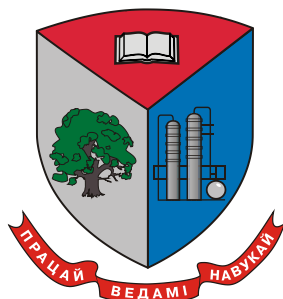


МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ
УНИВЕРСИТЕТ»



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

17–22 апреля 2017 г.

СБОРНИК НАУЧНЫХ РАБОТ

В 4-х частях

Часть 4

Минск 2017

УДК 005.745:378.6](476)(06)

ББК 66.75

Ш52

68-я научно-техническая конференция учащихся, студентов и магистрантов: сб. науч. работ : в 4-х ч. – Минск : БГТУ, 2017. – Ч. 4. – 356 с.

Сборник научных работ студентов и магистрантов университета составлен по итогам 68-й студенческой научно-технической конференции Белорусского государственного технологического университета, проведенной с 17 по 22 апреля 2017 г. На конференции было заслушано 1223 доклада, 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

Д.В. Шиман

методист сектора методического обеспечения учебного процесса

О.И. Ставровская

Редакционная коллегия:

зав.кафедрой ИСиТ, доц., канд.техн. наук

В.В. Смелов

проф. каф. ИСиТ, д-р техн. наук

П.П. Урбанович

зав. кафедрой ИиВД, доц., канд. техн.наук

Д.М. Романенко

зав. кафедрой физики, д-р физ.-мат. наук

Н.Н. Крук

© УО «Белорусский государственный
технологический университет», 2017

ОРГАНИЗАЦИОННЫЙ КОМИТЕТ

ДОРМЕШКИН О.Б. – проректор по научной работе (председатель);

САКОВИЧ А.А. – проректор по учебной работе, зам. председателя;

НАСОНОВ М.Д. – начальник учебно-методического отдела;

ЧЕРНИК Е.О. – зав. сектором ИВОНД;

ЛЕВИТСКАЯ А.А. – инж. 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 и технологии лесозаготовок;

РЫМОВСКАЯ М.В. – ответственная за НИРС факультета ТОВ, канд. техн. наук, доц. кафедры биотехнологии и биоэкологии;

ЗАЛЫГИНА О.С. – ответственная за НИРС факультета ХТиТ, канд. техн. наук, доцент кафедры промышленной экологии;

ШИШЛО С.В. – ответственный за НИРС факультета ИЭ, канд. эконом. наук, доцент кафедры экономической теории и маркетинга;

МЕДЯК Д.М. – ответственная за НИРС факультета ПИМ, канд. техн. наук, доцент кафедры полиграфических производств;

ЯРОЦКАЯ Л.Д. – ответственный за НИРС факультета ИТ, канд. физ.-мат. наук, доцент.

Секция
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

УДК 004.41

Студ. Д. С. Кунец, студ. И. В. Гордеев
Науч. рук. к.т.н., асс. С. А. Борисевич
(кафедра информатики и веб-дизайна, БГТУ)

ПРОГРАММИРОВАНИЕ ПРИЛОЖЕНИЙ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В СРЕДЕ UNITY

Дополненная реальность (*AR – Augmented reality*) – это дополнение физического мира цифровыми данными в режиме реального времени, при помощи компьютерных устройств.

По сути, и, несмотря на название, эта технология может, как дополнять окружающий мир объектами мира виртуального, так и устранять из него объекты — возможности AR ограничиваются лишь возможностями соответствующих устройств и программ. Тем не менее, сегодня все или почти все решения на основе дополненной реальности выполняют исключительно приведённую в названии функцию.

В технологиях одновременного взаимодействия пользователя с реальным и виртуальным мирами задействованы лишь слух и зрение, и, при условии встраивания в будущие системы AR сообщающих пользователю тактильные ощущения устройств, область их применения может быть значительно расширена, выйдя за рамки информационных и развлекательных решений. Потенциал этих технологий огромен.

Для отображения дополненной реальности используются различные типы устройств. Эти устройства в основном делают одну вещь: создают изображения, ориентируясь на те вещи, которые оно наблюдает перед собой. Следовательно, от того, куда направлен взгляд пользователя, зависит работа оборудования. Если сравнить с устройствами виртуальной реальности, то человек помещается в уже заготовленный виртуальный мир, в котором можно выполнять определенные действия.

Существуют различные виды классификации дополненной реальности. Первая разновидность дополненной реальности – это сенсорная дополненная реальность. Она основана на использовании GPS, гироскопа и акселерометра. Примерами этой разновидности Дополненной Реальности могут быть следующие технологии: *Layar*, *Wikitude*, *Sekai Cam*. Другой разновидностью является дополненная реальность, в основе которой лежит распознавание объектов и *Open GL (Open Graphics Library)*. Примера таких технологий являются *Vuforia*, *Metaio*, *Total Imersion ARToolKit*. Также можно выделить

смешанный тип Дополненной Реальности, которая является сочетанием компьютерного распознавания объектов и использования датчиков.

Принцип работы технологий технологий разнообразен. Рассмотрим в качестве примера технологию *Vuforia*. Чтобы добавить с помощью этой технологии в реальный мир, нужно в этом мире «зацепиться» за какой-либо объект. Именно для этой цели и служат мишени. Мишень – это некий реальный объект, зная который наше приложение может расставить виртуальные объекты в нужных местах и соответствующих пропорциях. *Vuforia* предоставляет богатый выбор мишеней, некоторые из них предоставлены на рис.1.



Рисунок 1 – Мишени Vuforia

Теперь немного о мишенях:

Image targets – базовый вид мишеней, представляющий собой обычную картинку, например, обложку журнала, фотографию или афишу нового фильма. Картинка выполняет роль своего рода двумерного штрих-кода, только без черно-белых регионов. По ней мы можем определить, какая именно картинка попала в объектив камеры, а также её расположение в пространстве и масштаб.

Simple 3D targets (Cube and Cuboid) – это мишени в виде прямоугольных параллелепипедов (включая куб). Например, такой мишенью может служить упаковка из-под сухих завтраков, спичечный коробок или только что купленная настольная игра. Как и любая коробочка, такая мишень состоит из шести плоскостей, и чтобы создать её, нам понадобится шесть картинок для каждой из них.

Cylinder targets – этот вид мишеней, несмотря на название, представляет собой усеченный конус с возможностью задавать диаметры оснований. Конечно если выбрать одинаковые диаметры, то получатся как раз цилиндр, но все же это частный случай. Для того, чтобы создать такую мишень нам понадобится не только указать диаметры оснований и высоту, но также добавить три картинки – по одной для каждого из двух оснований, и еще одну для боковой поверхности.

Frame markers – это мишень в виде специально подготовленной рамки, которая уже куда более походит на штрих-код. В такую рамку можно поместить любую картинку. Данный вид мишеней отлично подходит в случае, если картинка не была достаточно детализирована и создать из нее толковую *image target* не получилось.

Text (word targets). В библиотеку Vuforia встроено еще и распознавание текста, поэтому любое слово или их сочетание может являться мишенью. На данный момент поддерживается только латиница, но и это уже очень интересно. Только представьте насколько интересным может стать изучение иностранных слов.

Пример реализации *Image target* в среде Unity приведен на рис.2.

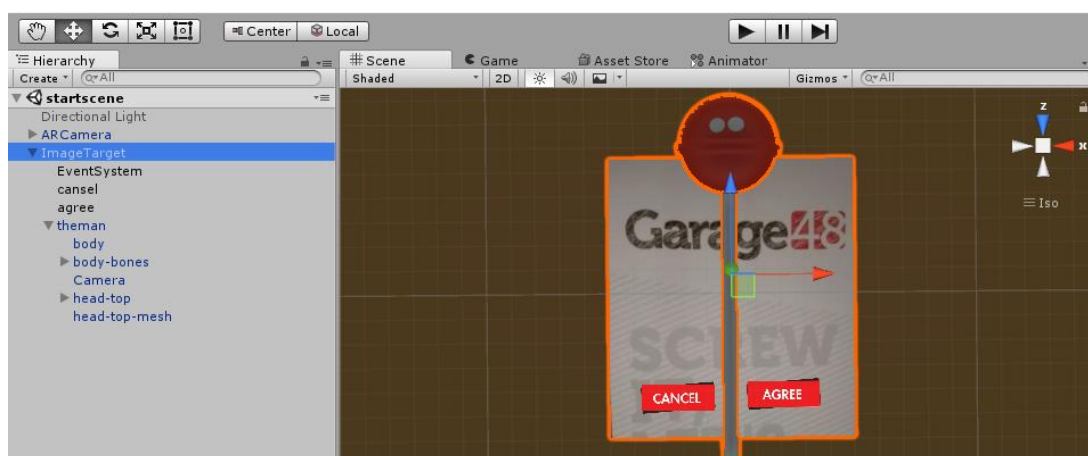


Рисунок 2 – Пример *Image target* в среде Unity

Также, мишени могут быть созданы непосредственно из самого приложения используя картинку полученную с камеры.

Стоит также отметить, что QCAR (библиотека, являющая ядром платформы) не требует попадания всей мишени в кадр. Для того чтобы её распознать и использовать достаточно будет лишь части. Это очень важно и позволяет рассматривать “магию” под самыми разными углами.

Устройства, для отображения дополненной реальности, могут быть использованы очки такие как *Google Glass*, *Vuzix Wrap 1200AR*, *Vuzix Smart Glasses M100*, а также смартфоны и планшетные компьютеры на операционных системах *iOS*, *Android*; игровые приставки *Sony PlayStation Vita*, *Nintendo 3DS* и *Nintendo XL*. Тип программного обеспечения *Google Glass* неизвестен, а *Vuzix Smart Glasses M100* работает на модифицированной версии *Android*

УДК 324.679

Студ. Е.А. Чевжик

Науч. рук. доц. А.А. Дятко

(кафедра информатики и веб-дизайна, БГТУ)

ИСПОЛЬЗОВАНИЕ GOOGLE FIREBASE В ANDROID- ПРИЛОЖЕНИИ

Большинство мобильных приложения используют сервер для хранения данных и приложение, которое интегрируется с этой базой данных. Для наилучшего результата требуются как специалисты *front-end* для разработки клиентской части (по сути, самого приложения, с которым общается пользователь) так и *backend*-специалисты для разработки серверной части (хранилище данных общее для всех пользователей приложения).

Роль последних стала несколько меньше после того, как появились облачные сервисы, предоставляющие возможность хранить гибкую систему данных у себя на сервере. В результате, процесс создания приложения значительно упростился, и многие стартап проекты заменили работу backend-специалистов на облачные сервисы.[1]

Бэкенд как сервис (BaaS) — модель, которая предоставляет разработчикам мобильных и веб-приложений возможность хранить и обрабатывать данные в облаке, не поднимая собственный сервер, а также предоставляет такие функции, как управление пользователями, push-уведомления и интеграцию с сервисами социальных сетей. Эти услуги предоставляются посредством использования комплектов для разработки специального программного обеспечения (SDK) и интерфейсов прикладного программирования (API).

Ранее флагманом движения BaaS считалась компания Parse. Она была основана в 2011 году и предоставляла решение для удаленной работы с данными (обработка и хранение), которое упрощало написание серверной бизнес-логики на JS.

Однако в 2016 году Facebook купили стартап и прекратили обслуживание клиентов через год. В связи с неожиданным решением Facebook закрыть Parse, многие разработчики задались вопросом, что использовать вместо него. Сегодня практически невозможно представить полностью автономное приложение, которое было бы полезно всем.

Среди наилучших и наиболее популярных BaaS можно выделить Firebase от компании Google.

Firestore — поставщик облачных сервисов и приложений. Основным направлением Firestore является облачная NoSQL база данных для real-time приложений, которая предоставляет API, позволяющее разработчикам хранить и синхронизировать данные между несколькими клиентами. База данных позволяет работать с данными, которые синхронизируются в реальном времени и доступны при отсутствии интернета[1]. Кроме базы данных Firestore предлагает различные возможности для программистов:

- Firestore Analytics – это инструмент для анализа мобильных приложений.
- Firestore Cloud Messaging (FCM)— сервис для доставки push-уведомлений из облака на устройства. Он полностью бесплатен, оптимизирован для экономии заряда батареи и работает на iOS, Android и в браузерах.
- Firestore Storage позволяет организовывать загрузку и хранение файлов (изображения, видео, аудио и др.).
- Firestore Remote Config позволяет подстраивать и обновлять элементы приложения на лету без необходимости обновлять пакет приложения и ждать, пока он станет доступен в магазине приложений, а затем обновится у пользователей. Вы можете включать и выключать определённые элементы приложений, распространять обновления на конкретные Аудитории пользователей.
- Firestore Crash Reporting собирает и отправляет самую важную информацию, которая может помочь в поиске проблем iOS/Android-приложений после релиза.
- Firestore Test Lab - облачное тестирование приложений на реальных устройствах, которые расположены в дата-центрах.
- Firestore Notifications — новый интерфейс, построенный на базе API Firestore Cloud Messaging, дополненный возможностями Firestore Analytics и Аудиториями. Он позволяет таргетированно разослать уведомления пользователям вашего приложения, не прикасаясь к редактору исходного кода.
- Firestore Dynamic Links улучшает работу со ссылками.
- Firestore App Indexing (бывший Google App Indexing) позволяет находить пользователям Google результаты непосредственно из вашего приложения.

