяцев.

С целью выявления положения, в котором находятся семьи, имеющие детей-погодок, в рамках исследования был проведён социальный опрос на тему: «Нуждаются ли семьи, имеющие детей-погодок, в дополнительной государственной поддержке?». В опросе приняли участие респонденты, вообще не имеющие детей (80 человек), и респонденты, которые имеют или ранее имели детей-погодок (20 человек).

Анализ ответов респондентов выявил ряд материальных, психологических и бытовых проблем, с которыми чаще всего сталкиваются (могут столкнуться) семьи с детьми-погодками.

К проблемам материального характера можно отнести:

- 1) затраты на приобретение вещей индивидуального пользования на новорождённого ребёнка и второго ребёнка;
- 2) затраты на питание детей-погодок, т.к. для каждого ребёнка оно будет индивидуальным в силу возраста;
- 3) затраты на платную няню, т.к. государство не оказывает таким семьям социальную помощь в этом плане, как для семей, имеющих близнецов.

К психологическим и бытовым проблемам относятся:

- 1) стресс родителя от нехватки времени, т.к. нужно успеть проследить, приготовить и сделать всё необходимое индивидуально для каждого ребёнка в силу их физиологии в определённом возрасте;
- 2) трудности, возникающие в связи с посещением какого-либо общественного места совместно с двумя детьми;
- 3) отсутствие возможности у родителей на посещение культурно-развлекательных мероприятий (театр, выставки и т.п.);
- 4) проблемы со здоровьем и восстановлением организма женщины после двух беременностей и родов подряд.

Исходя из всех выявленных проблем, можно сделать **вывод**, что семьи с детьми-погодками нуждаются в дополнительной государственной поддержке. Соответственно, в законодательство Республики Беларусь по социальному обеспечению необходимо внести опреде-

лённые изменения и дополнения, которые позволят уровнять в правах семьи с близнецами и детьми-погодками.

Во-первых, необходимо внести дополнение в Постановление Совета Министров Республики Беларусь от 27.12.2012 г. № 1218 «О некоторых вопросах оказания социальных услуг», а именно: добавить новый пункт 35.6, который будет предусматривать оказание помощи в уходе за детьми семьям, имеющим детей-погодок (предоставление няни) в дневное время по рабочим дням до 12 часов в неделю.

Во-вторых, внести изменение в Указ Президента Республики Беларусь от 29.01.2012 г. № 41 «О государственной адресной социальной помощи» (в пункт 2.5), согласно которому семьям, имеющим детей-погодок, будет предоставлена помощь в виде обеспечения продуктами питания новорождённого ребёнка до достижения им двух лет независимо от величины среднедушевого дохода такой семьи.

Посредством внесения таких изменений и дополнений семьи с детьми-погодками приобретут те же льготы, что и семьи с близнецами.

Определённая помощь таким семьям может быть оказана и со стороны общественности, отдельных социальных служб и организаций, например, предоставление скидки семьям, имеющим детей-погодок, на посещение мест, где можно оставить детей для присмотра, игровые комнаты, детские мини-сады или услугу «няни на час».

Кроме того, семьи с детьми-погодками нуждаются в психологической помощи, в получении консультаций психолога по воспитанию детей. Такие программы психологической помощи могут проводиться на базе учреждений социального обслуживания населения.

Таким образом, отвечая на главный вопрос исследовательской работы: «Государственная поддержка семей с близнецами - это прогресс или социальное неравенство?», можно ответить двояко. С точки зрения изменения законодательства о государственной поддержке таких семей исторически - это прогресс (в работе проанализировано современное социальное законодательство и законодательство советского периода). Но если сравнивать семьи с детьми близнецами с положением подобной социальной группы (семьи с детьми-погодками), которая не имеет достаточной государственной поддержки, можно сделать вывод о наличии социальной несправедливости.

УДК 004.9

Учащийся А.В. Пучковский
Науч. рук. преп. Л.М.Левина

(филиал БГТУ «Витебский государственный технологический колледж»)

СОВРЕМЕННЫЙ ПОДХОД ДЛЯ РАБОТЫ С БАЗАМИ ДАННЫХ. ВОЗМОЖНОСТИ ENTITY FRAMEWORK

EntityFramework – это платформа, основанная на ORM технологии, позволяющая работать с информацией из базы данных более абстрактно, посредством классов, не вдаваясь в подробности механизма работы определённой СУБД.

Цель исследования:

- изучить платформу Entity Framework 6.0, её возможности и преимущества, которые она предоставляет разработчикам;
- рассмотреть работу с платформой на примере проекта, использующего возможности данной технологии.

Объект исследования: ORM (Объектно-реляционное сопоставление) технология в библиотеке Entity Framework;

Гипотеза исследования: эффективна ли данная технология при работе с базами данных и в чём её отличие от ADO.NET?

В ходе исследования была изучена данная технология, её возможности, преимущества и недостатки.

Элементы ORM систем:

– Маппинг - проекция структуры классов на структуру базы, для которой требуются:

- проекции сущностей на таблицы;
- «свойства» на поля таблиц;
- связи между классами на связи между таблицами.

– Механизм запросов.

– Механизм обновления и отслеживание измененных объектов.

Преимущества Entity Framework:

- скорость разработки;
- работа с сущностями;
- отслеживание изменений в контексте и сохранение этих изменений в базе данных;
- «быстрая» настройка работы с базой данных;
- простота изучения.

Если говорить о недостатках этой системы, то главный из них – это потеря производительности.

Большинство современных ORM позволяют программисту при необходимости самому задавать код SQL-запросов. Без сомнений, для небольших проектов использование ORM будет куда более оправдано, чем разработка собственных библиотек для работы с БД.

Таким образом, в ходе исследования было установлено, что платформа EntityFramework обеспечивает простое и удобное сопоставление таблиц и классов. А также даёт возможность легкого создания и отслеживания изменений в базе данных, благодаря классу контекста.

УДК 630*813

Учащиеся А.А. Ширяева, М.Ю. Ратников

Науч. рук. преп. М.М. Шнитко, Н.Ф. Овчинникова

(филиал БГТУ «Витебский государственный технологический колледж»)

СТЕПЕНЬ ВЛИЯНИЯ ПОРОКОВ НА СВОЙСТВА ДРЕВЕСИНЫ. СНИЖЕНИЕ ВЕРОЯТНОСТИ ПОЯВЛЕНИЯ ПОРОКОВ ПУТЕМ ЛЕСОХОЗЯЙСТВЕННЫХ И ЛЕСОЗАЩИТНЫХ МЕРОПРИЯТИЙ

В современных условиях все большее внимание уделяется такому комплексному понятию, как качество. При этом важнейшей составляющей системы качества, характеризующей эффективность всех сторон деятельности любого предприятия, является качество продукции.

Производством продукции лесного хозяйства и одновременно сырьем для деревообрабатывающего производства являются круглые лесоматериалы.

Качество круглых лесоматериалов зависит от пороков и физико-механических свойств древесины, на которые в свою очередь оказывают влияние природные и лесоводственные факторы, а именно: качество посадочного материала, условия произрастания лесных культур, форма и состав насаждений и другие.

Цель исследования: установление взаимосвязи между причинами появления пороков в растущем дереве, мерами предупреждения и влиянием их на свойства древесины.

Задачи исследования:

- установить причины появления пороков в растущем дереве;
- установить влияние пороков на свойства древесины;
- выявить мероприятия по предупреждению появления пороков;

-определить практическую значимость влияния пороков на свойства древесины при изготовлении деревянных изделий.

Пороки древесины - это различные недостатки, которые чаще всего снижают ее качество и ограничивают возможности практического использования. Они разнообразны по видам и многочисленны по количеству. Более подробно мы остановимся на пороках, которые возникают в растущем дереве, на тех из них, которые в большей степени оказывают влияние на свойства древесины.

Различают следующие свойства древесины: химические, физические и механические.

Химические свойства древесины непосредственно связаны с ее химическим составом.

Физические свойства – это свойства, которые проявляются при ее взаимодействии с окружающей средой.

Механические свойства древесины – это способность древесины сопротивляться действию внешних механических усилий.

При использовании древесины в качестве конструкционного и поделочного материала, также в технологических процессах обработки древесины проявляются ее механические свойства.

Следовательно, изучая влияние пороков на свойства древесины, был сделан акцент именно на механические и частично на физические свойства древесины.

Начнем с наиболее часто встречающейся группы пороков сучков. Возможными причинами появления сучков является не соблюдение полноты при рубках ухода, а также неправильно подобранная схема смешения лесных культур.

Сучки в основном оказывают отрицательное влияние на свойства древесины, а именно: ухудшают внешний вид, затрудняют механическую обработку; затрудняют склеивание; снижают прочность древесины при растяжении вдоль волокон и изгибе. Но есть и положительное влияние данного порока: они увеличивают прочность при сжатии поперек волокон в радиальном направлении, при условии, что ось сучка совпадает с направлением прикладываемого усилия; а также повышают прочность древесины на скалывание в тангенциальной плоскости; повышают сопротивление раскалыванию в тангенциальной плоскости.

Предупредить появление сучков можно путем создания смешанных насаждений с введением под полог леса подгонных пород. При проведении рубок ухода необходимо контролировать полноту насаждения в зависимости от рубки и состава.

Трещины в растущем дереве появляются из-за резкого перепада температур внутри древесины и на поверхности в зимний период.

Трещины оказывают лишь отрицательное влияние на свойства древесины: снижают выход пилопродукции; уменьшают прочность древесины; способствуют проникновению влаги и спор дереворазрушающих грибов вовнутрь растущих деревьев.

Мероприятий по предотвращению появления пороков данной группы на сегодняшний день не существует.

Пороки формы ствола образуются из-за сильного ветра, большой освещенности в нижней части ствола; также оказывают влияние на образование данных пороков механические повреждения ствола дерева, заселение деревьев дереворазрушающими (трутовыми) грибами, повреждение культур млекопитающими.

Пороки формы ствола: ухудшают выход пиломатериалов; затрудняют использование круглых лесоматериалов по назначению; увеличивают количество древесных отходов при распиловке и лущении; снижают прочность пилопродукции, т.к. являются причиной радиального наклона волокон. Такой вид нароста, как кап ценится в качестве декоративного материала из-за красивой текстуры древесины.

Снизить вероятность возникновения пороков формы ствола можно путем создания ветроупорных опушек из лиственных пород, проведения санитарных рубок, очистки леса от захламленности, сбора и уничтожения плодовых тел трутовых грибов, огораживания лесных культур и защиты от механических повреждений при лесозаготовках.

Причинами появления пороков строения древесины являются сильные ветра, высокая влажность почвы, механические повреждения.

Проанализировав влияние данной группы на свойства древесины, определили главное: они снижают прочность древесины при растяжении и сжатии вдоль волокон, при изгибе; снижают ударную вязкость; увеличивают количество отходов. Но есть у них и положительные стороны: кренивая древесина отличается повышенной твердостью и прочностью при сжатии и статическом изгибе; свилеватость увеличивает сопротивление древесины раскалыванию; повышает ударную вязкость, имеет красивую текстуру.

Проведение санитарных рубок, очистка леса от захламленности, сбор и уничтожение плодовых тел трутовых грибов, защита от механических повреждений, создание ветроупорных опушек, мероприятия по защите лесных культур от млекопитающих - все это позволит предотвратить появление пороков строения древесины.