Также в Firebase интегрирована рекламная платформа AdWords. Firebase отслеживает прогресс вашей кампании в AdWords и автоматически заполняет соответствующие разделы в Firebase Analytics. Рекламная площадка Google AdMob также является частью Firebase.

По существу, Firebase является безусловно потрясающим в исполнении, реализации и эксплуатации решением. Firebase служит базой данных, которая изменяется в реальном времени и хранит данные в JSON. Любые изменения в базе данных тут же синхронизируются между всеми клиентами, или девайсами, которые используют одну и ту же базу данных. Другими словами, обновление в Firebase происходят мгновенно.

Вместе с хранилищем, Firebase также предоставляет пользовательскую аутентификацию, и поэтому все данные передаются через защищенное соединение SSL. Мы можем выбрать любую комбинацию email и пароля для аутентификации, будь то Facebook, Twitter, GitHub, Google, или что-то другое.

В добавку к Android SDK, у Firebase есть iOS SDK и JavaScript. Все платформы могут использовать одну базу данных.

Чтобы начать работу с Firebase, необходимо создать приложение в консоли.

Далее на вкладке Auth выбираем способ авторизации, например e-mail и пароль.

Начиная с версии 2.2 в Android Studio встроен плагин для работы с Firebase. С его помощью можно легко настраивать сервисы, а также автоматически подключаться к приложению и добавлять зависимости [3].

После создания приложения в консоли Firebase, необходимо подключить Firebase к проекту Android Studio. Для этого необходимо открыть плагин Firebase, выбрать необходимый сервис и нажать кнопку Connect. Далее в появившемся окне выбираем приложение, созданное ранее в консоли Firebase и нажимаем Connect to Firebase.

Таким образом Firebase — платформа, которая предоставляет разработчикам широкие возможности для построения приложений, привлечения пользователей, а также его монетизации и аналитики.

ЛИТЕРАТУРА

1. Firebase: аутентификация пользователей с помощью email и пароля в android приложении [Электронный ресурс] / Wikipedia. – 2017. / Режим доступа: <http://intellection.kz/blog/%D1%87%D0%B5%D0%BC-%D0%BC%D1%8B-%D0%B7%D0%B0%D0%BC%D0%B5%D0%BD%D0%B8%D0%BB%D0%B8-parse/>
2. Firebase [Электронный ресурс] / Wikipedia. – 2017. / Режим доступа: <https://ru.wikipedia.org/wiki/Firebase>. – Дата доступа: 25.03.2017
3. Users in Firebase Projects [Электронный ресурс] / Wikipedia. – 2017. / Режим доступа: <https://firebase.google.com/docs/auth/users>. – Дата доступа: 25.03.2017

УДК 324.679

Студ. В.В. Юранов

Науч. рук. доц. А.А. Дятко

(кафедра информатики и веб-дизайна, БГТУ)

РАЗВЕРТЫВАНИЕ САЙТА НА HEROKU

Heroku — облачная PaaS-платформа, поддерживающая ряд языков программирования. Компанией Heroku владеет Salesforce.com. Heroku, одна из первых облачных платформ, появилась в июне 2007 года и изначально поддерживала только язык программирования Ruby, но на данный момент список поддерживаемых языков также включает в себя Java, Node.js, Scala, Clojure, Python, Go и PHP. На серверах Heroku используются операционные системы Debian или Ubuntu.

Platform as a Service (PaaS, «платформа как услуга») — модель предоставления облачных вычислений, при которой потребитель получает доступ к использованию информационно-технологических платформ: операционных систем, систем управления базами данных, связующему программному обеспечению, средствам разработки и тестирования, размещённым у облачного провайдера. В этой модели вся информационно-технологическая инфраструктура, включая вычислительные сети, серверы, системы хранения, целиком управляется провайдером, провайдером же определяется набор доступных для потребителей видов платформ и набор управляемых параметров платформ, а потребителю предоставляется возможность использовать платформы, создавать их

виртуальные экземпляры, устанавливать, разрабатывать, тестировать, эксплуатировать на них прикладное программное обеспечение, при этом динамически изменяя количество потребляемых вычислительных ресурсов.

В 2011 году мировой рынок публичных PaaS оценён в сумму около \$700 млн, в числе 10 крупнейших провайдеров указываются Amazon.com (Beanstalk), Salesforce.com (Force.com, Heroku, Database.com), LongJump, Microsoft (Windows Azure), IBM (Bluemix), Red Hat (OpenShift), VMWare (Cloud Foundry), Google (App Engine), CloudBees, Engine Yard.

В 2012 году в OASIS предложен стандарт для прикладного программного интерфейса управления облачными платформами CAMP (англ. *cloud application management for platforms*), определяющий унифицированные форматы для команд программного управления облачными платформами (таких как запуск, приостановка, запрос на выделение ресурсов).

Джеймс Линденбаун, Адам Уиггинс и Орион Генри основали Heroku в 2007 году в поддержку проектов, основанных на Rack. 8 декабря 2010 года компания Salesforce.com купила Heroku, сделав её своей дочерней компанией. 12 июля 2011 года Юкихино Мацумото, создатель языка программирования Ruby, пришёл в компанию на должность ведущего инженера. В этом же месяце Heroku внедрила поддержку Node.js и Clojure. 15 сентября 2011 года Heroku и Facebook представили новую опцию «Heroku для Facebook». Heroku также предоставляет поддержку таких систем управления базами данных, как Cloudant, Membase, MongoDB и Redis, помимо основной — PostgreSQL.

Приложения, работающие на Heroku, используют также DNS-сервер Heroku (обычно приложения имеют доменное имя вида «*имя_приложения.herokuapp.com*»). Для каждого приложения выделяется несколько независимых виртуальных процессов, которые называются «dynos». Они распределены по специальной виртуальной сетке («*dynos grid*»), которая состоит из нескольких серверов. Heroku также имеет систему контроля версий Git.

Регистрация

Первым делом вам будет необходимо зарегистрироваться на Heroku. После того, как вы подтвердите свой e-mail нужно будет заполнить еще немного данных в своем профиле.

В общем, ничего сложного, ничего необычного. Вход в свой профиль и инициализация git-репозитория

Установка Heroku Toolbelt

Первым делом необходимо установить ПО для командной строки от Heroku на свой компьютер, если вы еще этого не сделали.

Для Ubuntu и Debian процесс установки заключается в выполнении следующей команды:

```
user@host:~/rails/myapp$ wget -qO https://toolbelt.heroku.com/install-ubuntu.sh | sh
```

Для остальных дистрибутивов Linux-этой:

```
user@host:~/rails/myapp$ wget -qO- https://toolbelt.heroku.com/install.sh | sh
```

Вход в профиль с командной строки

После того, как весь необходимый инструментарий установлен, вам необходимо залогиниться в Heroku с командной строки:

```
user@host:~/rails/myapp$ heroku login
```

```
Enter your Heroku credentials.
```

```
Email: user@example.com
```

```
Password:
```

```
Could not find an existing public key.
```

```
Would you like to generate one? [Yn]
```

```
Generating new SSH public key.
```

```
Uploading ssh public key /home/user/.ssh/id_rsa.pub
```

Если вы еще не генерировали открытый ключ для SSH то вам будет предложено это сделать.

Инициализация git-репозитория

Если вы не используете систему контроля версий или используете, но отличную от git то вам придется создать локальный git-репозиторий на своем рабочем компьютере т.к. Heroku поддерживает только эту систему контроля версий:

```
user@host:~/rails/myapp$ git init && git commit -am "init"
```

Некоторые специфические моменты.

Чтобы web-приложение корректно работало необходимо добавить rails_12factor в Gemfile (если используете Rails 4) и там же указать версию ruby (для всех версий Rails):

```
gem "rails_12factor", group: :production
```

```
#  
# Другие ваши гет-ы  
#
```

```
# Версию Ruby следует указывать в конце файла  
ruby "2.1.2"
```

Обратите внимание, вы должны использовать эту же версию Ruby и у себя.

Если вы используете web-сервер отличный от WEBrick (Н-р: Thin), что, кстати, рекомендуется самим сервисом Heroku, то вам необходимо будет создать так же Procfile приблизительно следующего содержания:

```
web: bundle exec thin start -p $PORT
```

ЛИТЕРАТУРА

Heroku [Электронный ресурс] / Wikipedia. – 2017. / Режим доступа:

<https://ru.wikipedia.org/wiki/Heroku/>. – Дата доступа: 25.03.2017

Развертываем свой сайт на Heroku [Электронный ресурс] / Habrahabr. – 2017. / Режим доступа: <https://habrahabr.ru/post/232679/>. – Дата доступа: 25.03.2017

Cloud Application Platform [Электронный ресурс] / Heroku. – 2017. / Режим доступа: <https://www.heroku.com/> – Дата доступа: 25.03.2017

УДК 004.5(075.8)

Студ. А. А. Етко

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

СЕРВИСЫ РАСПОЗНАВАНИЯ И СИНТЕЗА РЕЧИ ПЛАТФОРМЫ IOS

Актуальность набора текста голосом на сегодняшний день очевидна. При разработке приложения для смартфона в виде чата, в который входит: поиск контактов, добавление контактов, создание одиночных и групповых чатов, обмен сообщениями – было принято решение использовать возможность управления программным средством и ввода сообщения – голосом.

Реализация своей собственной системы распознавания речи является очень сложной, трудоемкой и ресурсозатратной задачей, которую сложно выполнить в рамках данной работы. Поэтому предполагается интегрировать представленную технологию видео-

идентификации в системы распознавания речи, которые имеют для этого специальные возможности.

В качестве платформы для создания приложения выбрана операционная система iOS, которая разработана американской компанией Apple – для смартфонов, электронных планшетов и носимых проигрывателей, первоначально – для iPhone и iPod touch, позже – для таких устройств, как iPad и Apple TV.

Эта платформа предоставляет качественные сервисы распознавания и синтеза речи и отличается от других платформ, наличием голосового помощника Siri, который стремительно развивается, постоянно увеличивая базу языков. Распознавание речи – процесс преобразования речевого сигнала в цифровую информацию (например, текстовые данные). Обратной задачей является синтез речи.

Siri – облачный персональный помощник и вопросно-ответная система, программный клиент которой входит в состав iOS компании Apple. Данное приложение использует обработку естественной речи, чтобы отвечать на вопросы и давать рекомендации. Siri приспосабливается к каждому пользователю индивидуально, изучая его предпочтения в течение долгого времени.

Sirikit поддерживает ряд сценариев, каждый из которых определяет задачи, которые могут быть выполнены, в том числе следующие: звонки через голосовую связь по IP-протоколу, действия с сообщениями, платежи, обработку фотографий, тренировки, бронирование такси, CarPlay (только для производителей автомобилей), предварительный заказ столиков в ресторане (требуется дополнительная поддержка от Apple).

Существует два типа App Extensions (сервисов), которые используются для взаимодействия с Siri:

1) Intents Extension – обеспечивает взаимодействие Siri и Maps с контентом приложений и выполняет задачи, необходимые для реализации любого из поддерживаемых intents.

2) Intents UI Extension – предоставляет пользовательский интерфейс, который отображает внутренний контент либо Siri, либо Maps.

Очень важно, чтобы приложение обеспечивало Intents Extension для поддержки SiriKit. Однако можно и отказаться от предоставления UI Extension, так как Siri обрабатывает все взаимодействия с пользователем, используя стандартный, встроенный интерфейс. Intent Extension может поддерживать один или несколько Intents. Apple требует, чтобы мы, как разработчики, старались ограничивать число

Intent Extensions, так как это помогает поддерживать использование памяти на низком уровне и повышает производительность Siri.

Системы распознавания речи классифицируются:

1) по размеру словаря (ограниченный набор слов, словарь большого размера);

2) по зависимости от диктора (дикторозависимые и дикторонезависимые системы);

3) по типу речи (слитная или отдельная речь);

4) по назначению (системы диктовки, командные системы);

5) по используемому алгоритму (нейронные сети, скрытые Марковские модели, динамическое программирование);

6) по типу структурной единицы (фразы, слова, фонемы, дифоны, аллофоны);

7) по принципу выделения структурных единиц (распознавание по шаблону, выделение лексических элементов).

Помехозащищённость систем автоматического распознавания речи, обеспечивается, прежде всего, использованием двух механизмов:

1) использование нескольких, параллельно работающих, способов выделения одних и тех же элементов речевого сигнала на базе анализа акустического сигнала;

2) параллельное независимое использование сегментного (фонемного) и целостного восприятия слов в потоке речи.

Работа с речью полностью завязана на доступе к Интернету, и это неудивительно, так как используется тот же самый движок Siri. Разработчику даётся возможность либо распознавать живую речь, либо речь, записанную в файл.

Но так как Speech.framework не содержит специального API для работы с микрофоном, разработчику потребуется задействовать AV Audio Engine. AV Audio Engine – это пространство для узлов, таких как схема или граф. Из возможных типов узлов строится цепочка обработки сигнала. Соединяя их различными способами, мы получаем требуемый эффект. Рассмотрим работу с real-time диктовкой. Вся работа представлена на принципиальной схеме (рис. 1):

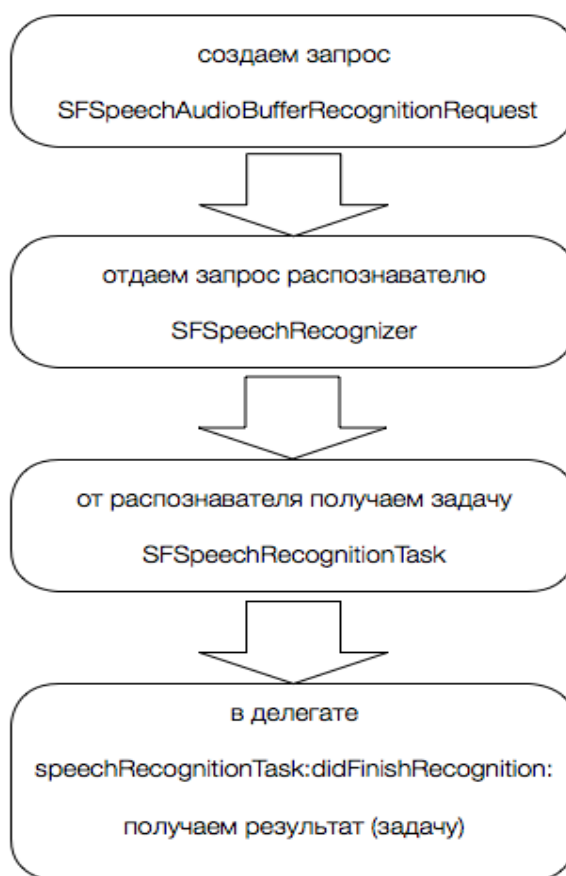


Рисунок 1

Преимущества преобразования голоса в текст:

- 1) возможность использовать свой голос, чтобы отправлять SMS-сообщения или электронные письма. Можно забыть о неудобном печатании и практически полностью ограничить использование клавиатуры;
- 2) возможность перевести текст на 32 иностранных языка;
- 3) корректировка текста с помощью голосовой диктовки или ввода с клавиатуры;
- 4) полная поддержка русского языка.

Независимо от того, насколько быстро мы вводим с клавиатуры текст, этот способ никогда не сможет работать быстрее, чем мы произносим слова. К счастью, современные технологии не стоят на месте, и преобразование голоса в текст затрагивает все больше и больше областей в нашей жизни. И скоро мы станем свидетелями серьезной конкурентной борьбы в области продвижения технологий преобразования голоса в текст. Чей алгоритм окажется лучше – покажет время.

УДК 004.5(075.8)

Студ. Г.В.Суша

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

ОСОБЕННОСТИ РАЗРАБОТКИ ВЕБ-СИСТЕМЫ ДЛЯ ПРОДАЖ СТРОИТЕЛЬНЫХ КОМПАНИЙ

В связи с возрастанием роли сети Интернет в жизни современного общества увеличивается потребность в создании уникальных сайтов для предприятий, компаний, частных лиц. Стремительное развитие мобильных технологий делает возможным создавать их удобные и компактные аналоги – мобильные приложения, значительно упрощающие доступ пользователей к сети вне зависимости от их местонахождения. Наличие большого количества разнообразных сайтов, неудобный интерфейс, навязчивая реклама, устаревшая и недостоверная информация, низкая наполняемость сайта – всё это делает затруднительным поиск необходимого предприятия или заведения среди обилия остальных. Правильным решением этой проблемы будет создание универсального сайта-каталога, в котором будет размещен список всех компаний и их товаров для покупателя.

Для того что бы решить эту проблему было принято решение разработать веб-систему, состоящую из сайта и приложения, для мобильных устройств под управлением операционной системы Windows Mobile 10. Схема веб-системы представлена на рис.1.

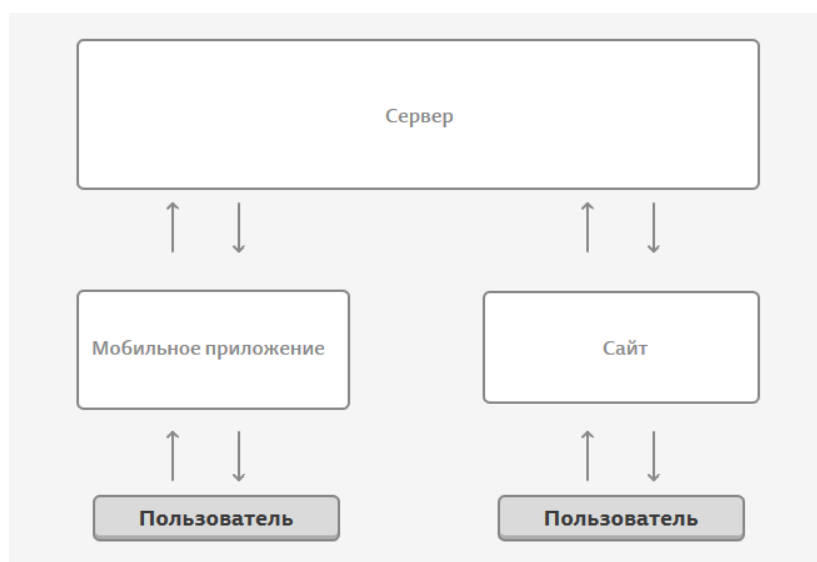


Рисунок 1 – Схема работы веб-системы

Сайт будет являться площадкой для продажи товаров, куда компании будут размещать свой товар, а покупатели выбирать товар из большого количества предложений. Макет страницы показан на рис. 2.

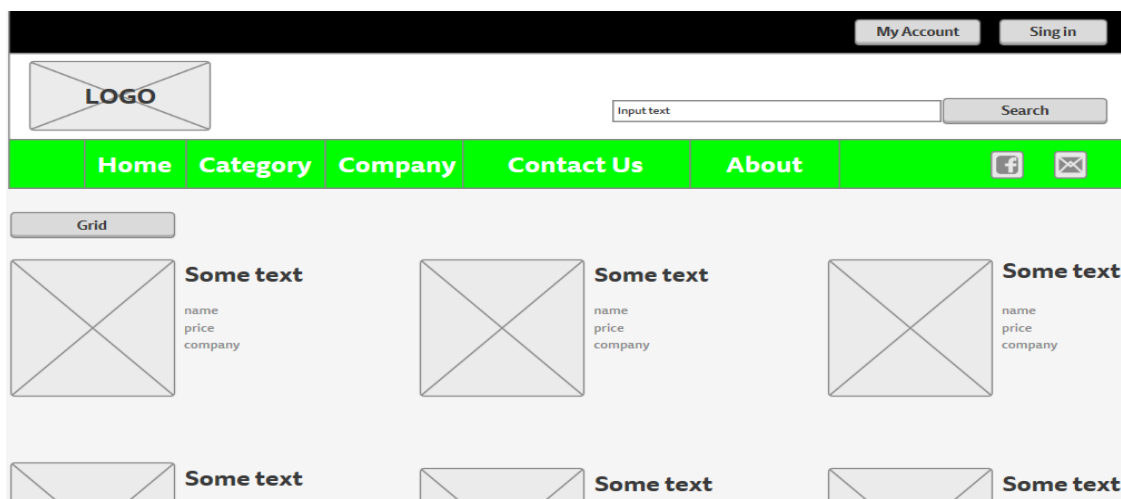


Рисунок 2 – Макет страницы товаров

Мобильное же приложение будет включать в себя как основной сайт, так и часть администрирования для пользователей, что позволит в любом месте в любое время с устройств, поддерживающих данное приложение, просматривать и администрировать свой раздел на сайте (рис. 3).

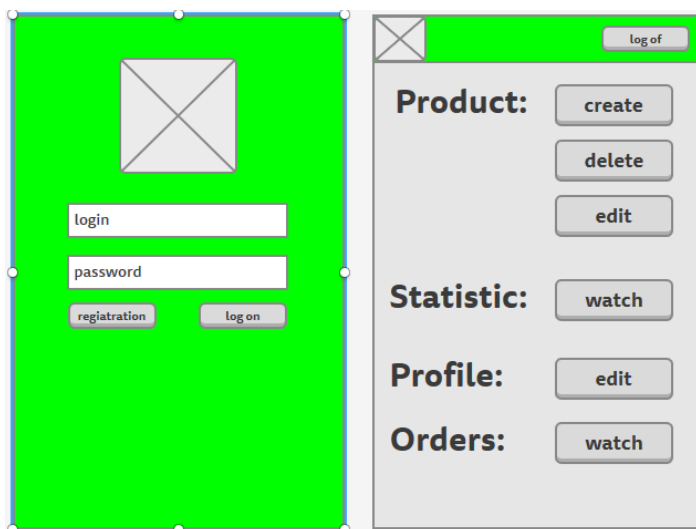


Рисунок 3 – Макет страницы мобильного приложения

Почему было выбрано мобильное приложение, а не просто мобильная версия сайта? Во-первых, не все мобильные браузеры поддерживают стандарты Java Script. Во-вторых, производительность

мобильных устройств не всегда позволяет обработать всю анимацию и все скрипты на сайте. В-третьих, с точки зрения энергоэффективности приложение более эффективно, чем сайт, поскольку скрипты на сайте даже при полной загрузке страницы порой требуют ресурсы для выполнения своего сценария.

Для обработки запросов сайта и хранения базы данных товаров и компаний будет использоваться сервер от Microsoft (Microsoft Azure), что даст нам требуемую скорость обработки данных и стабильную работу, при этом не тратив средств на оборудование собственных серверов.

В основе работы Microsoft Azure лежит запуск виртуальной машины для каждого экземпляра приложения. Разработчик определяет необходимый объем для хранения данных и требуемые вычислительные мощности (количество виртуальных машин), после чего платформа предоставляет соответствующие ресурсы. Когда первоначальные потребности в ресурсах изменяются, в соответствии с новым запросом заказчика платформа выделяет под приложение дополнительные или сокращает неиспользуемые ресурсы дата-центра.

Данная облачная система, была признана самой быстрой облачной платформой, что является, несомненно, плюсом для разработки нашей системы. Кроме этого Microsoft предоставляет также безопасную среду выполнения, что обеспечивает безопасность на уровне операционной системы и инфраструктуры.

Microsoft Azure подходит под все требования нашей системы, имея при этом мощный инструмент аналитики работы системы и нужный уровень безопасности данных.

Таким образом, разработка и внедрение подобных веб-систем позволит поднять сервис в сфере интернет услуг на качественно новый уровень.

УДК 004.41

Студ. Д.С. Бриль, С.А. Кузьмина, Д.И. Самасюк
Науч. рук. асс. И. А. Миронов
(кафедра информатики и веб-дизайна, БГТУ)

**СОЗДАНИЕ WEB-РЕСУРСА «WAYS TO HELP»
РАССЧИТАННЫЙ НА БЛАГОТВОРИТЕЛЬНУЮ
ДЕЯТЕЛЬНОСТЬ**

Благотворительность была бы не нужна, если бы мы жили в каком-то идеальном мире, где все без исключения друг другу

помогают, где деньги сами собираются, где вся помощь организовывается сама по себе. Тогда все прекрасно, и не надо, конечно, ни о чем кричать. Но мы живем в другом мире, где ничего само собой не организовывается.

Поэтому мы, выступая от лиц благотворительных организаций, предлагаем 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.

Фандрайзинг – деятельность некоммерческой организации, базирующаяся на ее уникальной миссии и стратегии, использующая эффективные и продуктивные способы получения ресурсов, необходимых для реализации программ и достижения стоящих перед организацией целей, обеспечивающая желаемую удовлетворенность дарителю (источнику ресурсов) и имеющая конечным результатом укрепление благополучия общества в целом.

Данный социальный проект предполагает создание международной платформы, которая позволяет:

- 1) благотворительным организациям публиковать информацию о ребенке, которому необходима транспортировка за границу, чтобы пройти лечение или курс реабилитации;
- 2) благотворителям приобретать мили за свои личные средства или жертвовать часть накопленных миль своих авиаперевозчиков.