Грибные поражения появляются в следствии заселения ствола дерева дереворазрушающими грибами. Проникновение спор гриба

происходит через механические повреждения. Заселению спорами гриба также способствуют насекомые.

Данная группа пороков нарушает целостность сортиментов, снижает выход пиломатериалов, отдельные виды пороков снижают механические свойства древесины. Однако под действием грибов древесина изменяет свой естественный цвет, что можно применять для искусственного старения древесины в процессе отделки.

Соблюдая санитарные нормы и правила, проводя санитарные рубки, уничтожая плодовые тела трутовых грибов, можно снизить вероятность появления грибных поражений.

Биологические повреждения возникают при заселении дерева стволовыми вредителями. Наиболее опасным из них является короед-типограф, борьба с которым в последнее время приобрела республиканский масштаб.

Неглубокая и глубокая червоточина нарушает целостность древесины, а при большом количестве ходов снижает ее механические свойства.

Борьба со стволовыми вредителями основывается на соблюдении санитарных норм и правил, очистке лесосек от порубочных остатков, привлечении птиц и млекопитающих и использовании феромонных ловушек.

Древесина в конструкциях и изделиях работает на различные виды нагрузок. Поэтому при выборе древесины как конструкционного материала очень важно знать каким образом имеющиеся на древесине пороки влияют на ее свойства.

Например: доска пола работает на истирание, наличие в ней небольших здоровых сросшихся сучков повышает сопротивление древесины истиранию за счет увеличения ее твердости.

А вот деревянная балка перекрытия работает на растяжение вдоль волокон и каждый имеющийся в ней сучок или трещина снижает ее прочность при растяжении.

По результатам выполненной работы можно сделать следующие выводы.

1. При правильной организации проведения лесохозяйственных и лесозащитных работ можно улучшить не только качество растущего леса, но и качество получаемых лесоматериалов, которые являются сырьем для деревообрабатывающих производств.

2. Владение знаниями о влиянии пороков на свойства древесины позволит правильно выбрать сырьевой материал для повышения эксплуатационных качеств деревянных изделий.

УДК 621.9.06

Учащиеся Ю.А. Шелопухо, В.П. Тарасенков

Науч. рук. преп. Е.Л. Амелишко

(филиал БГТУ «Витебский государственный технологический колледж»)

ВОЗМОЖНОСТЬ ИСПОЛЬЗОВАНИЯ ГИБКОЙ СВЯЗИ В МЕХАНИЗМАХ ПОДАЧ МЕТАЛЛОРЕЖУЩИХ СТАНКОВ

Гибкая связь – условный термин, объединяющий цепные, ременные и тросовые передачи. В настоящее время они используются широко из-за дешевизны изготовления и удобства эксплуатации. Но зубчатые передачи используются чаще: они гораздо долговечнее, выдерживают большие нагрузки, чем оправдывают трудоемкость и стоимость своего изготовления.

Цель исследования: выяснить, можно ли заменить дорогостоящую передачу в металлорежущем станке на более дешевую, т.е. восстановить вертикально-сверлильный станок с наименьшими затратами.

Исходные данные:

- 1) Дрель, обеспечивающая вращательное движение сверла.
- 2) Станина от вышедшего из строя настольного вертикально-сверлильного станка.

Сверление - вид обработки металлов резанием, при котором и движение подачи, и главное вращательное движение задается инструменту (сверлу). Соответственно, основной задачей является выбор механизма подачи (движения сверла вверх-вниз).

В классическом вертикально-сверлильном станке используются либо реечная зубчатая передача, либо передача винт-гайка. Обе передачи достаточно дороги в изготовлении, требуют специального оборудования: фрезерование либо долбление зубчатого колеса и рейки для реечной передачи; нарезание резьбы для передачи винт-гайка.

Решает задачу преобразования вращательного движения в поступательное тросовая передача (канатная).

Такие передачи просты по конструкции, сравнительно дешёвые, позволяют передавать движение на расстояние, плавно работают.

К их недостаткам относят непостоянство передаточных чисел из-за скольжения в передаче, повышенные нагрузки на опоры из-за необходимости натяжения ветвей троса и пониженная точность.

Трос изготавливается плетением из стальной оцинкованной проволоки.

Первым шагом стало приобретение 1 метра каната диаметром 1,5 мм, состоящего из 42 проволочек, левая крестовая свивка с орга-

ническим сердечником. Стоимость данной покупки составила 1 руб.ВУ.

Следующим шагом стал расчёт диаметра барабана, на который наматывается трос. Используем формулы справочной литературы.

$$D_{\min} = (\beta - 1) \cdot d_m, [1], \text{ с135}$$

где d_m - диаметр троса, мм; β - коэффициент, зависящий от режима работы передачи.

Повышенные нагрузки не предвидятся, выбираем спокойный режим работы.

$$\beta = (8 \dots 12) [1], \text{ с135}$$

$$\text{Примем } \beta = 9, \text{ тогда } D_{\min} = (9 - 1) \cdot 1,5 = 12(\text{мм}).$$

Приобретаем шпильку диаметром 12 мм, чтобы использовать ее в качестве ведущего звена передачи. Стоимость шпильки- 2 руб.ВУ.

Для практических целей достаточно знать значение разрывного усилия и рабочую крепость троса.

Разрывное усилие:

$$R = f \cdot c^2, [1], \text{ с 136}$$

где f - эмпирический коэффициент, определяющийся по формуле:

$$f = (650 - 0,75c) / 100 \quad [1], \text{ с 136}$$

где c – длина окружности троса ($C = 2\pi r$). [1], с 136

$$r = d_m / 2;$$

$$r = 1,5 / 2 = 0,75 \text{ мм};$$

$$C = 2 \cdot 3,14 \cdot 0,75 = 4,71 \text{ мм};$$

$$f = (650 - 0,75 \cdot 4,71) / 100 = 6,46$$

Определяем разрывное усилие R :

$$R = 6,46 \cdot (4,71)^2 = 143,3(\text{Н})$$

Определяем рабочую крепость (допускаемое напряжение) – нагрузка, при которой трос может работать в данных условиях в течение продолжительного промежутка времени без нарушения целостности отдельных элементов и всего троса.

$$P = R / n, \quad [1], \text{ с 137}$$

где n - коэффициент запаса прочности, для данных условий $n = 4$

$$P = 143,3 / 4 = 35,8 \text{ (Н)}$$

Масса дрели $\approx 2,5$ кг, масса штатива, который будет опускаться и подниматься вместе с дрелью $\approx 0,7$ кг. Т.е. масса, которая подлежит подъёму и опусканию:

$$m = (2,5 + 0,7) = 3,2 \text{ (кг)}$$

Вес конструкции:

$$P = mg;$$

$$P = 3,2 \cdot 10 = 32 \text{ (Н)}$$

$32 \text{ Н} < 143,3 \text{ Н}$ - разрыв тросу не грозит.

$32 \text{ Н} < 35,8 \text{ Н}$ – возможность длительной работы обеспечена.

Расчёт на прочность выполнен, себестоимость передачи 3 руб.ВУ.

Для сравнения: речная передача обошлась бы в 10-50 руб.ВУ, а передача винт-гайка обошлась бы примерно в 50-200 руб.ВУ (цена варьируется в зависимости от габаритов, материалов и качества самой передачи). Если нет разницы, зачем платить больше?

Практическая часть: изготовление вертикально-сверлильного станка.

Как уже было отмечено, у нас имелась станина и дрель.

Расчеты показали, что гибкую связь использовать допустимо, значит, приобретённый трос и шпилька могут обеспечить поступательное движение дрели.

Для повышения устойчивости станка из фанеры было изготовлено дополнительное основание.

Далее следовало изготовление подвижного суппорта из пластин МДФ, в котором и будет закрепляться дрель и устанавливаться подъёмный механизм.

Возникла необходимость в изготовлении задней направляющей для обеспечения строго поступательного движения суппорта.

Установка гибкой связи. В суппорт, используя подшипники, вставляем вал-шпильку, предварительно намотав трос. Верхняя часть троса жестко крепится к соединительному узлу между направляющими, который также увеличивает жесткость конструкции, нижняя - к натяжному устройству.

Натяжное устройство представляет собой винт с изогнутым концом, оно находится внутри нижнего основания. Натяжение производится с помощью гайки. Поворотное колесо соединяется с ведущим звеном тросового механизма.

Дрель для жёсткого её закрепления стягивается шпилькой.

Провода от дрели идут на димер, регулирующий частоту оборотов, затем на тумблер и затем в розетку 220 В.

Настольный вертикально-сверлильный станок готов, осталось загрузить деревянный корпус для защиты его от влаги и улучшения его эстетического вида.

Вывод: в металлорежущих станках не промышленного назначения, неработающих в условиях повышенных нагрузок гибкая (тросовая) связь вполне может быть использована.

ЛИТЕРАТУРА

1. Анурьев В.Н. Справочник конструктора-машиностроителя. М., 1982.
2. Дунаев П.Ф., Леликов О.П. Конструирование узлов и деталей машин. М, 1985.
3. Обработка металлов резанием. Металлорежущий инструмент и станки. Горбунов Б.И. М, 1992.
4. Тросовая передача [Электронный ресурс] // МЭЛП. – Режим доступа: http://www.melp.ru/prod03_1.html?yclid=1240885570580259222. – Дата доступа: 01.02.2019.
5. Расчёт механических передач [Электронный ресурс] – Режим доступа: <http://argovita.ru/blog/2009/09/> – Дата доступа: 15.02.2017.

УДК 681.513.2

Учащийся И.В. Рыжиков
Науч. рук. преп. С.А. Никулин
(филиал БГТУ «Гомельский государственный политехнический колледж»)

ЛАБОРАТОРНАЯ УСТАНОВКА ДЛЯ ПЛК MITSUBISHI ELECTRIC ALPHA-2

Целью проекта является создание установки, позволяющей продемонстрировать работу систем автоматизации на базе программируемого контроллера (ПЛК).

Установка будет дополнительным модулем к учебному лабораторному стенду НТЦ-12.100 «Автоматизация производственных процессов и автоматика» (рисунок 1).

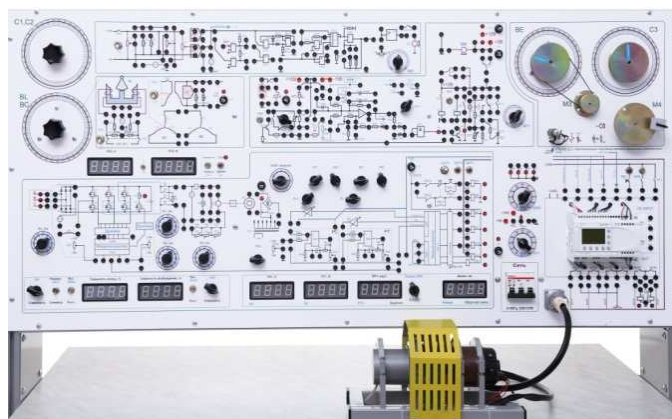


Рисунок 1 - Общий вид стенда НТЦ-12.100 (НТЦ-09.12)

В лабораторном стенде установлен ПЛК MITSUBISHI AL2-14MR-D. Элементная база стенда позволяет наглядно демонстрировать только автоматизацию работы электропривода постоянного тока. Наличие установки расширит возможности лабораторного стенда.