Например, на данной платформе регистрируется международная благотворительная организация «ЮниХелп» и размещает информацию о ребенке, которому нужно лететь в Германию из Беларуси на операцию. Предположим, что из Минска до Берлина данный путь составляет 1200 миль, а стоимость билета на такое-то число – 240\$ с человека. Так как летит ребенок с мамой туда-обратно, то соответственно количество миль составит 4800, а их стоимость – 960\$. Т.е. выходит, что одна миля стоит 20 центов. Благотворитель оплачивает то количество миль, которые он хочет пожертвовать, и эти деньги переводятся на счет организации, разместившей пост.

Наш WEB-ресурс дает новую возможность участвовать в благотворительности, а также популяризировать благотворительность.



Рисунок 1 – Логотип «Ways to help»

Есть еще одна возможность переводить мили, но не конкретно на ребенка, а на организацию. Суть состоит в следующем. Например, белорусская авиакомпания «Белавиа» предлагает своим пассажирам программу под названием «Лидер», а российская авиакомпания «Аэрофлот» – «Бонус», которые дают возможность своим пассажирам-участникам накапливать баллы на персональном счете за регулярные полеты, которые потом можно обменивать на билеты. Так вот, в этом приложении можно осуществлять социальное партнерство благотворительных фондов с авиаперевозчиками, которое позволит пассажирам данной авиакомпании перечислять накопленные мили на счет организации.

Если вы выступаете от лица фонда, то при регистрации указывается информация подтверждающая законную деятельность фонда. После регистрации появляется возможность создавать посты с детьми для сбора средств, но до публикации правильность поста проверяется администрацией сайта. После всех этих шагов информация о ребенке появляется на сайте и начинается кампания по сбору средств для переезда в другую страну в целях лечения или реабилитации.

Сайт должен быть понятным как пользователям, так и благотворительным фондам, чтобы вызвать больший интерес. Помощь должна осуществляться более просто, не заняв много времени у пользователя.

Посетив сайт (рис. 1) вы сразу же увидите 2 кнопки, которая дает выбор: выступать от роли фонда, либо быть рядовым пользователем, который не равнодушен.

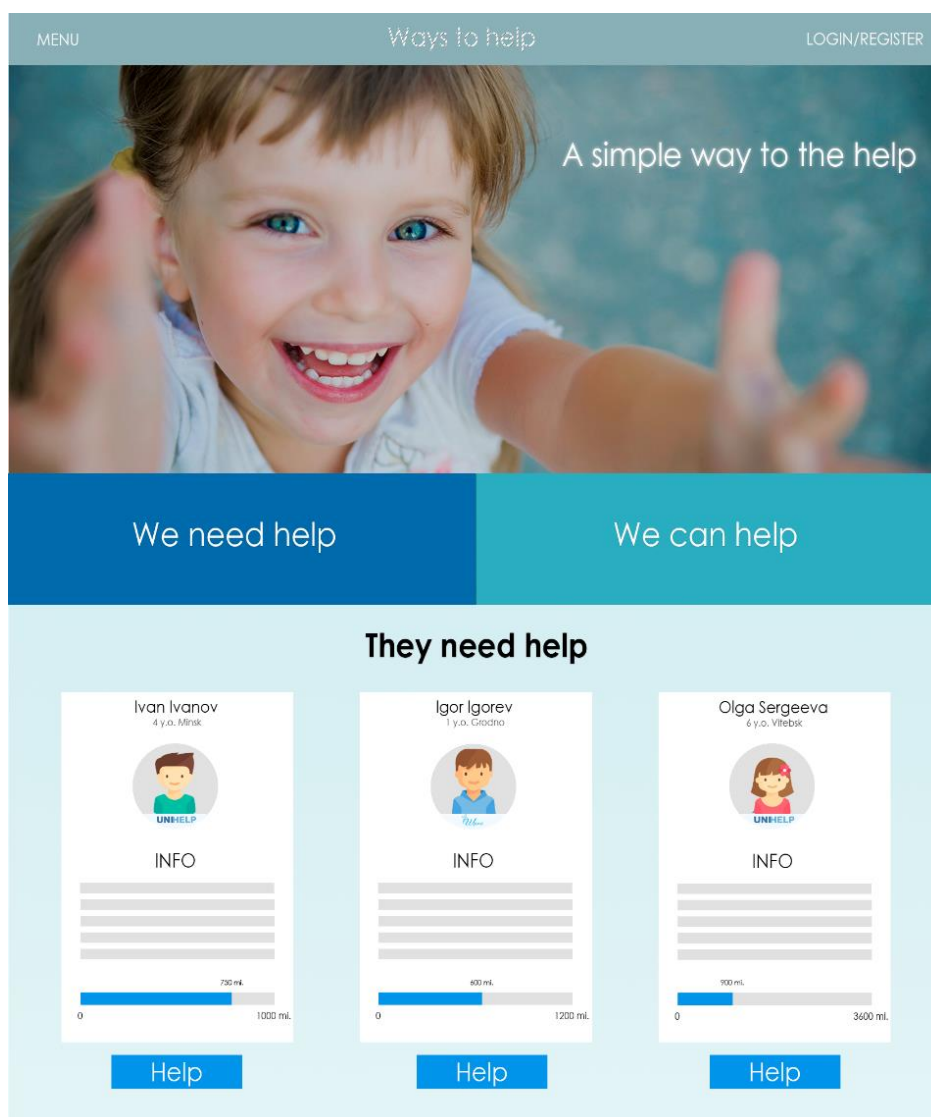


Рисунок 1 – Web-ресурс «Ways to help»

Если вы выступаете от физического лица и пришли просто помочь, то при регистрации необходимо привязать свой аккаунт к аккаунту в авиакомпании для того, чтобы свободно делиться накопившимися милями, но если такого аккаунта нет, то этот шаг пропускается и заполняется платежная информация.

На ресурсе есть вся необходимая информация о ребенке, а также о фонде, который разместил пост с этим ребенком.

После заполнения нужной информации вы можете в один клик помочь ребенку.

УДК 004.41

студ. В. В. Бугаенко, Д. Д. Карнеева
 Науч. рук. асс. И. А. Миронов
 (кафедра информатики и веб-дизайна, БГТУ)

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ IBOOKS AUTHOR ДЛЯ ВИЗУАЛИЗАЦИИ КОНТЕНТА

Современные студенты плохо воспринимают традиционную модель обучения, в связи с чем необходимо разработать способы обучения, обладающие мобильностью, практико-ориентированностью, самопополняемостью контента, рецензированием контента. Одним из решений данной задачи является электронная книга.

Электронная книга приходит на смену бумажной. Новые технологии дают новые возможности. Она может и должна включать в себя всё многообразие современных средств мультимедиа. Таких, чтобы читатель мог в процессе чтения просмотреть видеосюжет о прочитанном, прослушать аудио-фрагмент, не только посмотреть статичное изображение, но и поработать с трехмерной моделью, посмотреть комментарии к тексту и изображениям, пройти тест сразу после прочтения.

Для представления учебного контента в виде электронного интерактивного учебника существует множество приложений, большинство из которых доступны по лицензии типа *open source*. В таблице 1 представлены некоторые приложения, которые могут быть использованы для создания интерактивных учебников.

Таблица 1

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

Каждое из представленных в таблице 1 приложений позволяет работать с несколькими интерактивными элементами. Приложение iBooks Author обладает наибольшим функционалом для разработки электронных учебников: управление контентом при помощи тезауруса; представление содержимого в виде текста, видео, диаграмм, галереи графических рисунков; размещение 3D-моделей, java-тегов; поддержка тестирования и др.

В 2013 году авторами была проведена работа по созданию электронных учебников при помощи приложения iBooks Author. При разработке интерактивных учебников с использованием приложений MacOS особое внимание уделялось структурированию и визуализации учебного контента, разработке отдельных элементов контента. На рис. 1 представлены основные виджеты приложения iBooks Autor для разработки интерактивного контента.



Рисунок 1 – Основные виджеты iBooks Author

За счет создания библиотеки виджетов сохраняется простота создания учебника преподавателем, а за счет интеграции виджетов в содержание учебника появляется возможность изменять содержание виджетов (на пример, кейсы, тесты, страницы с полезным содержанием, видео и аудиоматериал) без необходимости загрузки учебника заново. Кроме того, статистика использования виджетов позволяет анализировать процесс изучения материала студентами, а

также получать обратную связь об актуальности материала. Для реализации варианта необходимо создание библиотеки виджетов, что несколько увеличивает стоимость за счет привлечения программистов для создания библиотеки.

Ещё недавно создание таких книг требовало глубоких знаний в области программирования. Программа iBooks Author решает множество задач, требуя от автора 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 ИКТ (информационно-коммуникационных технологий), делая акцент на создании увлекательных современных учебников.

Для создания книги в iBooks Author и просмотра всех её возможностей и эффектов требуется iPad с установленной программой iBooks 3. Программа работает под управлением операционной системы Mac OS X 10.7.4 или более поздней версии.

В любой момент в программе iBooks Author можно просмотреть, как будет выглядеть готовая книга. Кроме того, на iPad и Mac всегда доступна функция предварительного просмотра книги.

Использование дополнительных возможностей устройства – Apple iPad – также добавляет ценности данного решения для слушателей. Однако это одновременно является и недостатком, поскольку учебник изначально работает только на Apple iPad, а пользователь должен иметь iPad для пользования учебником. К недостаткам можно отнести также увеличение стоимости обучения с учетом стоимости iPad и его доставки.

iBooks Author помогает автору не только создать книгу, но и самостоятельно опубликовать её в Интернете. Существует несколько вариантов размещения учебного электронного контента типа .ibooks в сети. Во-первых, учебник можно опубликовать в интернет-магазине iTunes Store в разделе AppStore (в открытом или платном доступе), однако данный вариант вызывает некоторые сложности. Публикация платной книги в iBooksStore предполагает наличие американского номера налогоплательщика и предоставление информации о банковском счете, на который будут поступать деньги. Размещение учебника в открытом доступе не вызывает проблем, но пользователям с белорусскими аккаунтами Apple учебник не будет доступен в связи с отсутствием белорусского сектора iBooksStore. 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. При создании нашей книги мы воспользовались вторым вариантом.

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. Книгу можно сохранить как пакет *.itmsp и опубликовать в iBookstore.

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.

Программа iBooks Author отлично решает вышеставленные задачи, позволяя не только создавать электронные книги, но и делает процесс обучения более увлекательным и запоминающимся.

ЛИТЕРАТУРА

1. Спиридонов О. В. Создание электронных интерактивных мультимедийных книг и учебников в iBooks Author / Спиридонов, О. В. – М.:Национальный Открытый Университет «ИНТУИТ», 2016.

УДК 004.75

Студ. Д. Л. Клыбик

Науч. рук. асс. И. А. Миронов

(кафедра информатики и веб-дизайна, БГТУ)

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ ВИРТУАЛИЗАЦИИ ДЛЯ СОЗДАНИЯ ОТКАЗОУСТОЙЧИВОГО КЛАСТЕРА

Виртуализация — это процесс создания программного (или виртуального) представления чего-либо в противоположность его физической реализации. Виртуализацию можно применять для приложений, серверов, систем хранения и сетей.

Это единственный и самый эффективный способ сокращения расходов на ИТ-инфраструктуру с возможностью повышения эффективности и адаптивности для компаний любых размеров. На сегодняшний день ресурсы большинства серверов используются менее чем на 15%, что приводит к росту числа серверов, а сама инфраструктура становится все более сложной и дорогой.

Виртуализация серверов может решить эти проблемы благодаря возможности запускать на одном физическом сервере несколько операционных систем в виде виртуальных машин, каждая из которых имеет доступ к вычислительным ресурсам сервера (рис. 1).

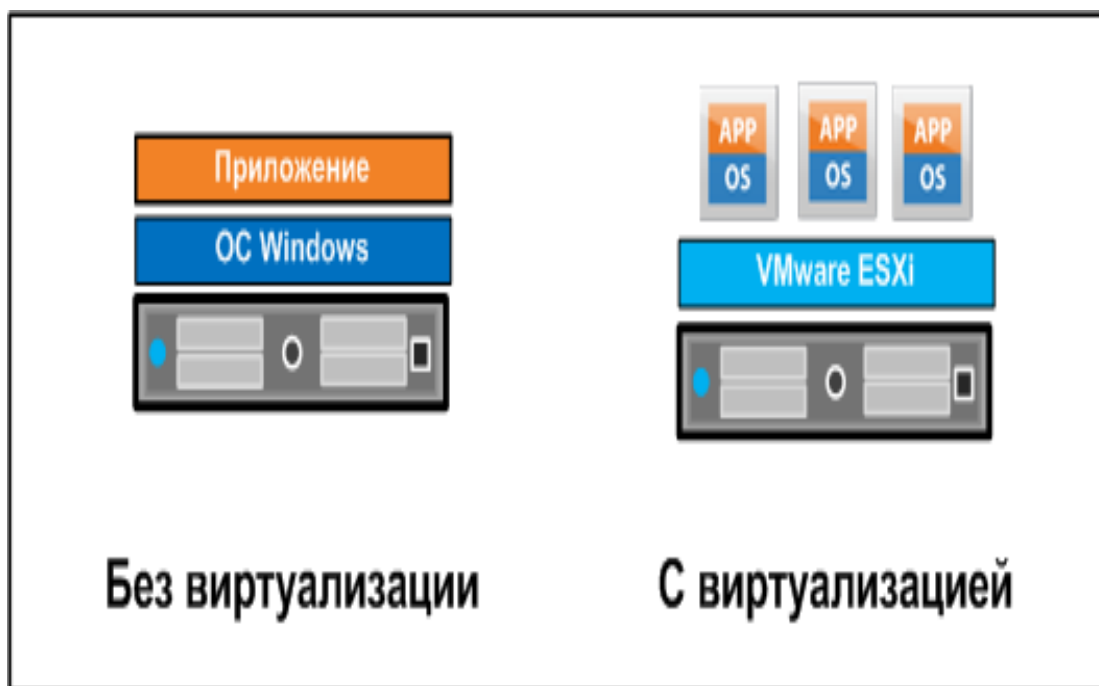


Рисунок 1 – Технология виртуализации

Гипервизор – это платформа виртуализации, позволяющая запускать на одном физическом компьютере несколько операционных систем. Именно гипервизор предоставляет изолированное окружение для каждой виртуальной машины, и именно он предоставляет гостевым ОС доступ к аппаратному обеспечению компьютера.

Отказоустойчивый кластер — это комбинация из одного или более узлов (серверов) на двух или более общих дисках, которые называются группой ресурсов (рис. 2). В сети отказоустойчивый кластер представлен как один компьютер, но при этом он обеспечивает переход на другой узел в случае, если текущий узел становится недоступным. Отказоустойчивый кластер в сети выступает в роли обычного приложения или отдельного компьютера, но поддерживает дополнительные возможности, увеличивающие его доступность.

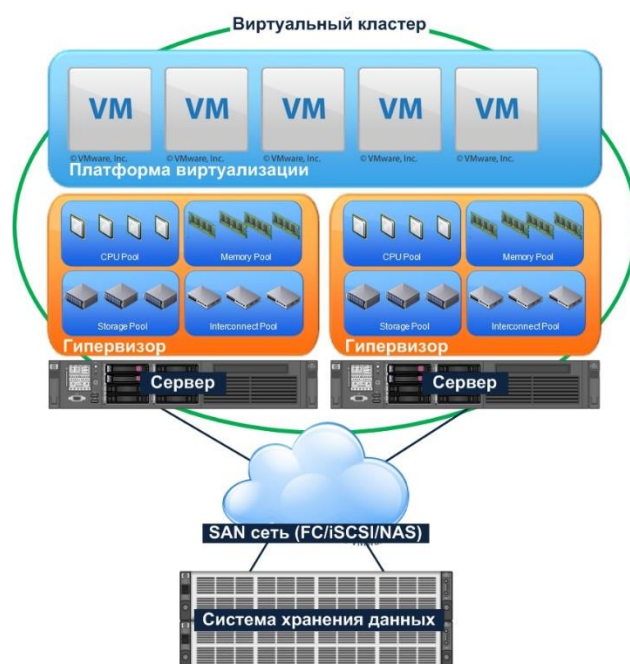


Рисунок 2 – Архитектура отказоустойчивого кластера

vSphere ESXi — это аппаратный гипервизор, который устанавливается непосредственно на физический сервер и разделяет его ресурсы на несколько виртуальных машин.

VMware vCenter — это централизованная платформа для управления средами VMware ESXi, с помощью которой можно автоматизировать виртуальную инфраструктуру и предоставлять к ней доступ.

Технология High Availability позволяет свести к минимуму простои с помощью универсальной экономичной системы аварийного переключения при сбоях оборудования и операционных систем в виртуальной среде. Она отслеживает состояние серверов и виртуальных машин, а при возникновении сбоев выполняет автоматический перезапуск виртуальных машин на других серверах кластера. Компонент обеспечивает универсальную автоматическую защиту всех приложений без внесения изменений в эти приложения и гостевую операционную систему.

VMware DRS — технология, которая выбирает на каком физическом хосте запустить виртуальную машину, или на какой хост отправить работающую виртуальную машину, с целью выровнять нагрузку на всех хостах кластера (рис. 3). DRS объединяет ресурсы кластера в один большой пул и в полностью автоматизированном режиме распределяет виртуальные машины равномерным слоем по физическим хостам кластера.

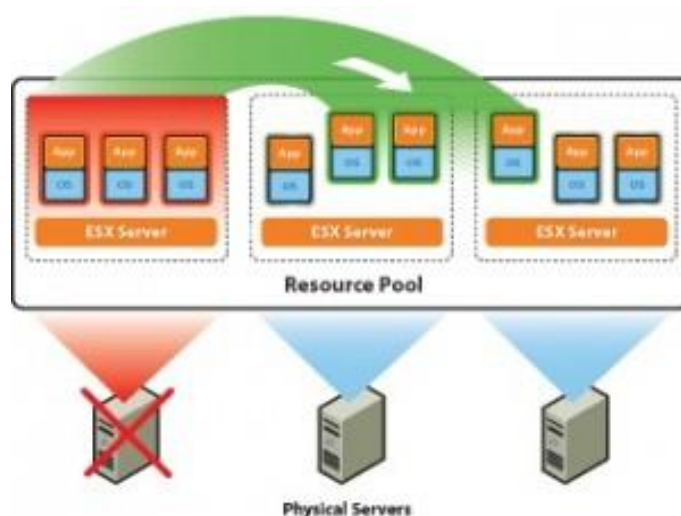


Рисунок 3 – Принцип работы HA + DRS

vSphere vMotion обеспечивает перенос работающей виртуальной машины с одного физического сервера на другой без простоя. При этом сохраняются сетевые параметры и сетевые подключения виртуальной машины, что обеспечивает ее беспрепятственный перенос (рис. 4). Копирование активной памяти и точного рабочего состояния виртуальной машины осуществляется по высокоскоростной сети, благодаря чему работающая виртуальная машина переносится с одного сервера vSphere на другой. В гигабитной сети Ethernet весь процесс занимает не более двух секунд. Эта возможность доступна при использовании виртуальных коммутаторов, сервера vCenter, а также при значительных расстояниях между серверами.

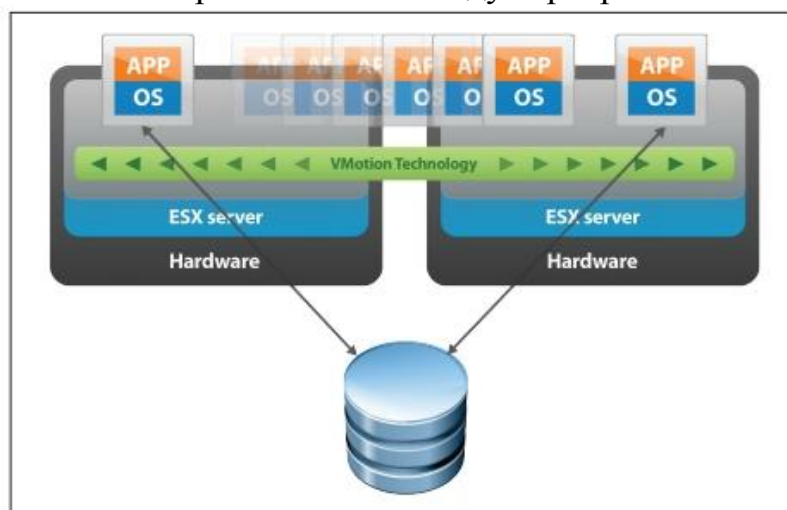


Рисунок 4 – Принцип действия переноса виртуальных машин

УДК 004.438

Студ. Н. А. Лаптев, М. В. Дубовик

Науч. рук. ассист. И. А. Миронов

(кафедра информатики и веб-дизайна, БГТУ)

НОВЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ JAI

В текущий момент программирование игр является одним из направлений программирования в целом. Для такого типа программирования очень важна скорость обработки кода, т.к. графика в играх очень динамична и пользователю будет неприятно работать с временными задержками. Для выигрыша в скорости удобно использовать низкоуровневые языки, такие как язык ассемблера, но они очень объёмные по коду, поэтому для разработки обычно выбор падает на языки среднего уровня, такие как C/C++.

В 2014 году разработчик игр Джонатан Блоу предложил концепцию нового языка программирования Jai, ориентированного преимущественно на разработку игр. Компилятор под данный язык на текущий момент разрабатывается Блоу. Автор анонсировал, что новый язык будет выигрывать в скорости обработки кода у языка C. Данный факт заставляет рассмотреть данный язык более подробно.

Jai – высокоуровневый, процедурный, строго типизированный, C-подобный язык программирования. Описание языка предполагает его как современную замену языку C.

Среди особенностей языка можно отметить:

– Произвольное выполнение кода во время компиляции. Любая функция программы может быть запущена во время компиляции с помощью *#run*;

– рефакторинг кода с облегченным синтаксисом. Синтаксис языка облегчает повторное использование кода, упрощая его перемещение из *локального блока* → *локальной функции* → *глобальной функции*;

– интегрированный процесс сборки. Процесс сборки и параметры задаются самим исходным кодом для обеспечения согласованности;

– автоматическое преобразование между структурой массивов и массивом структур, избегает классы и наследование;

– рефлексия (процесс, во время которого программа может отслеживать и модифицировать собственную структуру и поведение во время выполнения) и информация о типе времени выполнения. Информация о статическом типе для каждой структуры доступна во время выполнения;

- новый подход к полиморфным процедурам. Полиморфизм на функциональном уровне, с программным управлением специальными процедурами;

- низкоуровневые инструменты управления памятью. Более эффективный контроль над тем, как библиотеки выделяют память, автоматическое управление собственностью, сборка мусора отсутствует;

- явный контроль над оптимизацией и производительностью. Явный контроль над такими вещами, как встраиваемые функции (inline-функции), «проверка границ» и инициализация.

Основная идея Блоу состоит в том, что видеоигры – это огромные блоки памяти, которые перемещаются последовательно с жёсткого диска в оперативную память, с оперативной памяти загружаются в видеокарту или кэш процессора для обработки, а далее возвращаются в оперативную память. Большое время ожидания выполнения этих процессов не нравится пользователям данных игр. Автор сказал, что использование сборщиков мусора, шаблонов и классов помогают быстрее писать код программистам, но это не делает их код более быстрым.

Особо привлекательной выглядит попытка оптимизации памяти при работе с массивами структур. Её суть состоит в том, что массивы структур при явном указании соответствующего ключевого слова SOA преобразуются к структуре массивов, что позволяет сократить размеры неиспользованной памяти.

Важная цель в проектировании Jaі – уменьшение того, что Блоу называет «трением» в программировании. «Трение» случается, когда синтаксис языка мешает работе программиста. Создатель языка утверждает, что увеличение производительности и уменьшение «трения» при отсутствии механизмов защиты памяти более рационально, чем восполнение времени, потраченного на отслеживание ошибок, особенно если учесть, что хорошие программисты допускают относительно небольшое количество ошибок. Поэтому в Jaі низкая терпимость к «трению», особенно когда «трение» не нужно.

Джонатан Блоу создал концепцию такого языка, который будет ориентирован на программирование игр, но, как отметил сам автор, язык может быть использован и в других направлениях программирования. Даже, если язык после создания компилятора не получит большого распространения, то идеи Джонатана Блоу могут

послужить отправной точкой для дальнейшего развития других языков программирования.

УДК 025.5:004(072)

Студ. Е.А. Богатырев

Науч. рук. доц. Романенко Д.М., ассист. Т.П. Брусенцова
(кафедра информатики и веб-дизайна, БГТУ)

ОСОБЕННОСТИ РАЗРАБОТКИ КЛИЕНТСКОЙ ЧАСТИ ИНФОРМАЦИОННО-ПОИСКОВОГО СЕРВИСА В СФЕРЕ УСЛУГ

Информационно-поисковые системы являются неотъемлемой частью жизни современного человека. С появлением Всемирной паутины и сети Интернет люди получили доступ к огромному объему информации, расположенной по всему миру. Своевременно появились специализированные информационно-поисковые системы, которые собирали, хранили, обрабатывали и осуществляли поиск по информации определенного характера: информация об отелях и свободных номерах, сеансах в кинотеатрах, постановках в театрах, концертах. Так начали появляться информационно-поисковые системы в сфере услуг.