Назначение разрабатываемой установки – визуализация работы автоматических систем.

Готовым решением визуализации работы является лабораторный стенд НТЦ-09.50 "Средства автоматизации и управления". Однако стоимость стенда составляет порядка 7000 рублей.

Целесообразней использовать ПЛК имеющегося стенда НТЦ-12.100 с возможностью подключения к нему необходимых датчиков и исполнительных устройств. Для расположения дополнительного оборудования создаем данную установку. Это позволит собирать системы автоматического управления на базе имеющегося ПЛК.

Установка может быть использована при реализации образовательных стандартов по специальностям:

- 2-36 03 31 «Монтаж и эксплуатация электрооборудования»;
- 2-39 03 02 «Программируемые мобильные системы»;
- 2-53 01 01 «Автоматизация технологических процессов и производств».

Назначение и конструкция установки

Учебный лабораторный стенд (установка) предназначен для использования в качестве учебного оборудования. Назначение разрабатываемой установки – демонстрация работы систем автоматического управления с применением микроконтроллера модели MITSUBISHI-AL2-14MR-D. При адаптации коммутационных программ возможно подключение других моделей контроллера.

Установка позволяет демонстрировать работу системы автоматики «Смеситель», «Автоматические двери», «Светофор», «Равномерная загрузка трех потребителей».

Конструктивно стенд состоит из корпуса, в котором установлено необходимое электрооборудование, и лицевой панели.

Лицевая панель установки показана на рисунке 2.

Стенд питается от однофазной сети переменного тока 220 В, 50 Гц. Для сборки схемы используются изготовленные перемычки необходимой длины.

Пример использования

Рассмотрим наиболее простой пример использования установки «Светофор».

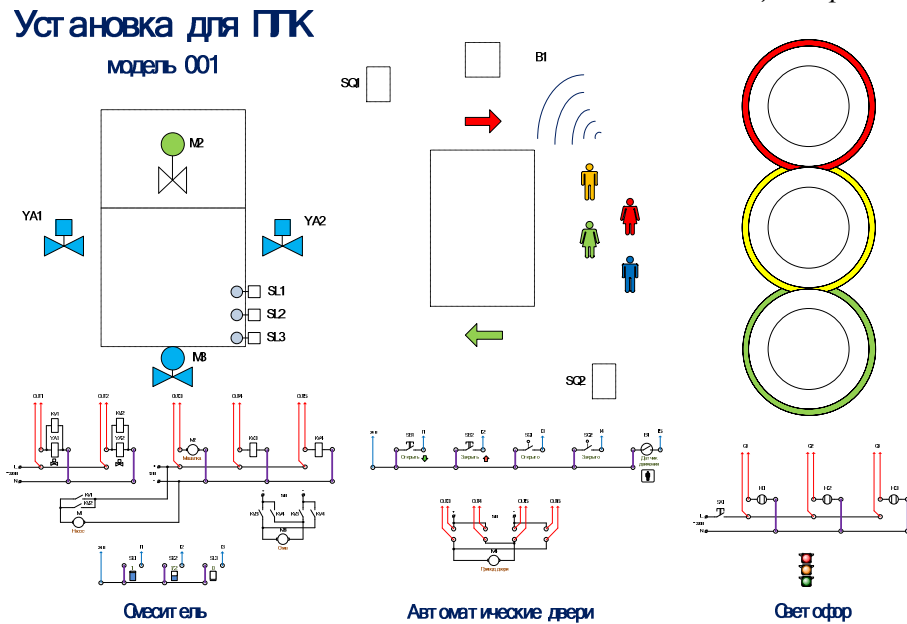


Рисунок 2- Лицевая панель установки для ПЛК

Алгоритм работы светофора:

- 1) горит красный цвет, время 5 с.
- 2) горят одновременно красный и желтый цвета, время 2 с.
- 3) горит зеленый цвет, время 5 с.
- 4) зеленый цвет мигает три раза.
- 5) горит желтый цвет, время 2 с.

Цикл повторяется.

Контроллеры Alpha 2 программируется при помощи программы SW0D5-ALVLS-EU или при помощи клавиш контроллера. На рисунке 4 представлена коммутационная схема для управления светофором.

Коммутационная программа отражает алгоритм (логику) работы схемы управления. Графически реализованная схема на типовых блоках и соединительных элементах в дальнейшем преобразуется в логическую структуру контроллера.

Для имитации работы светофора используем три источника света HL1 – HL3 напряжением 220 В, имеющих три цвета: красный, желтый, зеленый. Для прерывания работы схемы схема содержит переключатель SA1.

На рисунке 4 представлена часть схема управления работой светофора. Основным элементом схемы является программируемый контроллер MITSUBISHI AL2-14MR-D. Контроллер установлен в лабораторном стенде НТЦ12.100. В стенде имеются одиночные разъемы для подключения входов и выходов ПЛК.

На схемах установки показаны номера входов и выходов для подключения внешних устройств.

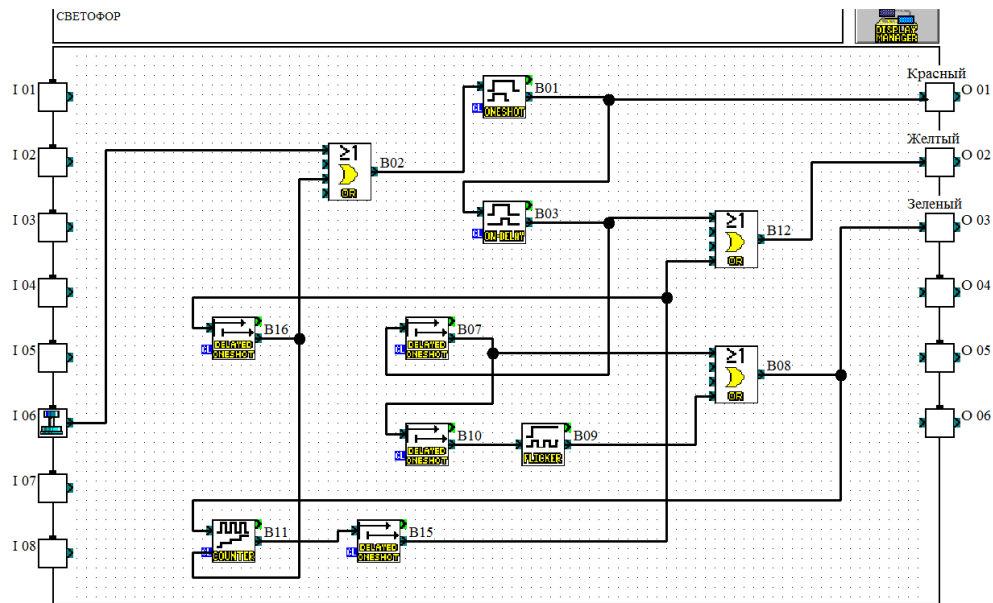
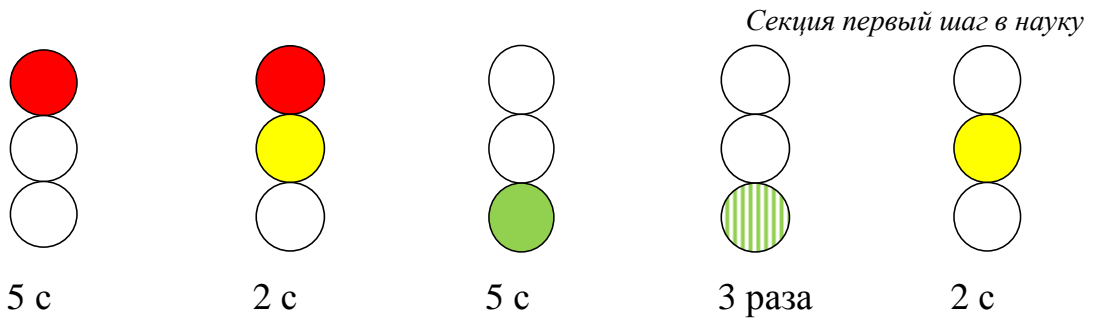


Рисунок 3 – Коммутационная схема управления светофором

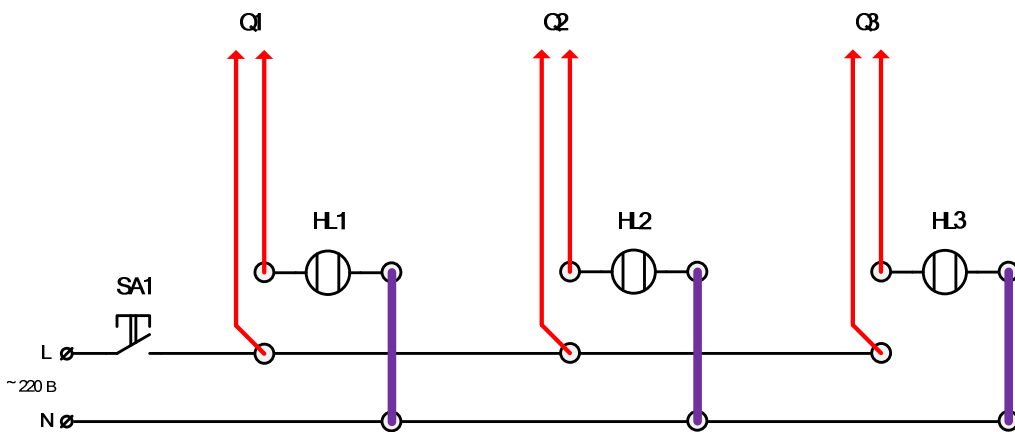


Рисунок 4 – Схема управления работой светофора

Примерная стоимость затрат при изготовлении стенда составила 56 руб. При условии использования комплектующих, имеющих в лабораториях колледжа стоимость сборки установки будет еще ниже.

По адресу <https://youtu.be/xsiDXds9Hjg> расположен демонстрационный ролик работы установки (для понимания работы «Смесителя» требуется предварительное пояснение алгоритма работы).

ВЫРАЩИВАНИЕ ПОСАДОЧНОГО МАТЕРИАЛА ЕЛИ ГОЛУБОЙ

Ель голубая часто используется как декоративное растение. Представители этого вида украшают стены Брестской крепости, растут у Кремлёвской стены в Москве, являются незаменимым декоративным элементом на многих садовых участках и в коттеджных посёлках. В последнее время использование хвойных пород деревьев и кустарников, в частности нетрадиционных оттенков (голубых, серебристых, жёлтых и др.), приобрело особую актуальность. Можно заметить, как разнообразился ассортимент питомников, зеленхозов, фермерских хозяйств этими породами. Такое широкое предложение товара обусловлено большим спросом на подобную продукцию. Люди уходят от устоявшихся стандартов и сложившихся стереотипов в оформлении своих земельных владений. Использование вечнозелёных посадок оживляет пространство даже в зимний период, придавая благородный и величественный облик участку. Кроме того, из преимуществ выращивания ели голубой можно выделить её неприхотливость, выносливость, тене- и морозоустойчивость, долговечность.