Информационно-поисковая система – это компьютерная система, предназначенная для поиска информации. Сейчас наиболее распространенным видом таких систем является веб-приложение, построенное по архитектуре «клиент-сервер». Это обусловлено тем, что позволяет обеспечить одновременную работу большого числа пользователей. Клиентское приложение предоставляет пользовательский интерфейс, взаимодействуя с которым пользователь формирует запрос на получение нужной ему информации и получает результат.

В данный момент широкое распространение получили следующие виды приложений:

1. WEB-приложение;
2. Мобильное приложение;
3. Desktop приложение.

WEB-приложение – ПО, исполняемое в интернет браузере, использующее протоколы HTTP/HTTPS для обмена данными. Достоинствами такого приложения является независимость от операционной системы, платформы, удобство распространения приложения, обновлений. К недостаткам можно отнести потребление

интернет трафика, т.к. необходимо загружать javascript библиотеки, таблицы стилей и другие ресурсы, а также некоторые проблемы с отображением приложения в различных браузерах.

Мобильное приложение – ПО, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах. Достоинствами являются наличие простых механизмов распространения приложения и обновлений, удобство использования, безопасность. Недостатками являются зависимость от платформы, операционной системы и ее версии.

Desktop приложение – ПО содержащая в себе всю полную функциональность и работающая стационарно на машине пользователя. Достоинствами являются: безопасность, надежность, отсутствие необходимости в интернет соединении, высокая скорость работы. Недостатками являются: зависимость от платформы и операционной системы, сложности с распространением приложения и обновлений.

Рассмотрим клиент-серверное взаимодействия. Эти модели определяют роль клиентского и серверного приложений в процессах работы с данными, модель взаимодействия представлена на рисунке 1.

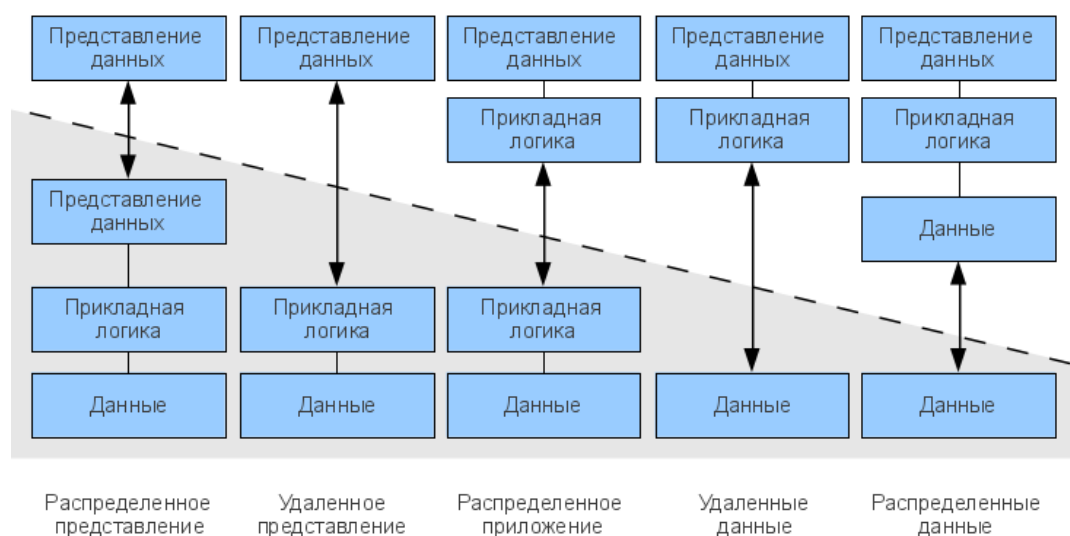


Рисунок 1 – Модели клиент-серверного взаимодействия

Данные модели определяют где будут храниться данные, где будет происходить обработка данных и формирование представлений, которые потом будут отображаться в клиентском приложении. Выбор модели целиком зависит от задач, которые должно решать приложение.

Информационно-поисковая система может возвращать в качестве ответов большие объемы данных, часть из которых пользователю может и не понадобиться, необходимо реализовать механизмы, способные сократить затраты интернет трафика на получение информации. Основными механизмами являются:

1. Кэширование данных;
2. Поэтапная загрузка данных;
3. Сжатие данных.

Кэширование – это хранение часто используемых данных в кэш памяти. Т.е. можно реализовать хранение редко изменяемых данных, которые часто запрашиваются пользователем, на клиентской стороне. Также можно кэшировать представления, результаты SQL запросов, информацию о пользователе и многое другое. В таком случае перед запросом на получение информации от сервера, приложение сначала будет обращаться к кэшу. Если нужная информация присутствует в кэше, и она не является устаревшей, то приложение будет использовать ее. В противном случае будет выполняться запрос на сервер. Существуют следующие типы кэша:

1. Lazy cache, он же *ленивый кэш*. Кэш просто сохраняет данные и отдает их пока не устареет.

2. Synchronized cache, *синхронизированный кэш*. Такой тип кэширования позволяет всегда иметь свежие данные, хранит метку времени, но очень сложен в реализации.

3. Write-through cache, или *кэш сквозной записи*. Этот тип кэша может никогда не устаревать, но возникают проблемы с так называемой “*когерентностью*”.

Механизм поэтапной загрузки данных подразумевает дробление информации на блоки и загрузку этих блоков по необходимости или требованию пользователя. Существует два основных варианта реализации такого механизма:

- Пагинация – показ ограниченного количества информации на одной странице. Для получения следующего блока информации пользователь запрашивает следующую страницу.

- Scroll загрузка – механизм, вызывающий функции загрузки данных при достижении определенной позиции страницы. Новый блок информации будет загружаться в тот момент, когда пользователь пролистает уже полученный блок данных до определенной позиции.

Еще одна особенность – это удобство пользовательского интерфейса. Ниже приведены основные принципы разработки клиентского приложения, которым желательно следовать:

1. Принцип KISS. От англ. «keep it short and simple». Интерфейс простой и понятный, задачи должны решаться минимальным числом действий, все должно быть понятно и очевидно.
2. Не заставляйте думать. Нужно избегать сложных действий, которые заставляют пользователей думать.
3. Интуитивная понятность. Понимание лучше запоминания.
4. Все полезное на виду. Все важные элементы интерфейса должны быть на виду и соответствующим образом выделены.
5. Принцип 3х кликов. Должно быть не более 3х кликов для перехода из одного раздела в другой. Это же правило применимо к главной странице: любая важная информация должна быть доступна не более чем в 3 клика.
6. Принцип единства. Настройки и элементы управления нужно стараться не прятать в отдельные разделы, а дать возможность управлять из одного места, если это уместно.
7. Тренды. Стоит учитывать современные тенденции, чтобы интерфейс не устарел еще до выхода проекта.

Последней особенностью, которой следует уделить должное внимание при разработке клиентского приложения, является безопасность. Безопасностью клиентского приложения нельзя пренебрегать. Если информационно-поисковая система кроме поиска информации осуществляет оказание и других услуг, например, заказ авиабилетов или бронирование номеров в отелях, и предполагает передачу через сеть конфиденциальной информации, такой как паспортные данные или данные платежной карты, данный процесс необходимо обезопасить, во избежание возможности кражи этих данных. Также необходимо обезопасить и передачу логина и пароля, используемых пользователем для получения доступа к системе. Повысить безопасность передачи данных через сеть можно используя протокол HTTPS, а также используя другие механизмы шифрования данных.

ЛИТЕРАТУРА

1. Wikipedia. Поисковая система [Электронный ресурс] – https://ru.wikipedia.org/wiki/Поисковая_система – Дата доступа 25.04.2017
2. Учебно-методические материалы для студентов кафедры АСОИУ. Модели клиент-серверного взаимодействия [Электронный ресурс] – <http://www.4stud.info/networking/lecture5.html> – Дата доступа 18.04.2017

3. Хабрхабр. Стратегия кэширования в приложении [Электронный ресурс] – <https://habrahabr.ru/post/168725/> – Дата доступа 18.04.2017

4. Хабрхабр. Памятка UX / UI дизайнеру. 19 принципов построения интерфейсов [Электронный ресурс] – https://habrahabr.ru/company/SECL_GROUP/blog/182208/ – Дата доступа 18.04.2017

УДК 025.5:004(072)

Студ. А.А. Сакольчик

Науч. рук. доц. Д.М. Романенко

(кафедра информатики и веб-дизайна, БГТУ)

ОСОБЕННОСТИ РАЗРАБОТКИ СЕРВЕРНОЙ ЧАСТИ ИНФОРМАЦИОННО-ПОИСКОВОГО СЕРВИСА В СФЕРЕ УСЛУГ

Очень часто в нашей жизни нам приходится искать различную информацию, будь то рецепт нового блюда или номер телефона какой-то организации. Для этой и других целей были разработаны специальные поисковые системы. Поисковая система - компьютерная система, предназначенная для поиска информации. Т.к. эта система предназначена для поиска и предоставления пользователям какой-то информации, ее также можно назвать поисковой информационной системой.

Поисковой информационной системы должна предоставлять пользователям интерфейс для выбора интересующего их города, достопримечательностей, интересующих пользователя мест, а также информации о них, построения маршрутов по указанным требованиям. Интерфейс приложения должен быть простым, чтобы пользователь без лишнего труда мог разобраться в работе данного приложения. Должна поддерживаться одновременная работа большого числа пользователей. Сервер должен реализовывать гибкие способы получения, обработки, хранение и представление данных для пользователя. Так же сервер должен быть расширяемым, быстродействующим, удобной в администрировании. [1].

Что бы реализовать такой поисковой сервис необходим источник большого объема данных о городах с его достопримечательностями, кафе, ресторанах и т.д. Этим требованиям соответствует сервисы Google Maps API. Google Maps API это

мощный ресурс предоставляющий данные для создания сторонних сервисов на основе Google Map. Данный сервис выставлен в сеть Internet и может работать как через JavaScript интерфейс, или на одном из серверных языков, например, Java, используя протокол HTTPS.

Для построение информационно-поисковой системы используются следующие Google Map API сервисы [2]:

1. Google Places API Web Service. Реализуйте подсказки мест и добавляйте актуальную информацию о миллионах мест на свой сайт или приложение.
2. Google Map Direction API. Расчет маршрутов между различными точками.
3. Google Map Geolocation API. Поиск местоположения на основе информации, передаваемой по сотовой связи и Wi-Fi.

Google устанавливает ограничения на количество выполняемых запрос к Google Maps API в сутки [3], значения лимитов представлено на рисунке 1. С целью минимизирования обращения к Google, был разработан Фреймворк, который позволяет снизить количество запросов и реплицировать данные в собственную базу данных.

Наименование сервиса	Количество бесплатных запросов, запросов / сутки	Количество платных запросов, запросов / сутки	Количество запросов с привязанной банковской картой, запросов / сутки
Google Places API Web Service	1 000	>150 000	150 000
Google Map Directions API	2 500 (50 запросов / сек)	100 000	100 000
Google Map Geolocation API	2 500 (50 запросов / сек)	100 000 (50 запросов / сек)	100 000

Рисунок 1 - Google Maps API ограничения на количество запросов

Принцип работы разработанного Фреймворка можно изобразить в виде диаграммы, представленной на рисунке 2.

Процесс работы Фреймворка можно описать в качестве запросов и ответов от одного элемента схемы к другому.

Порядок взаимодействия и элементов схемы, изображенной на рисунке 2:

1. Пользовательский запрос к клиенту. Пользователь, через интерфейс приложения, подает входные данные для построения маршрута. В качестве входных данных могут выступать:

- a. Название страны;
- b. Название города;
- c. Название места посещения, таки как: достопримечательности, кафе, рестораны и т.д.;
- d. Последовательность посещения.

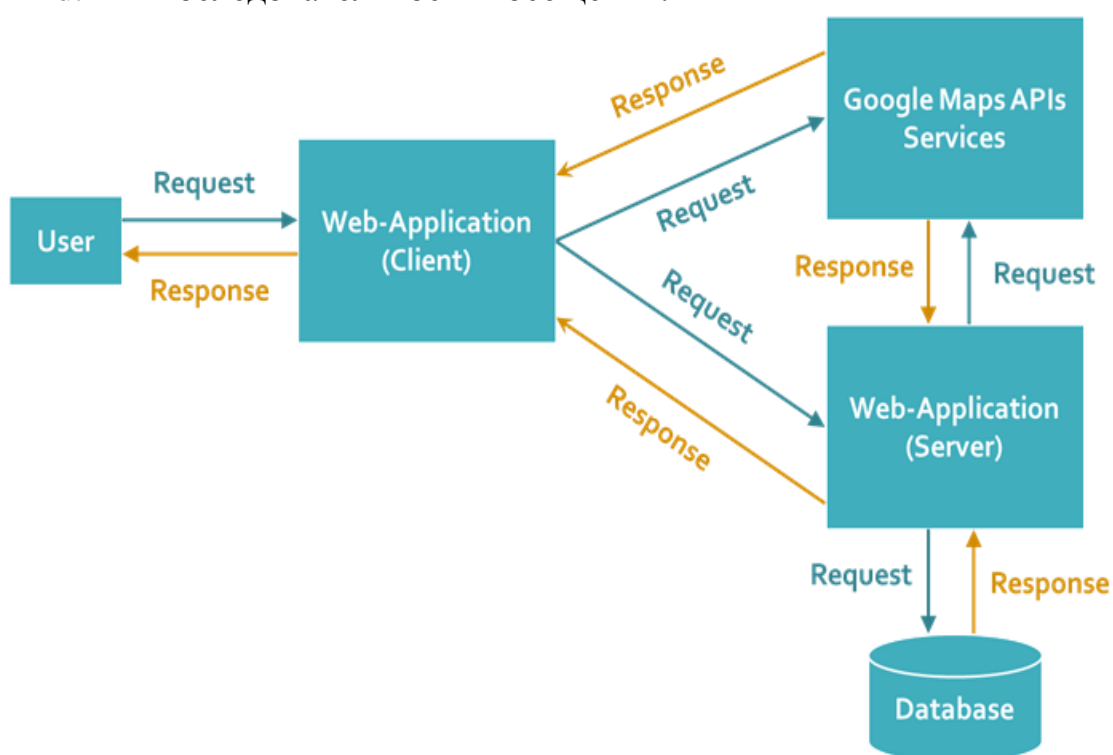


Рисунок 2 - Принцип работы Фреймворка

2. Клиентский запрос к сервису. Веб-приложение на стороне пользователя, далее клиент, выполняет запрос на Google Place API Web Services, используя Google Auto Complete механизм, для получения списка Google Places ID городов, достопримечательностей, ресторанов и кафе. Google Places ID – это уникальные идентификаторы сгенерированные специальным способом Google.

3. Ответ сервиса клиенту. Google Places API Web Service присылает ответ на Клиент содержащей Google Places ID с кратким описанием.

4. Клиентский запрос на сервер. Клиент обрабатывает полученные данные от Google и посылает запрос на сервер веб-приложения, далее просто сервер, с целью поиска в базе данных веб-приложения, далее просто база данных, по Google Places ID подробную информацию о запрашиваемых местах.

5. Запрос сервера к базе данных. Сервер выполняет запрос в базу данных с целью поиска мест, и в случае, если данные были найдены, подготавливает и отправляет ответ клиенту, в противном случае выполняет запрос к Google Places API Web Service для получения недостающих данных.

6. Запрос сервера к Google. Google Places API Web Service посылает ответ на сервер с подробным описанием мест, запрашиваемых по Google Places ID. Сервер принимает данные и сохраняет в базе данных.

7. Ответ сервера клиенту. Сервер отсылает запрошенные данные на клиент. Клиент представляет результат пользователю в удобном для него виде.

Таким образом можно сделать вывод, что особенностью разработки информационного-поискового сервиса является построение правильного процесса работы с большим объемом информации [4], правильное использование сторонних ресурсов такого как Google Maps API. Результатом работы это разработанный Фреймворк позволяющий реализовать собственную информационно поисковую систему в сфере услуг.

Спроектированный Фреймворк выделяет 5 основных элементов, такие как: пользователь, клиент, сервер, база данных, сторонние сервиса, и описывает порядок и правило их взаимодействия. На основании понятий запрос-ответ был описан процесс обработки запросов пользователя с целью получения информации от поискового сервиса.

ЛИТЕРАТУРА

5. Wikipedia. Поисковая система [Электронный ресурс] – https://ru.wikipedia.org/wiki/Поисковая_система – Дата доступа 24.04.2017
6. Google Maps API. Документация [Электронный ресурс] – <https://developers.google.com/maps/documentation/> – Дата доступа 24.04.2017
7. Google Maps API. Цены и планы [Электронный ресурс] – <https://developers.google.com/maps/pricing-and-plans/> – Дата доступа 24.04.2017
8. Когаловский, М. Р. Энциклопедия технологий баз данных. 2002, М.: Финансы и статистика – 800 с

УДК 655.533, 535.421

Студ. В. С. Бриль, С. А. Кузьмина
Науч. рук. ст. преподаватель. О. А. Новосельская
(кафедра информатики и веб-дизайна, БГТУ)

РАЗРАБОТКА ЛАТЕНТНЫХ УЗОРОВ СРЕДСТВАМИ РАСТРОВОЙ И ВЕКТОРНОЙ ГРАФИКИ

Латентное изображение – невидимый глазом результат фотофизических и фотохимических процессов, произошедших в фотоматериале при его экспонировании [1]. Преобразуется видимое изображение в процессе проявления. Также используется термин визуализация (особенно для процессов с переносом вещества). Формирование латентных изображений основано на поглощении определенной доли спектра слоем вещества. Такое поглощение может осуществляться как для видимых областей спектра, так и для ультрафиолетового и инфракрасного излучений. В качестве теоретических основ формирования латентных изображений выступает наличие основных и дополнительных цветов в видимой части спектра. Два цвета являются дополнительными друг другу, если их пигменты, будучи смешанными, дают нейтральный серо-черный цвет либо в случае действия излучений – белый. Расположенные рядом дополнительные цвета максимально возбуждают друг друга и взаимоуничтожаются при смешивании, образуя серо-черный тон, как огонь и вода. Каждый цвет имеет лишь один-единственный цвет, который является по отношению к нему дополнительным. В цветовом

круге дополнительные цвета расположены диаметрально один другому. Они образуют следующие пары дополнительных цветов:

- желтый – сине-фиолетовый;
- оранжевый – фиолетовый;
- красный – голубой;
- пурпурный – зеленый и т. д. (рисунок 1).

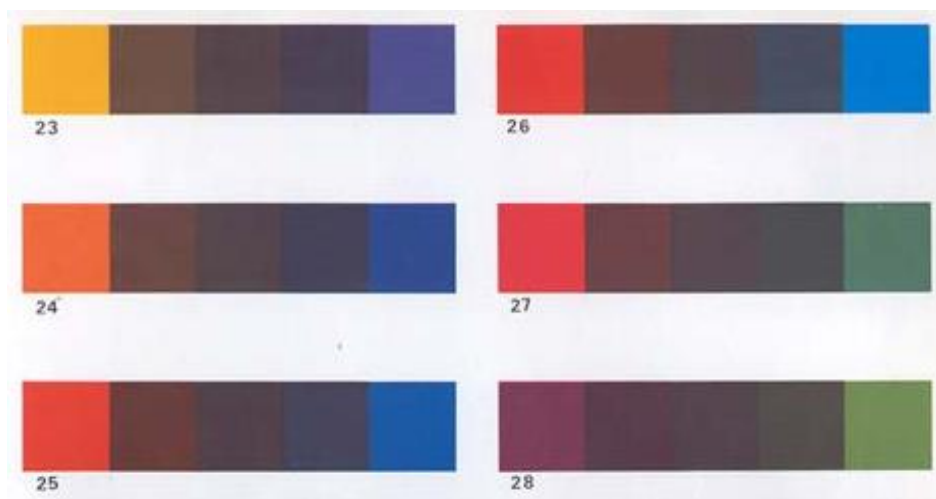
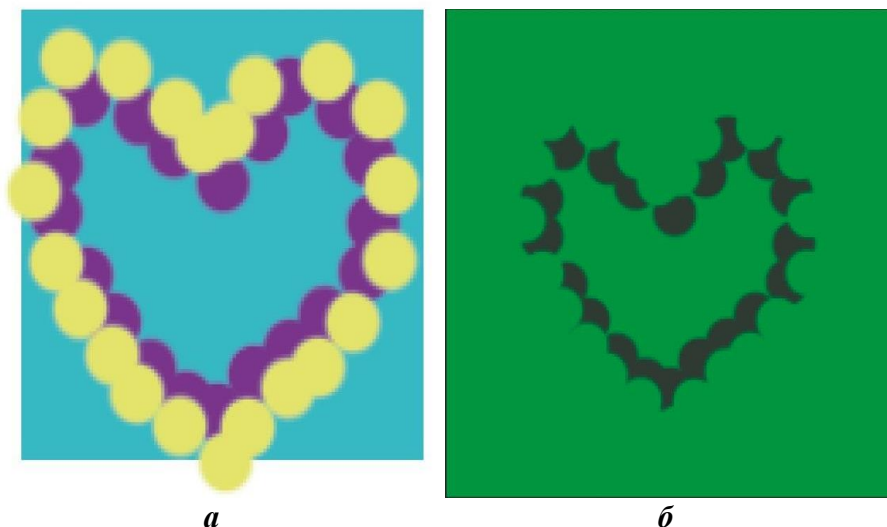


Рисунок 1 – Примеры дополнительных цветов и результат их смешения

По аналогии с выделением цветов системы CMYK управление слоями красок осуществляется в каналах красного (дополнительного голубой краске – Cyan), зеленого (дополнительного пурпурной краске – Magenta), синего (дополнительного желтой краске – Yellow) и нейтрально серого излучений (для выделения черного цвета – Key color, black). Конструируется изображение за счет смешения соответствующих красок, в отличие от RGB, где осуществляется смешение излучений. Поэтому, чтобы создать латентное изображение достаточно задать для выделяемого цвета необходимый узор, а затем подобрать близкие цвета действующему дополнительному излучению. Так, для выделения пурпурного цвета необходимо задать базовое изображение в виде узора из красок пурпурного цвета. Наложить на него краски, содержащие зеленый и соседние к нему цвета, т. е. голубой и желтый и изменить режим наложения в зеленом слое на Multiply (Умножение). Режим Multiply имитирует наложения красок друг на друга, отсекая светлые пиксели, оставляя темные [2]. И именно так выделяются краски. Чем больше краски тем темнее оттенок. И как итог получается точно такое же изображение, сделанное своими руками (рисунок 2 а, б).



**Рисунок 2 – Результат разработки латентного изображения:
а – основной цветной узор; б – скрытый узор**

Основой латентности получаемых изображений составляет метамерность человеческого зрения. Так в определенных условиях освещения мы можем увидеть абсолютно одинаковые цвета, в то время как при освещении в других спектрах они окажутся различными. На метамерности человеческого цветовосприятия основаны способы подбора красок. Например, при освещении поверхности голубого и синего цвета синим излучением обе поверхности окажутся зрительно неразличимыми для человеческого глаза. А при освещении этих же поверхностей белым светом, мы почувствуем разницу в оттенках. Примером такого воздействия является рисунок 3 а, в котором при нормальных условиях освещения создан текстовый слой с желтым кругом на фоне. Если Экранировать такое изображение синим излучением мы увидим вместо текста основное изображение в виде круга, на фоне которого имеется текст (рисунок 3 б). Т. е. происходит зрительная инверсия получаемого изображения. За основу взято бинарное сочетание двух дополнительных цветов, что и приводит к такому эффекту.



Рисунок 3 – Разработка инверсного изображения:
а – основное изображение; б – результирующее изображение

Основной областью применения разработанных латентных изображений является дизайн и защита графической информации от копирования, например в логотипах и других элементах фирменного стиля. Можно использовать подобные узоры в качестве частного элемента фирменного стиля.

ЛИТЕРАТУРА

1. [Иттен, Иоханнес. Искусство цвета / И. Иттен: пер. с немецкого; 2-е издание – М.: Изд. Д. Аронов, 2001. – С. 78.](#)
2. Каналы в Фотошопе Ч.2. Блог Дмитрия Веровски Hronofag [Электронный ресурс]. – Режим доступа: <http://hronofag.ru/2012/08/photoshop-channels-смук>. – Дата доступа: 20.04.2017.

УДК 004.738.1-025.13

Студ. А. А. Ковалевская

Науч. рук. ассист. Т. П. Брусенцова

(кафедра информатики и веб-дизайна, БГТУ)

РАЗРАБОТКА ДИЗАЙНА ЛОНГРИДА ИНФОРМАЦИОННОГО САЙТА

Ежедневно в сети появляются тысячи новых терминов и понятий, в том числе и тех, которые касаются текстов. И каждый современный, уважающий себя копирайтер просто обязан быть в курсе всех новомодных течений и тенденций. Тем более, если они

набирают популярность практически со скоростью света. Одним из таких понятий является «лонгрид». Многие употребляют этот термин, но все ли хорошо понимают его значение?