Желание людей подчеркнуть индивидуальность и создать неповторимый ландшафтный дизайн порождает спрос на хвойные растения в целом и на ель голубую в частности. Саженцы этого растения можно встретить как на предновогодних ёлочных базарах, так и на весенне-осенних сельскохозяйственных ярмарках. Анализ прейскурантов, выставляемых хозяйствами, продающими ель голубую, показал, что цены на этот товар не маленькие. Это обстоятельство подчёркивает *актуальность* выбранной тематики исследования.

Однако выращивание посадочного материала требует научного подхода.

Таким образом, *целью исследования* является выбор способов разведения ели голубой и подбор грунта для выращивания сеянцев ели голубой. *Задачи исследования:*

- изучение биологических особенностей ели голубой,
- обзор литературы по способам выращивания сеянцев и саженцев ели голубой,
- заготовка черенков и семян ели голубой,
- подбор состава грунта,
- выращивание сеянцев и саженцев ели голубой,
- статистическая обработка и анализ результатов исследований,

– разработка рекомендаций по выращиванию посадочного материала ели голубой.

Объект исследования: ель голубая (лат. *Picea pūngens*)

Предмет исследования: способы выращивания и использование различных видов грунта при выращивании посадочного материала ели голубой.

Методы исследования:

- изучение научной литературы,
- наблюдение и учёт полученных данных,
- систематизация накопленного материала,
- биометрический анализ полученных результатов.

Во многом успехи любых исследований зависят от планирования и использования методов статистической обработки полученных данных, т. е. от биометрии.

В литературе предлагается три традиционных способа выращивания посадочного материала ели голубой: черенкование, прививка, посев семян.

Первые два способа обеспечивают точное повторение материнских признаков у посадочного материала. А при посеве семян в потомстве будет наблюдаться расщепление признаков, т.е. появятся всходы как с голубой, так и с зелёной окраской хвои. Истинный цвет по данным селекционеров, проявляется только на второй год жизни. Проведённые в работе исследования с посевом подтвердили, что в первый год жизни все сеянцы были однородно окрашены.

Для выбора наиболее оптимального способа выращивания посадочного материала ели голубой авторами работы было выбрано два варианта: черенкование и проращивание семян, т.е. выращивание сеянцев. Прививка не была включена в план исследования вследствие того, что этот способ требует определённых навыков. Для обоих способов в качестве материнского дерева была выбрана ель, с насыщенным сизо-голубым оттенком хвои и богатой пышной кроной дерева.

Черенкование проводилось в два этапа: 16 января и 16 февраля 2019 года. Проводилось сравнение влияния периода посадки черенков на их приживаемость. В качестве черенков отбирались боковые побеги методом обдиранья веточек. При этом на черенках оставалась «пятка» со слоем старого дерева. Это позволяет черенку нормально расти и не загнивать. Черенки в дни сбора (16 января и 16 февраля) были сразу помещены в раствор стимулятора корнеобразования «Корневин». Предварительно на нижней части черенков на высоту 1,5-2 см была обрезана хвоя. Спустя 3 часа черенки были дополнительно припудрены стимулятором корнеобразования и высажены в

грунт, состоящий из смеси торфа, песка и верхнего слоя земли, взятой из-под материнского дерева.

Для посадки использовались пластиковые контейнеры. После помещения в грунт черенки были политы и укрыты полиэтиленовой плёнкой с небольшими отверстиями для воздухообмена. Спустя 4 недели после посадки в обоих случаях был обследован посадочный материал на предмет приживаемости. С 13 черенков из 15, высаженных 16 января опала хвоя, а сами веточки стали сухими и ломкими, что свидетельствовало об их отмирании и низком проценте приживаемости (13 %). 10 из 15 черенков (67 %), отобранных и высаженных 16 февраля спустя 4 недели сохранили хвою, хотя на более мелких её цвет изменился и стал слегка бурым и более тусклым. Это можно объяснить предвесенней активизацией обменных процессов и, в том числе, началом сокодвижения в растениях.

Вторая часть исследования касалась выбора грунта для посева семян. Изучение литературы показало, что для посева семян различными авторами рекомендуются разнообразные составы грунта и нет единого мнения по этому вопросу. Это обстоятельство ещё раз подчёркивает актуальность выбранной тематики. В связи с этим для исследований были выбраны некоторые из рекомендуемых субстратов, которые были помещены в отдельные пластиковые контейнеры.

Шишки для отбора семян были собраны 10 февраля. На 5 дней они были помещены возле отопительного радиатора, благодаря чему подсохли, раскрыли кроющие чешуи, и семена без труда были извлечены лёгким постукиванием шишкой о стол. Затем семена для обеззараживания были помещены на несколько минут в розовый раствор перманганата калия, после чего промыты и высушены. Было отсчитано 5 порций семян по 100 штук в каждой.

Каждая партия семян была посеяна в заранее подготовленный грунт определённого состава. Всходы появились на 8 день. Спустя один месяц после посева семян, т.е. 16 марта 2019г. было подсчитано количество проростков в каждом контейнере. Результаты представлены в таблице:

Таблица 1 – Влияние состава грунта на всхожесть семян и выживаемость сеянцев ели голубой

№ п/п	Состав грунта	Число взошедших и выживших сеянцев
1	питательный грунт + почва из-под ели = 2:1	56
2	питательный грунт + почва из-под ели + песок = 1:1:1	39
3	питательный грунт + торф + песок = 1:1:1	51
4	песок	33
5	торф	40

Сравнение влияния состава грунта на всхожесть семян и выживаемость сеянцев ели голубой показало наибольшую эффективность при использовании субстрата под №№ 1 и 3. Оба состава содержат питательный грунт, что обеспечивает молодые всходы элементами питания, и содержат компоненты, дающие кислую реакцию почвенной среды (почва из-под ели и торф), что также благоприятно для развития хвойных деревьев. Наименьший показатель всхожести наблюдался в чистом песке — 33%.

Сравнивая способы разведения ели голубой, можно сделать выводы, что наиболее эффективным в процентном отношении является черенкование ели в феврале (67%) против максимальной всхожести семян (56%). Черенкование выигрывает по другим показателям:

- 100%-ое повторение у посадочного материала материнских признаков (цвет хвои),

- достижение желаемого результата за более короткий срок.

К преимуществам семенного размножения можно отнести получение большого количества посадочного материала с наименьшими трудовыми и материальными затратами на подготовку грунта.

Выводы, которые можно использовать как *рекомендации* при выращивании посадочного материала ели голубой:

- черенкование — более эффективный способ размножения ели голубой по сравнению с семенным, в том случае, если необходимо получить незначительное количество посадочного материала,

- при массовом разведении саженцев ели голубой лучше использовать семенное размножение,

- наилучшим субстратом для семенного размножения служит смесь питательного грунта и почвы, взятой из-под материнского дерева ели голубой, в отношении 2:1.

УДК 631.53.011

Учащиеся К.Э. Тесленко, А.А. Минкова

Науч. рук. преп. О.С. Луговская

(филиал БГТУ «Гомельский государственный политехнический колледж»)

ВЛИЯНИЕ ТЕМПЕРАТУРЫ НА ПРОРАСТАНИЕ СЕМЯН СОСНЫ ОБЫКНОВЕННОЙ

К числу основных задач, стоящих перед лесным хозяйством нашей республики, относятся улучшение породного состава и качества лесов, повышение их продуктивности. Решающую роль в осуществлении этих задач играет искусственное лесовосстановление и лесоразведение. Необходимым условием для этого является обеспечение лесо-

культурных работ семенами деревьев и кустарников с лучшими наследственными свойствами и высокими посевными качествами.

Основная цель определения качества семян – установление пригодности их для посева. Одним из способов определения качества семян является определение их всхожести.

Прорастание семян – сложный биологический процесс, который ещё остаётся во многом непознанным. Сложность процесса прорастания семян определяется не только тем, что он сопряжен со множеством последовательно проявляющимися морфолого-анатомическими и физико-биохимическими явлениями, но и тем обстоятельством, что на него оказывают влияние многочисленные внешние факторы. Все воздействия, регулирующие прорастание, называются факторами прорастания: они, замедляя или ускоряя биохимические превращения, вызывают глубокие изменения в физиологических процессах, влияя на всхожесть и скорость прорастания семян. Важными факторами, влияющими на прорастание семян, являются влажность, температура, аэрация, свет.

Актуальность: задачей современной агротехнологии является получение качественных семян, дружных всходов, сильных и высокопродуктивных насаждений, изучения темы заключается в том, что данная работа будет полезна лесоводам, садовникам, и всем, кто хочет посадить данное растения.

Гипотеза: мы предполагаем, что на прорастание семян влияют такой факторы как температура.

Целью исследования является изучение влияния различных температур на прорастание семян сосны обыкновенной.

Задачи:

1. Дать характеристику процессу прорастания семян;
2. Выделить основные условия, влияющие на прорастание семян;
3. Установить на практике влияние конкретных температур на прорастание семян;
4. Получить сравнительную характеристику по таким показателям, как всхожесть и энергия прорастания.
5. Сформулировать вывод о влиянии конкретных температур на прорастание семян.

Методика исследования. Всхожесть семян – это их способность образовывать нормально развитые в определенный срок проростки. Всхожесть фиксируют путем учета нормально развитых проростков, в том числе предварительного подсчета быстро и дружно проросших семян (энергия прорастания), а также учета других категорий семян.

Энергия прорастания – это способность семян давать нормальные проростки за установленный государственным стандартом срок, более короткий, чем для определения всхожести (для семян сосны обыкновенной – равен семи дням). Этот показатель определяют одновременно со всхожестью, он характеризует дружность прорастания семян. Методика определения всхожести семян и обработка полученных результатов проводятся в соответствии с ГОСТ 13056.6-75 «Семена деревьев и кустарников. Метод определения всхожести», отбор образцов и измерение их массы – согласно ГОСТ 13056.1-67.

Объектом исследования является сосна обыкновенная. Было взято на проращивание для определения энергии прорастания и всхожести, 3 пробы семян сосны обыкновенной по 100 штук каждая. Семена перед этим проверили на пригодность для проращивания, поместив в ёмкость с водой (те, которые начали постепенно тонуть – пригодны). Далее каждая партия была замочена на 24 часа в воде согласно ГОСТ 13056.6-75 «Семена деревьев и кустарников. Метод определения всхожести».

После этого семена разложили в 3 чашки петри на влажное «ложе» так чтобы они не соприкасались друг с другом. Каждый образец поместили в различные температурные условия. 1-й оставили при температуре 24°C, 2-й оставили при температуре 15°C, 3-й поместили в помещение где температура была 8°C (остальные факторы: влажность, свет были одинаковые).

Срок проращивания установлен – 15 дней. Учет проросших семян сосны проводят на 7-й, 15-й дни (началом проращивания считают день, следующий за днем раскладки семян). Проросшие семена можно классифицировать как нормально развитые проростки и как различные категории ненормально развитых проростков. К нормально проросшим относятся семена с длиной корешка не менее длины семени.

Результаты исследования. Посевные качества семян установлены в ГОСТ 14161-86 «Семена хвойных древесных пород. Посевные качества. Технические условия», согласно его требованиям необходимая всхожесть семян сосны обыкновенной должна быть равна 80-85%.

Энергию прорастания определяют на 7 день постановки опыта, а всхожесть на 15 день.

При подсчете энергии прорастания считают нормально проросшие семена, если имеются загнившие семена, их удаляют и подсчитывают, не проросшие и ненормально проросшие семена оставляют для дальнейшего проращивания. При определении всхожести считают все проросшие и не проросшие семена. Вычисляют процент всхожести семян по каждой пробе, рассчитывают процент всхожести и энергии прорастания семян.

Рассчитывается по формуле простой пропорции, данные заносятся в таблицу 1 и таблицу 2. Для наглядности показатели всхожести отражены в диаграмме (рисунок 1).

Таблица 1 – Определение энергии прорастания семян

№ пробы	Количество в пробе семян	% энергии прорастания
1	100	79
2	100	60
3	100	3

Таблица 2 – Определение всхожести семян

№ пробы	Количество в пробе семян	% всхожести семян
1	100	92
2	100	80
3	100	7

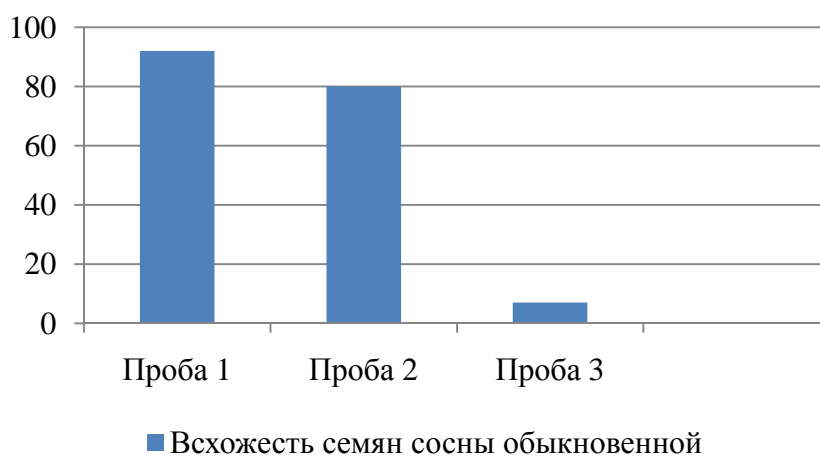


Рисунок 1 – Показатели всхожести семян сосны обыкновенной

Из полученных результатов видно, что худшие показатели всхожести имеет проба №3 (рисунок 1). Её показатель – 7 %, что не отвечает требованиям ГОСТ. Пробы №№ 2 и 3 имеют показатели 92 % и 80 % соответственно. Анализируя показатели всхожести семян из проб можно предположить, что на прорастание семян большое влияние оказывает температура.

Выводы. Приведенные данные подтверждают, что семена, проращиваемые при низких температурах, обладают худшими показателями всхожести и энергии прорастания по сравнению с семенами, оставленными на проращивание в оптимальных, или близких к тому условиях. Из выше сказанного следует вывод, что для успешного прорастания семян кроме воды и кислорода, необходима благоприятная температура. Если семенам достаточно воды и воздуха, но не хватает тепла, они не прорастут и, в конце концов, погибнут.

УДК 634.513

Учащаяся Н.С. Нестеренко

Науч. рук. преп. В.А. Копилец

(филиал БГТУ «Полоцкий государственный лесной колледж»)

УЧЕБНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА «JUGLANS MANDSHURICA – НОВОЕ В МЕДИЦИНЕ»

Сейчас в озеленение можно увидеть различные экзотические растения. Но не каждое растение может похвастаться своей ценностью. А вот я нашла такое растение- Орех маньчжурский! Увидев эту красоту и отведав вкусных орешков, решила исследовать. Ведь это растение не только декоративное внешне, но и обладает лечебными свойствами! Маньчжурский орех является близким братом грецкого ореха, однако, менее популярен по причине своего редкого произрастания на нашей земле. Немногие знают, что компактные плоды этого дерева изобилуют целебными веществами и давно применяются в рецептах как традиционной, так и народной медицины, косметологии и кулинарии.

Целью моей работы стало изучение лечебных свойств и использования Ореха маньчжурского в народной медицине.

Задачи:

1. Выявить практическую ценность Ореха маньчжурского;
2. Выявить лекарственные свойства Ореха маньчжурского.
3. Выявить применения Ореха маньчжурского в народной медицине.
4. Произвести финансовый расчет экономии от использования лечебных свойств Ореха маньчжурского в альтернативу лекарственным препаратам из аптеки.

Актуальность работы: в Белоруссии наблюдается дефицит йода в организме. А это может привести ко многим проблемам со здоровьем: снизится иммунитет, нарушится гормональный баланс, возникнут проблемы с сосудами и сердцем, ухудшится работа мозга. Применение йодированной соли недостаточно для восполнения йода. Можно применять лекарственные препараты на основе йода, но он в неорганическом виде плохо усваивается.

Организмом лучше всего усваивается йод органического происхождения, который поступает вместе с продуктами питания. Больше всего йода содержится в морепродуктах и морской рыбе, но они доступны далеко не всем. Применение средств на основе Ореха маньчжурского обеспечит организм необходимым количеством органического йода, который содержится практически во всех частях растения.

Следовательно, широкая интродукция этого растения поможет в решении острой проблемы дефицита йода.

Для достижения своих целей нами был использован ряд литературных источников и сеть Интернет. Экологические и биологические особенности Ореха маньчжурского была изучена в учебной литературе по учебной дисциплине «Дендрология». Практическая ценность этого растения описана во многих источниках. Отмечают и орех маньчжурский как противовоспалительное, противомикробное, бактерицидное, витаминное, общеукрепляющее, вяжущее, повышающее аппетит средство. В народной медицине используют листья для приготовления растворов, суспензий и мазей. Незрелые плоды принимают при гастритах и язвах желудка.

Методика исследования: для достижения целей исследования нами были использованы следующие методы: изучение литературы, сравнение, анализ, систематизация и обобщение полученных данных. Литература по теме исследования была изучена в библиотеке и в сети Интернет.

В результате моего исследования я пришла к заключению, что *Júglansmandshúrica* — действительно ценное растение, которое может приносить огромную пользу человеку. Его практическое значение в озеленении населенных пунктов. Это растение является естественным природным антибиотиком, обладая мощнейшим противомикробным действием, высвобождает в воздух фитонциды.

Установлено, что все части растения: листья, кора, плоды обладают лекарственными свойствами и традиционно применяются в народной медицине. Отмечено использование ореха маньчжурского как противовоспалительное, противомикробное, бактерицидное, витаминное, общеукрепляющее, вяжущее, повышающее аппетит средство. Различные средства, приготовленные на основе этого растения, применяются при профилактике и лечении многих заболеваний: аллергии, кровоточивости десен, лечении гнойных ран, ревматизме, экземе и лишаях, язвенных болезнях желудочно-кишечного тракта, для укрепления иммунитета.

Использование предложенных рецептов приготовления лечебных средств на основе Ореха маньчжурского поможет сэкономить семейный бюджет. Широкая интродукция и применение Ореха маньчжурского поможет в решении проблемы дефицита йода в организме людей нашего региона.

УДК 630*443.3

Учащиеся С.В. Морозов, О.Г. Хитрик

Науч. рук. Е.В. Башлыкова

(филиал БГТУ «Полоцкий государственный лесной колледж»)

**УЧЕБНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА
«PHLEBIOPSISGIGANTEA – СУПЕР СОВРЕМЕННОЕ
СРЕДСТВО В БОРЬБЕ С КОРНЕВОЙ ГУБКой»**

Корневая губка (*Heterobasidion annosum*) на протяжении 2-х столетий активно развивается на всей территории Республики Беларусь. Общая площадь очагов, которой составила на начало 2018 года 123685 га. При поражении корней у зараженных деревьев обнаруживаются признаки ослабления. Они проявляются, прежде всего, в снижении ростовых процессов по высоте и диаметру ствола, в более слабом развитии ассимиляционного аппарата. В результате дальнейшего отмирания корней крона становится изреженной, опадает значительная часть 2-3-летней хвои, хвоя текущего года формируется укороченной в виде небольших пучков. Такие деревья сильно отстают в росте, часто заселяются стволовыми вредителями и отмирают, что приводит к потере деловой древесины. В настоящее время решение проблемы с корневой губкой решается рубками промежуточного пользования, выборочными и сплошными санитарными рубками, что значительно уменьшает запас получаемой древесины и снижает ее товарность. Данные мероприятия направлены на уборку уже пораженной древесины, но не позволяют предотвратить развитие очагов корневой губки. Огромную роль в выполнении данной задачи играет применение современных способов борьбы с корневой губкой, в частности применение отечественного биопрепарата «Флебиопин».

Цель работы: определение эффективности использования биопрепарата «Флебиопин» для защиты сосновых насаждений от корневой губки на базе филиала БГТУ «Полоцкий государственный лесной колледж» учебно-опытный лесхоз.

В результате исследования выделены преимущества использования препарата «Флебиопин» для защиты сосновых насаждений от корневой губки.

Степень внедрения – используется в учебных целях как методический материал при изучении раздела «лесная фитопатология».

Прежде чем сосну массово атаковал короед, на протяжении нескольких лет ее защитные функции ослабляли другие вредоносные факторы, и самым серьезным из них была и остается корневая губка. Она намного упрощает работу жука, подготавливая дерево к его напа-

дению. Чтобы спасти сосну, ученые разработали первый отечественный биопрепарат.

Метод биологической защиты сосновых от корневой губки используется в европейских странах, таких как Великобритания, Германия, Финляндия и Швеция. Однако стоимость импортных препаратов достаточно высока. Именно поэтому, против корневой губки в лесном хозяйстве Беларуси, применяется лишь один метод – рубки различной направленности. Но ее очаги продолжают расти, что доказывает невысокую эффективность такого метода.

На наш взгляд решить данную проблему позволит применение отечественного препарата «Флебиопин» - средство, основанное на биологических особенностях местных изолятов гриба *P. gigantea* для защиты сосновых насаждений от корневой губки по эффективности не уступающее зарубежным аналогам.

На основании наблюдений в насаждении и эксперимента проводимого в лабораторных условиях можно с уверенностью сказать о том, что препарат «Флебиопин» выполняет двойное действие:

- ✓ уменьшает количество инфекции и скорость ее распространения, снижает объем доступного для корневой губки субстрата (подтверждено лабораторными наблюдениями);

- ✓ способствует биоутилизации древесины пней и крупных корней, что предотвращает возможность передачи заболевания от больной корневой системы к здоровой (наблюдения в насаждении – кв.28).

Кроме этого «Флебиопин», как отечественный препарат, имеет ряд преимуществ:

- более адаптирован к местным условиям;
- экономичен(одна ампула на 1га);
- доступен по цене(стоимость 1 ампулы около 4 рублей);
- качественен(доказано исследованиями);
- не несет вреда другим организмам(чистый состав);
- прост в использовании (ранцевые опрыскиватели или установка на харвестеры специальных устройств для нанесения препарата на спиленный пенё).

Сегодня в мире созданы и применяются биопрепараты способные сдерживать развитие корневой губки, но в связи с высокой стоимостью и фитосанитарными рисками, данные препараты не применяются в лесном хозяйстве нашей страны. С этой целью предлагается разработка отечественного биопрепарата на основе местных штаммов гриба *P. gigantea* для ограничения вредоносности корневой губки.

УДК 630*377.45

Учащаяся К.М. Зародий
Науч. рук. А.Н. Четырбок

(филиал БГТУ «Полоцкий государственный лесной колледж»)

**УЧЕБНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА
«ИЗУЧЕНИЕ ОСНОВНЫХ ТЕХНИЧЕСКИХ
ХАРАКТЕРИСТИК, СРАВНИТЕЛЬНЫЙ АНАЛИЗ
ЭКОНОМИЧЕСКИХ ПОКАЗАТЕЛЕЙ
ВАЛОЧНО-СУЧКОРЕЗНО-РАСКРЯЖЕВОЧНЫХ МАШИН
(ХАРВЕСТЕР) РАЗЛИЧНЫХ ПРОИЗВОДИТЕЛЕЙ»**

Харвестер – вид специальной лесозаготовительной техники, которая осуществляет комплекс работ по валке деревьев, обдирке сучьев и распилке ствола на сортименты.

Один подобный механизм дает возможность заменить несколько рабочих (вальщик и его помощник, чокеровщик и др.), работает в разы эффективнее, обеспечивает высококачественную раскряжевку и отличается повышенной проходимостью. Специальная харвестерная голова дает возможность за пару секунд срезать дерево, очистить от сучков, и разрезать хлысты согласно их размеру.

Цель работы:

- ✓ Изучить основные технические характеристики валочно-сучкорезно-раскряжевочных машин различных производителей.
- ✓ Произвести сравнительный анализ экономических показателей валочно-сучкорезно-раскряжевочных машин различных производителей.
- ✓ Сделать выводы о необходимости приобретения харвестеров зарубежного или отечественного производства опираясь на технические характеристики и экономические показатели различных харвестеров.

При изучении выбранной темы использовались следующие методы исследования:

- Изучение основных технических характеристик по данным руководства по эксплуатации.
- Исследование основных систем машин методом сравнения.
- Расчет экономических показателей целесообразности.

На современном этапе ведения лесного хозяйства большая роль отводится рациональному использованию лесных ресурсов. Современный подход к заготовке древесины - это использование лесозаготовительных машин и механизмов.

Для того чтобы оценить целесообразность приобретения дорогостоящего харвестера, необходимо знать экономические показатели,

технические характеристики различных машин, которые идеально подойдут под определенные лесорастительные условия территории данного лесхоза и формы ведения лесного хозяйства в нем.

Перед руководителем лесохозяйственного предприятия всегда возникает вопрос об увеличении показателей заготовки и рациональному использованию древесины. Большинство руководителей принимают решение о закупке новой лесозаготовительной техники порой, не проводя оценку технических характеристик и экономических показателей. На сегодняшний день рынок перенасыщен различной лесозаготовительной техникой, как отечественных, так и зарубежных производителей. Для того чтобы сделать правильный выбор, необходимо учитывать все показатели приобретаемого оборудования. Среди многообразия предлагаемых машин для лесозаготовок были выбраны две марки лесозаготовительных машин, а именно харвестер Амкодор 2551 и Komatsu 911, которые имеют один класс, а также примерно одинаковые технические и экономические характеристики.

Харвестер Амкодор 2551. Данный харвестер считается на сегодняшний день одним из наиболее затребованных из числа большого количества моделей Амкодор. Его высокая проходимость, а кроме того мощность позволяющая осуществлять лесозаготовительные работы в местах, которые не всегда доступны для других машин этого типа.

Харвестер Комацу (Komatsu). Принадлежит к числу наилучшей техники, используемой в лесозаготовительной промышленности. Выделяется значительной экономичностью и небольшими энергозатратами. Осуществляет все без исключения функции, свойственные машине такого типа, валку, удаление сучьев, раскряжёвку, складирование сортимента. Может быть использован как на массовой вырубке леса, так и для прореживания единичных участков.

Опираясь на проведенное исследование можно сделать ряд заключений о целесообразности использования машин харвестеров на территории Республики Беларусь.

По техническим характеристикам, харвестер **Komatsu 911** уступает харвестеру Амкодор (вес машины, тип двигателя, гидравлическая система).

По сервисному обслуживанию, харвестер Амкодор 2551 наиболее ремонтно-пригодный, что выражается в наименьшем количестве проводимых работ, стоимостью работ и расходных материалов.

По экономическим показателям: рентабельной является машина Амкодор-2551.

Считаем целесообразным приобретение машин харвестеров Амкодор для лесохозяйственной деятельности на территории РБ.

УДК 687.1;391

Навуч. Я.А. Ганус; навуч. В.А. Камінская
Навук. кір. выкл. Н.М. Лісіца
(Філіял БДТУ «Полацкі дзяржаўны лясны каледж»)

ВЫКАРЫСТАННЕ ЭЛЕМЕНТАЎ ТРАДЫЦЫЙНАГА ДЭКОРА Ў СУЧАСНЫМ АДЗЕННІ

На працягу гістарычнага і культурнага развіцця склаліся найбольш дарэчныя формы адзення, яе крой, разнастайныя метады і спосабы дэкаратыўнага афармлення. Сярод іх вядучае месца займае вышыванка. Яна шырока выкарыстоўвалася ў народным касцюме і ў побыце. Вышыўка беларускага касцюма багатая і разнастайная. Ёю ўпрыгожвалі жаночыя і мужчынскія кашулі, верхнюю вопратку, галаўныя ўборы, аксесуары. Матывы арнаменту, кампазіцыі, колеру перадаваліся з пакалення ў пакаленне, становячыся традыцыйнымі.

Вышыванка – гэта беларуская нацыянальная вопратка, якая ў наш час становіцца больш папулярнай. Нашы продкі не насілі вышыванае адзенне кожны дзень, яны бераглі яе для асаблівых урачыстых выпадкаў. Адзенне, аздобленае ўзорамі, захоўвалася для вяселля і іншых святаў, была “выхадная”.

Сярэднекантынентальны досыць халодны клімат Беларусі застаўляў насіць цёплае і закрытае адзенне з мясцовай сыравіны – ільну, радзей каноплі, воўны натуральнай і паўнатуральнай, вытканай у хатніх умовах, на кроснах. У сувязі з тым, што праз Беларусь ішоў гандлёвы шлях, завозіць фабрычную тканіну пачалі значна раней. Таму вырабы з самаробнай тканіны амаль не захаваліся.

Колер рэчам надавалі прыродныя фарбавальнікі: настоі траў, кара, лісце, балотная жалезная руда. Для тканіны, вырабленай з ільну, для кашулі, фартуха, галаўных убораў фарбаванне не прымянялася. На белую тканіну часцей наносіўся чырвоны геаметрычны арнамент, выкананы спосабам натывання ці вышыўкі. Чырвоны колер – сімвал жыцця, акрамя гэтага ён выконваў функцыю засцерагальнага абярэга. Калі арнамантам расшывалі каўнер, пазухі, рукавы, ніз адзення, лічылі, што гэта засцерагала чалавека ад сурокаў, чорнага вока і надавала жыццёвыя сілы. З 19 стагоддзя ў вытворчасці ўзніклі анілінавыя фарбы і фабрычныя ніткі, і ў адзенні з’явіліся аранжавы, фіялетаваы, блакітны, малінавы, ярка – зялёны колеры.

Асаблівай фантазіяй у выкананні, колерах, вышыўцы вызначаўся жаночы касцюм. Кашулю насілі заўсёды, у любым узросце, людзі любога сацыяльнага статуса кожны дзень. Традыцыйны быў у кашулі тунікападобны крой, яе

ўпрыгожваліўстаўкі- полікі. Матэрыял для кашулі браўся ў залежнасці ад матэрыяльнага становішча, таму гэта мог быць як кужаль, так і бавоўна, шоўк. Бяднейшыя людзі маглі шыць будзеныя сарочкі з кужалю верхнюю частку (чэхлік), а ніжнюю (постаўку) – з грубай таннайтканіны. Паўсядзённая кашуля аздаблялася нешырокім стаячым або адкладным каўняром, а святочная – шыкоўным хвалепадобным каўняром, і нават здымным. Шыкоўная вышыўка або тканярнамент упрыгожвалі плечы, манжэтырукавоў, каўнерык вырабу. Адзенне ў беларускіх жанчын было дастаткова разнастайным, і нават спадніцы былі многіх відаў і назваў (андарак, палатнянік, саян, летнік), панёвы, фартухі. Андарак шылі з тонкага сукна дамашняга вырабу і часцей за ўсё ўпрыгожвалі тканым арнаментом, і на чырвоным, сінім, чорным фоне з’яўляліся вузкія рознакаляровыя палоскі вішнёвага, зялёнага, блакітнага, чорнага і іншых колераў. Узоры маглі ўтвараць прыгожыя гарызантальныя палоскі рознай шырынi. Для дэкарыравання выкарыстоўвалі цясёмкі і стужкі. Вылікай папулярнасцю карысталіся шыкоўныя гафрыраваныя андаркі. Іх акуратна складвалі ў гармонік, адпарвалі загорнутым ў ручнік гарачым хлебам, пасля – прэсавалі. Панёва– адзенне, якое насілі паверх кашулі як спадніцу, і складалася з трох сшытых полікаў. Наверх яе завязвалі фартух. Выкарыстоўвалі нераспашныя глухія панёвы, полкі якіх сшываліся напалову зверху. Сустракаліся панёвы - плахты, полкі якіх сшываліся ўверсе, і ромбам спускаліся на кашулю. Падол кашулі расшывалі або ткалі арнамент, бо панёва шылася карацей за кашулю.

Абавязковым і непасрэдным элементам нацыянальнай адзежы беларусаў быў рознакаляровы, звычайна чырвона – зялёна - белага арнаменту, пояс - тканы, плечены ці вязаны, з кутасамі, махрамі або пампонамі. З’явіцца на людзях без пояса лічылася непрыстойнасцю, нявыхаванасцю. Адсюль і пайшоў выраз “распаясаўся”. Найбольш пашыраныя паясы шырынёй ад 2-3 да 20-30 см, даўжынёй да 2 - 3 м і болей. Апаясваліся 2 - 3 разывакол таліі і завязваліся збоку, канцы звисалі амаль да каленяў. Зараз на Беларусі вырабляюць тканыя паясы ў г. Слуцку на базе музея слуцкіх паясоў.

Галаўныя ўборы – неад’емная частка нацыянальнага адзення беларусаў, у іх вызначалася сацыяльнае і сямейнае становішча. Вырабляліся яны з мануфактурных тканін (аксаміта, парчы, шоўка), футра собаля, куніцы, бобра, лісы, вавёркі, багата аздаблялі каштоўнымі камянямі, жэмчугам, пер’ем, вышыўкай і карункамі з залатых і срэбраных нітак. Жаночыя галаўныя ўборы, асабліва святочныя, былі багатыя на формы, спосабы нашэння, майстэрства,

вытанчанасцю выканання. Дзяўчаты зачэсвалі валасы на прамы прабор, запляталі ў 1-2 касы з каснікамі (рознакаляровымі стружкі, паскамі тканіны, ніткамі бісеру). Гэта маглі быць абручкі ці перавязкі - ручнікі, у выглядзе вянка, скіпдачкі, кубка, хусткі. Вельмі маляўнічымі былі вясельныя, купальскія ўборы, іх упрыгожвалі кветкамі, галінкамі дрэў, стужкамі, пер'ем. Замужнім жанчынам не дазвалялася паказвацца на людзях з непакрытай галавой, таму іх ўборы былі разнастайныя і складаныя па сваёй будове і спосабу нашэння, але больш сціплыя па ўпрыгожванні. Увогуле галаўныя ўборы надавалі жанчыне велічнасць і статнасць, як бы скульптурную ізысканасць і цэласнасць. Жаночыя галаўныя ўборы падзяляюцца на ручніковыя, па назвах яны былі разнастайныя: намітка, сярпанка, хустка; рагацістыя: галовачка, рожкі, сарочка; каптуровыя: каптур, чапец, наколка. Не было летніх і зімовых галаўных убораў, проста ў холад на іх вязалася хустка.

Абутак – неабходная неад'емная частка адзення, якая дапаўняе яго ў адзінае цэлае. Аднымі з самых пашыраных з'яўляліся скураныя або плецяныя лапці. Амаль у кожнага селяніна было некалькі пар лапцей, яны пляліся з ліпавай або лазовай кары, бяросты, паньковых або льняных аборака. Было 2 тыпы лапцей у залежнасці ад тэхнікі вырабаў: прамога (пашыраны паўсюдна) і касога (на ўсходзе Беларусі) пляцення. Самымі пашыранымі былі лапці з глыбокім закрытым носам - кавярзні. На Віцебшчыне, часткова Магілёўшчыне плялі шчарбакі (бяспятнікі) - неглыбокія, можна сказаць з адной падэшвы. Можна было сустрэць лапці, у якіх спераду былі дзве шырокія лыкавыя пятлі і запятак, а з кожнага боку – пара вузкіх петляў для прадзявання аборака. Называлі іх шчарбакі, насілі іх ўлетку на палявых работах. На Палессі, поўдні Гродзеншчыны і Міншчыне насілі “зрачыя лапці”, у якіх насок пасярэдзіне зверху быў адкрыты – “вока”.

У канцы 19 – пачатку 20 ст. атрымалі распаўсюджанне валёнкі, якія насіліся ўзімку. Да падэшвы валёнак прышываліся тоўсты лямец або кусочкі скуры. Пазней на валёнкі прыдумалі надзяваць гумаваыя галошы. Па абутку можна было вызначыць мясцовыя традыцыі, пол, узрост, статус гаспадара. Зараз таксама вельмі модным зімнім абуткам з'яўляюцца валёнкі.

Важнае сімвалічнае значэнне мае колер у любым нацыянальным касцюме. У касцюме беларуса выкарыстоўваліся тры колеры: чырвоны, белы, чорны. Усе яны носяць сакральны сэнс. Белы колер сімвалізуе чысціню і свет, з'яўляецца сімвалам карысці, сілы і здароўя, чысціні, жыцця, урадлівасці і пладавітасці. Чырвоны колер атаясамліваецца як з добром, так і са злом, нясе сілу і жыватворную

кроў для чалавека, жывёлы, птушкі. Чорны колер –негатыўны, сімвал зла, няшчасця, хваробы, смерці, ночы, цемры.

Канешне, кожны ўзор на вышытым ці вытканым адзенні штосьці сімвалізуе. Ва ўзорах тканых і вышываных вырабаў зашыфраваны аповяд пра жыццё народа, прыроду, людзей. Беларускі арнамент можна падзяліць на тры асноўныя катэгорыі: *геаметрычны* – арнамент, які складаецца з прамых ліній, геаметрычных фігур - прамавугольнікаў, квадратаў, ромбаў; *раслінны* - створаны на аснове малюнкаў раслін, лісця, кветак ці пладоў; *жывёльны* - які ўключае ў сябе выявы жывёл і птушак.

Белая вышытая сарочка, так званая вышыванка, - неадемная складовая частка жаночага і мужчынскага беларускага нацыянальнага адзення. Вышытая сарочка была характэрнай прыкметай кожнай мясцовасці. У многіх культурах ў розныя стагоддзі адзенне з вышыўкай было сведчаннем дабрабыту.

СОДЕРЖАНИЕ

Секция ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

1	<i>Карнеева Д. Д.</i> особенности разработки интерактивного учебника по дисциплине «Обработка изобразительной информации. Раздел “Коррекция изображений”».....	5
2	<i>Карпач А. М.</i> Генерирование фиктивных данных при помощи FAKER.JS.....	10
3	<i>Карпач О. М.</i> Особенности дизайна и разработки игрового портала «Тайны забытого острова».....	13
4	<i>Ахремчик Е. А.</i> Особенности дизайна и разработки веб-сайта «YUOLLA».....	16
5	<i>Харланович А. В.</i> Особенности дизайна и разработки веб-сайта «Аман Трейдинг».....	18
6	<i>Михайлова А. М.</i> Разработка веб-сайта «УП "Беллесэкспорт"».....	22
7	<i>Карпач А. М.</i> Разработка функциональной структуры социальной сети «Social exchange».....	25
8	<i>Юхновец Л. С., Леонова А.С.</i> Анализ электронных изданий СМИ на «отзывчивость».....	29
9	<i>Леонова А. С., Юхновец Л. С.</i> Использование стерео (анаглифных) изображений при изучении технических дисциплин.....	36
10	<i>Смирнов Е. Р.</i> Разработка на React/Redux с использованием Ramda.....	39
11	<i>Пустовалов А. К.</i> Разработка сайта для анализа хозяйственной деятельности предприятия.....	43
12	<i>Буйко А. В.</i> Использование сценариев JavaScript в HTML-документах	46
13	<i>Кажушко К. Н.</i> Разработка сайтов с использованием фрейм-мов.....	48
14	<i>Раковец Р. В.</i> Использование таблиц CSS в HTML-документах.....	49
15	<i>Сорока В. В., Синькевич А. А.</i> Электронный каталог резюме выпускников.....	51
16	<i>Марченко А. А.</i> Навигация в MATERIAL DESIGN 2.0.....	53
17	<i>Тозик К. В.</i> Защита информации в открытых облачных хранилищах.....	57

18	Назаренко В. В. Темные паттерны: использование UX дизайна для обмана пользователей.....	60
19	Булохова В. Ю., Сорока В. В. Применение семантических тегов HTML5 <SECTION> и <ARTICLE>.....	63
20	Крушинский Я. С. Создание интернет-магазина в социальной сети Вконтакте.....	65
21	Ратайко Ю. Н. Игровой портал «The Secret of the Past».....	67
22	Ляшкевич Д. И. Fluttel – фреймворк для кросс-платформенной разработки.....	71
23	Радченко М. Ю. О проблеме темных паттернов в веб-дизайне.....	73
24	Радченко М. Ю. Использование Actions программы Adobe Photoshop для создания эффекта акварельного рисунка....	76
25	Белькевич Р. И. Метод осаждения авторской информации в звуковые файлы.....	80
26	Олеферович А. В. Влияние дефектов на процесс распознавания объектов на изображении методом контурного анализа.....	85
27	Буйко Е. А. Рендеринг графических изображений.....	88
28	Карленок Ю. А. Дискретное косинусное преобразование..	92
29	<i>Савельева</i> М. Г. Игровой портал «Миньон в лабиринте»...	94
30	КантаровичВ. С. Интернет-издание о зимних видах спорта «Misha».....	97
31	<i>Жаденова</i> Е. И.Технология WebGL в дизайне сайтов.....	100
32	<i>Попеня</i> Н. В.Повышение конверсии сайта с помощью увеличения кликабельности.....	104
33	<i>Попеня</i> Н. В. О технологии создания авторского шрифта	106
34	<i>Хамутовская</i> А. В. On-line игра «Cucaracha».....	108
35	<i>Ромашко</i> Н. Н.Разработка и создание сайта кафе «Вафли есть».....	112
36	<i>Буртыль</i> Д. В., <i>Кондрашова</i> А. А. Особенности и структура интерфейса в интернет-телевидении.....	114
37	<i>Кондрашова</i> А. А. Разработка системы дистанционного обучения по программе MS EXCEL.....	117
38	<i>Леонова</i> А. С., <i>Юхновец</i> Л. С. Основные элементы дизайна, участвующие в создании композиции.....	119
39	<i>Гришаева</i> Е. В.Использование принципов юзабилити в современных веб-сайтах.....	122
40	<i>Гарбуз</i> Е. В.Как с помощью дизайна текста, цвета и изображения повысить юзабилити интерфейса.....	127

41	<i>Левченко Н. В.</i> Использование дизайн-системы для ускорения разработки и поддержки.....	131
42	<i>Жаденова Е. И.</i> Motion-дизайн в интерфейсах мобильных приложений для улучшения юзабилити.....	134
43	<i>Завадская В. О.</i> Как улучшить веб-доступность с помощью выбора правильной палитры.....	138
44	<i>Целуйко И. С.</i> Веб-сайт французской выпечки «deLepel»...	141
45	<i>Зенькович И. А.</i> Игровой портал «Labyrinth Escape».....	144
46	<i>Юхновец Л. С.</i> Использование внешних программных средств для стилизации изображений.....	147
47	<i>Леонова А. С.</i> Разработка 3D-модели по фотографии.....	150
48	<i>Шаплыко В. С.</i> Влияние дизайна на продажи интернет-магазина.....	153
49	<i>Коротыш Т. А.</i> Об использовании svg-масок.....	156
50	<i>Панасик Г. А.</i> Веб-сайт ветеринарной клиники «ПЁССИ-КОТ».....	160
51	<i>Гусакова А. Н., Жуков С. В.</i> Интернет-платформа консалтинговых услуг «CONSU».....	165
52	<i>Красовская Н. Н.</i> Интернет-приложение «Консалтинг фриланс».....	172
53	<i>Байденок А. С.</i> Разработка игрового веб-приложения «Ремонт космической станции».....	175
54	<i>Шаплыко В. С.</i> Тренды дизайна 2019.....	178
55	<i>Бриль В. С.</i> Веб-площадка для сбора помощи тяжелобольным детям «Ways to help».....	182
56	<i>Кузьмина С. А.</i> Веб-площадка «Pickofpic».....	186
57	<i>Процукович К. М.</i> Сравнительный анализ алгоритмических свойств криптовалютных технологий....	189
58	<i>Клишунова К. В., Дубовик Е. И.</i> Стеганографическое преобразование информации в компьютерных сетях.....	193
59	<i>Тарасевич Д. А.</i> Особенности дифференциального и линейного криптоанализа.....	198
60	<i>Глушко А. В.</i> Система учета рабочего времени сотрудников IT-компании.....	202
61	<i>Дмитрачёнок В. В.</i> Программный модуль для озвучивания электронных почтовых сообщений.....	205
62	<i>Костюк А. А.</i> Веб-ресурс для распространения аудиокниг..	208
63	<i>Рыжковский А. А.</i> Веб-приложение для поддержки принятия решения при выборе земельного участка.....	211
64	<i>Цедрик Е. А., Грудинский П. В.</i> Человеческое мышление или машинный алгоритм.....	215

65	<i>Колодко В. А.</i> Блокнот 2.0.....	217
66	<i>Бурая Е. А.</i> Веб-приложение для управления персональными данными кандидатов и сотрудников компании.....	220
67	<i>Дуденкова Е. И.</i> Электронная очередь на консультацию к преподавателю вуза.....	222
68	<i>Жук Е. В.</i> Обучение программированию посредством авторского программного продукта.....	224
69	<i>Бондарчик Е. Н.</i> Хранилище данных как инструмент автоматизации распределения нагрузки преподавателей....	228
70	<i>Сахонь Е. С.</i> Интерфейс программного комплекса для дистанционного выполнения виртуальных лабораторных работ.....	230
71	<i>Лосик Д. С.</i> Модель задачи оптимизации маршрутов перевозки нефтепродукта.....	233
72	<i>Баранов В. А.</i> Шифрование информации методом создания динамического ключа.....	236
73	<i>Оганесян А. Г.</i> Анализ данных с использованием библиотеки Pandas на языке Python.....	238
74	<i>Ермаков К. А.</i> Особенности практической реализации квантовой криптографии на основе протокола Беннета.....	240
75	<i>Карпович М. Н., Карпович Е. В.</i> Шифрование информации в беспроводных сетях. Уязвимости алгоритма «Рукопожатия».....	243
76	<i>Суценя А. А.</i> Стеганографический метод внедрения текстовой информации в контейнер формата EPUB.....	247
77	<i>Тихонович М. А.</i> Криптостойкость гомоморфных шифров....	251
78	<i>Савенко А. Г., Черняк А. Д.</i> Реализация «легковесных» криптографических алгоритмов для устройств с небольшой вычислительной мощностью.....	256
79	<i>Чопик А. А.</i> Реализация сервера для предоставления удобного интерфейса к данным блокчейна ECHO и возможности подписаться на их обновление.....	260
80	<i>Шпаковский Е. А.</i> Сетевая безопасность: сниффинг и спуфинг.....	263
81	<i>Юрашевич Д. Э.</i> Коллизии и необратимость хеш-преобразований в алгоритме SHA-256.....	267
82	<i>Ющик Д. С.</i> Разработка программного средства для повышения финансовой грамотности школьников.....	270
83	<i>Куницкий Н. Д.</i> Распределенная система менеджмента дополнительных занятий.....	273

84	<i>Ивлиева А. А.</i> Распределенное приложение для управления покупкой и доставкой подарков.....	276
85	<i>Ахраменок К. А.</i> Применение скоринга в работе с Bigdata.....	278
86	<i>Тарасенко А. В.</i> Сервис для подбора новостных статей на основе предпочтений пользователя.....	280
87	<i>Зайцев А. Н.</i> Статический анализ эффективности использования контейнеров из библиотеки STL в программах на языке программирования C++.....	281
88	<i>Демещик А. С.</i> Подходы к проведению сентимент-анализа....	285
89	<i>Мартынюк А. В.</i> Помехоустойчивые кодеки на основе каскадных схем кодирования/декодирования для систем передачи информации с одиночными и модульными ошибками.....	289
90	<i>Капелько И. В.</i> Анализ работоспособности алгоритма трехуровневой токенизации для автоматического реферирования текста.....	292
91	<i>Панченко О. Л.</i> Методы оперативной классификации текстовых документов.....	295
92	<i>Ничипорчик Д. С.</i> Осаждение информации в изображениях.....	298
93	<i>Ральцевич Э. В.</i> Разработка веб-ресурса «RESOURCE PLAINING SYSTEM».....	301
94	<i>Ермакович А. А.</i> Разработка веб-ресурса «отображение аналитических данных фармацевтических компаний».....	303
95	<i>Плотников Д. А.</i> Обзор архитектуры Linux x64.....	306
96	<i>Богдан Ю. С.</i> Бэк-офис система для управления сетью магазинов.....	308
97	<i>Карпович П. И.</i> Веб-приложение для автоматизации процесса создания однотипных рекламных роликов.....	312
98	<i>Литвинович И. А.</i> Алгоритмы и подходы к определению принадлежности профиля в социальной сети.....	315
99	<i>Ghassan Jaber</i> Scalable content routing system for icn networks.....	317
100	<i>Бернацкий П. В.</i> Разработка веб-приложения «Форум для разработчиков программного обеспечения» с использованием JAVA EE.....	320
101	<i>Мелентьев Н. И.</i> Веб-ориентированная система поддержки работы риэлтерской компании.....	323
102	<i>Цыбулько К. Д.</i> Анализ алгоритмов сжатия информации.....	326
104	<i>Гордеев И. В.</i> Социальная платформа для продуктов.....	328

105	<i>Курлович Е. Ю.</i> Информационная система для совершенствования образовательного процесса в университетах.....	330
106	<i>Синица А. Д.</i> Автоматизированная система оценки недвижимости.....	332
107	<i>Дубовик М. В.</i> Веб-приложение для оценки эмоциональной тональности отзывов на фильмы.....	335
108	<i>Борсук Н. И.</i> Сервис подбора недвижимости для долгосрочной аренды.....	338
109	<i>Сазонова Е. С.</i> Веб-платформа для музыкантов.....	341
110	<i>Коляго Е. В.</i> Веб-приложение «Афиша мероприятий».....	345
111	<i>Татарович Д. А.</i> Разработка веб-приложения «Такси агрегатор».....	348
112	<i>Самаль А. Д.</i> Сравнительный анализ алгоритмов классификации изображений и их реализаций.....	351
113	<i>Столяров Д. С., Трифонов М. И.</i> Разработка игры «Змейка» на языке программирования Python 3.7.2.....	353
114	<i>Радиванович Д. А.</i> Использование методов машинного обучения для классификации электронной корреспонденции.....	356
115	<i>Мушук А. Н.</i> Разработка приложения «Онлайн-библиотека».....	359
116	<i>Яромич С. С.</i> Web-приложение для организации делопроизводства деканата.....	362
117	<i>Шкода Т. М.</i> Система управления проектами на базе платформы «Sales force».....	365
118	<i>Кремко И. Е.</i> Онлайн платформа для организации учебного процесса в учреждении среднего образования.....	367
119	<i>Разумовский И. И.</i> Информационный ресурс «Postim.by».....	370
120	<i>Русакович А. Д.</i> Применение фреймворка Prism при создании настольных приложений, основанных на технологии WPF.....	373
121	<i>Кунец Д. С.</i> Прогрессивные веб-приложения.....	377
123	<i>Помазай Е. С.</i> Прогнозирование результатов теннисных матчей с применением методов машинного обучения.....	380
124	<i>Луцицкий А. Н.</i> Программный модуль для эффективного управления элитными зданиями.....	383
125	<i>Шенец П. С.</i> Регрессионный анализ при решении задачи прогнозирования.....	386
126	<i>Кебец И. В., Рекунов З. В.</i> Многообразие фракталов: их применение и построение.....	390

127	<i>Черенков А. Т.</i> Метод Штрассена.....	394
128	<i>Алексеев М. Е.</i> Шифрование методом гаммирования.....	398
129	<i>Злобин Р. Ю.</i> Некоторые применения теории графов.....	401
130	<i>Радчиков А. Д.</i> Сравнение скорости сходимости частичных сумм ряда Фурье.....	405
131	<i>Марчук К. С.</i> Использование теории групп точек на эллипти- ческой кривой для создания электронной подписи.....	408
132	<i>Жерноклёва А. В.</i> Некоторые представления натуральных чисел.....	413
133	<i>Фирьян М. Д.</i> Использование математических моделей для анализа работы одномашинных лесопромышленных систем без запаса.....	415
134	<i>Цмак М. М.</i> Математическая модель работы одномашинных лесопромышленных систем с запасом.....	418
135	<i>Минченя А. А., Климович П. Г.</i> Депротонирование молекул свободных оснований коррола в растворах.....	423
136	<i>Пилипчик М. С.</i> Элементный анализ вещества с применением метода лазерной эмиссионной спектроскопии.....	426
137	<i>Синявский В. В.</i> Спектрально-люминесцентные свойства асимметрично замещенных молекул тетрафенилпорфирина.....	430
138	<i>Байдук А. В.</i> Оценка безопасности водоемов технического назначения (очистных).....	433
139	<i>Невар В. П.</i> Осаждение твёрдых частиц в отстойниках технического назначения.....	435
140	<i>Мигай В. В., Тимановский Д. И.</i> Число e в математических моделях.....	439
141	<i>Шелегова Д. С.</i> Математическая модель скорости химической реакции	443

Секция

ПЕРВЫЙ ШАГ В НАУКУ

1	<i>Телешова А. А.</i> Изучение влияния на изменение свойств при введении в гипсовую массу гранитного отсева.....	447
2	<i>Крук П. В.</i> «Жыццё» ўнутры верша.....	450
3	<i>Емяльянава А. А.</i> Сродкі прыцягнення ўвагі чытачоў ў загалюўках газетных публікацый.....	451
4	<i>Силич К. М.</i> Иван-чай – белорусский чай.....	453

5	<i>Воронец И.С., Крутолевич С.А.</i> Краудфандинг – будущее ваших идей.....	454
6	<i>Овчинникова П.А.</i> Государство для близнецов: прогресс или социальное неравенство?.....	456
7	<i>Пучковский А.В.</i> Современный подход для работы с базами данных. возможности entityframework.....	459
8	<i>Ширяева А.А., Ратников М.Ю.</i> Степень влияния пороков на свойства древесины. Снижение вероятности появления пороков путем лесохозяйственных и лесозащитных мероприятий.....	460
9	<i>Шелопухо Ю.А., Тарасенков В.П.</i> Возможность использования гибкой связи в механизмах подачи металлорежущих станков.....	464
10	<i>Рыжиков И.В.</i> Лабораторная установка для плк Mitsubishi electric alpha-2.....	467
11	<i>Кокотова А.И., Боровик В.В.</i> Выращивание посадочного материала ели голубой.....	471
12	<i>Тесленко К.Э., Минкова А.А.</i> Влияние температуры на прорастание семян сосны обыкновенной.....	474
13	<i>Нестеренко Н.С.</i> Учебно-исследовательская работа «júglansmandshúrica – новое в медицине».....	478
14	<i>Морозов С.В., Хитрик О.Г.</i> Учебно-исследовательская работа «phlebiopsis gigantea – супер современное средство в борьбе с корневой губкой».....	480
15	<i>Зародий К.М.</i> Учебно-исследовательская работа «изучение основных технических характеристик, сравнительный анализ экономических показателей валочно-сучкорезно-раскряжевых машин (харвестер) различных производителей».....	482
16	<i>Ганус Я.А., Камінская В.А.</i> Выкарыстанне элементаў традыцыйнага дэкора ў сучасным адзенні.....	484

Научное издание

Ответственный за выпуск *Е.О. Черник*

**70-я НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ
УЧАЩИХСЯ, СТУДЕНТОВ
И МАГИСТРАНТОВ**

СБОРНИК НАУЧНЫХ РАБОТ

В 4-х частях

Часть 4

В авторской редакции

Компьютерная верстка *Е.О. Черник, А.В. Сильванович*

Издатель:

УО «Белорусский государственный технологический университет».

ЛИ № 02330/0549423 от 08.04.2009.

ЛП № 02330/0150477 от 16.01.2009.

Свердлова, 13а, 220006, г. Минск