Лонгрид (от англ. “long read” – длинное чтение) – это принципиально новый способ подачи объемного текста в Интернете. Как известно, длинный текст в сети не приветствуется, ведь зачастую в Интернет люди приходят получить информацию быстро и в максимально сжатые сроки. Но лонгрид – дело совершенно другое. Это не сплошной массив текста, а целый комплекс, состоящий из текста, разбавленного видеороликами, инфографикой, презентацией, фоновой музыкой, иллюстрациями (в том числе интерактивными), сносками, встроенными цитатами и т. д. Самое важное, что все они взаимосвязаны между собой – они не просто дополняют текст, а являются его равноправными партнерами. В таком материале крайне важна последовательность изложения, гармоничное сочетание всех его частей – размер и количество иллюстративного ряда, шрифтов, цветов и другое. Лонгрид должен восприниматься как целостная история. Иными словами, человек, который открыл ваш материал, должен полностью в него погрузиться, прочувствовать эту историю и прочувствовать так, чтобы его ничто не отвлекало.

Второй отличительной особенностью лонгридов, кроме разнообразия в структуре информации, является подача материала с необычной точки зрения или же глубокая проработка вопроса – другими словами, самая настоящая аналитика. Главная цель автора лонгрида – увлечь читателя, рассказав ему необычную и увлекательнейшую историю либо же подав «заезженную», но актуальную тему с совершенно новой точки зрения.

В качестве примера рассмотрим разработанный мною лонгрид, темой которого является «Что такое лонгрид». Он представлен в виде лендинга, т.е. одностраничного сайта (рисунок 1). Весь текст расположен в одном блоке, находящемся в разделе «Статья».



Рисунок 1 - Макет сайта

В статье есть 4 основных раздела:

1. *Что такое лонгрид.* Здесь приведено определение лонгрида и примеры сайтов такого типа.

2. *Типы лонгридов.* В этом пункте описываются типы лонгридов.

3. *Мультимедийные лонгриды.* Рассказывается о том, что есть мультимедийный лонгрид, в чем в связке используется и примеры новостных порталов с таким типом.

4. *Выводы.*

Кроме этого на макете имеются картинки и ссылки на профили в социальные сети (Facebook, Instagram, Twitter, Вконтакте).

Подведём итоги. Сказать однозначно, что за лонгридами будущее, нельзя. Ведь всё зависит от того, научимся ли мы разрабатывать качественные проекты, не сведём ли хорошую идею и кропотливую работу к подсчёту знаков, будем ли ориентироваться на читателя и понимать, какие вопросы он задаёт и какие проблемы его волнуют прямо сейчас.

Пока лонгриды ещё не так популярны в бизнес-среде, есть возможность освоить разработку и коммерческих, и мультимедийных лонгридов. Если вы пока не определились, куда направить свой профессиональный рост, перед вами один из самых конкурентоспособных вариантов.

ЛИТЕРАТУРА

1. Что такое лонгрид «Ваш редактор» [Электронный ресурс] — Режим доступа: <http://vashredaktor.com/chto-takoe-longrid/> Дата доступа: 12.06.2015

2. Лонгрид: причины популярности длинных текстов «Madcats» [Электронный ресурс] — Режим доступа: <http://madcats.ru/content-marketing/longreads-how-to/> Дата доступа: 06.12.2016

УДК 004.738.51070

Студ. А.С. Зябко

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

ВИЗУАЛЬНОЕ ПРЕДСТАВЛЕНИЕ МУЛЬТИМЕДИЙНОГО РЕПОРТАЖА В НОВОСТНЫХ ИНТЕРНЕТ - СМИ

Мы живем в уникальное время конкуренции среди средств массовой коммуникации за внимание аудитории. На наших глазах

печатные версии газет, традиционное радио, массовое эфирное телевидение сдают позиции перед Интернетом как каналом коммуникации. Скорость, мобильность, универсальность, интерактивность, мультимедийность – вот ключевые слова современного СМИ, современной редакции и современного медиапространства. Задачей СМИ становится не просто собрать, получить и изложить информацию, а с помощью новых технических возможностей предложить контекст и объяснить людям новости.

К традиционным СМИ и традиционным видам коммуникации добавились новые медиа и новые виды коммуникации — блоги, социальные сети, мобильные телефоны, планшеты и т.д. В обществе происходит процесс афiliationи — участие в онлайн-обществах и социальных сетях, таких как Facebook, MySpace, OK.ru, Vk и др. Теперь потребитель может получать новости в любой ситуации: на ходу, в очереди, в личном и общественном транспорте, на работе, во время приема пищи, во время занятий спортом — лишь бы под рукой был гаджет, позволяющий получать новости.

Чтобы выдерживать конкуренцию все привычные печатные издания, например, такие как: Daily Telegraph, РИА «Новости», BBC – уходят в Интернет.

Почему же печатные издания уходят в прошлое? Причин много, но одна из них, это ограниченность вариантов подачи информации. Благодаря высоким технологиям стало возможным более привлекательно подать информацию. Красочность материала, возможность поместить видео сюжеты, различные мультимедийные «плюшки», вольность в написании – все это делает интернет СМИ более привлекательными.

Исследователи выделяют следующие жанры новых медиа: текстовые, мультимедийные и синтетические.

К текстовым относятся как традиционные жанры (новость, очерк, колонка), так и новые жанры – текстовая трансляция, статус в соцсетях.

Синтетический жанр подразумевает под собой объединение нескольких элементов – текста, иллюстрации, звука, интерактива. Данный жанр широко распространен на сегодняшний момент. Так как он позволяет более широко преподнести информацию.

Среди мультимедийных жанров выделяют иллюстративные жанры, аудиальные жанры, видеожанры.

К иллюстративным жанрам относятся:

- Статичная иллюстрация (отражает происходящее, дополняет информацию, формирует визуальный ряд). Обычное фото в статье.

- Фотолента, фоторепортаж.

- Слайд-шоу (меняющаяся картинка) – широко применяется в современных интернет СМИ

- Инфографика (воссоздание полной картины события, несколько уровней материалов, объединение жанров, 3D).

Аудиальные жанры:

- Подкаст (отдельный файл или серия регулярно обновляемых аудиофайлов. По длительности выделяют: короткая новость, сюжет или передача). Самая распространенная платформа iTunes.

- Аудиоиллюстрация (фрагмент аудиозаписи, относящийся к определенной части текста; добавляет эмоциональную или смысловую нагрузку). Часто используется в образовательных порталах.

- Аудиосюжет (осмысленная и срежисированная аудиозапись небольшого формата).

Видеожанры:

- Видеоиллюстрация (нет законченного сюжета).

- Видеосюжет (законченное видеосообщение: завязка, кульминация, развязка). Конетент на Ютуб.

- Потокое видео (перископ, твич).

- Интерактивный видеомост (интервью с ньюсмейкером в прямом эфире)

Синтетические жанры:

- Слайд-шоу со звуком (аудиослайд-шоу).

- Мультимедийное ток-шоу (интерактивность, аудитория пользователей, мультиэкранность, twitch, перескоп).

- Информационные игры (интерактивный инфотеймент).

- Интерактивный рисунок.

- Интерактивный сюжет (в тело сюжета встроены ссылки на другие материалы – ссылки, иконки, микробаннеры).

Все эти жанры активно используются в интернет СМИ, фото, собранные в один слайд-шоу дадут возможность более детально увидеть, то о чем идет речь в статье, видеоконтент к тексту статьи позволяет «увидеть все своими глазами».

Также приятным бонусом является возможность встроить в статью интерактивные элементы, такие как тесты, игры, карты и т.д. К примеру, в новости рассказывается о типах характера человека, и в

конец статьи можно встроить тест, пройдя который читатель сможет узнать о своем типе характера.

Но не только привычные печатные издания ощущают конкуренцию Интернета, также и телевидение теряет долю рынка.

Например, медиа площадка YouTube большими темпами охватывает аудиторию телевидения. И в скором будущем сможет обойти по количеству зрителей ТВ.

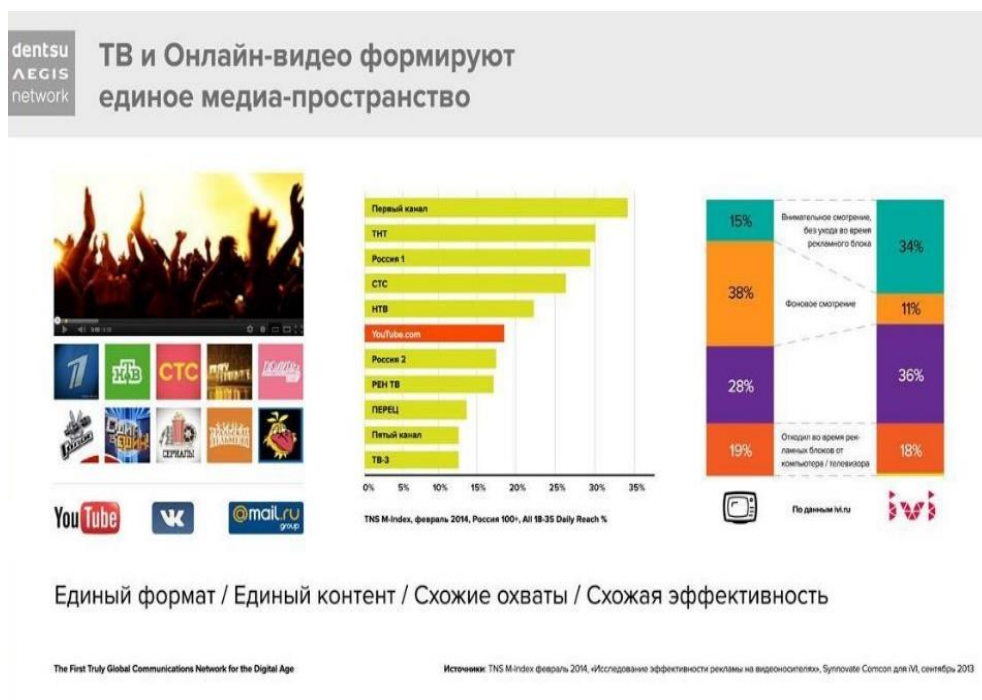


Рисунок 2

Огромный выбор контента на любой вкус, возможность смотреть то, что не покажут по ТВ, формировать свой список каналов, практически полное отсутствие рекламы – все это дает возможность YouTube обойти телевидение.

УДК 004.582

Студ. Е.В. Буряя

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

РАЗРАБОТКА НАВИГАЦИИ САЙТА «GAMESTOP»

Не существует единого формата разработки удачной навигации на веб-сайте. Тем не менее, можно выделить некоторые общепринятые правила, помогающие в разработке навигации, а также

отметить наиболее частые ошибки, допускаемые в процессе ее разработки. Навигация сайта является удачной, если соответствует следующим критериям:

1) Понятность. Навигация должна быть понятна любому посетителю сайта с первого взгляда. В противном случае он может запутаться во множестве ссылок и в конечном итоге покинуть сайт.

2) Узнаваемость. Каждый посетитель сайта должен без особых проблем отличить навигацию от прочих гипертекстовых ссылок.

3) Единообразие. Гипертекстовые навигационные ссылки должны быть одинаковыми на всех страницах сайта.

4) Легкодоступность. Не следует прятать навигацию в глубине сайта. Ознакомившись с какой-либо информацией, посетитель, должен без труда продолжить пользование веб-ресурсом.

5) Удобство. Удобство использования навигации предполагает крупный текст ссылок, легкочитаемость текста, ярко выраженные названия подразделов и внутренние ссылки для длинных страниц.

При создании навигации наибольшее внимание следует обратить на то, насколько она удобна с точки зрения юзабилити.

Среди основных ошибок при создании навигации можно выделить следующие:

1) Наличие тупиков. Если, завершив знакомство с информацией на странице, пользователь не видит, что еще он может сделать на сайте, он теряет интерес к ресурсу и не видит всех его возможностей.

2) Присутствие так называемых «ромбов». Так называют ситуацию, когда две разные ссылки на странице ведут на одну и ту же страницу. Данную ситуацию лучше избегать, т.к. одна из ссылок получается лишней. Каждая ссылка должна вести на свой раздел.

3) Нет возврата вверх. Это особенно актуально для длинностраничных сайтов, а также для сайтов с большим количеством информации. Если нет желания делать кнопку «вернуться наверх», данную проблему можно решить, сделав меню фиксированным.

4) Нагроможденность ссылками. Когда между ссылками нет свободного пространства, пользователю тяжело разбираться в контенте сайта и выделять необходимую ему информацию. Между ссылками и кнопками всегда должно оставаться пустое пространство.

5) Отсутствие индикаторов местоположения. Если на сайте присутствуют длинные цепочки ссылок, необходимо использование так называемых «хлебных крошек», чтобы пользователь видел каждый свой шаг.

б) Длинные меню и списки. Листать огромное количество ссылок неудобно хотя бы потому, что анализировать информацию лучше «небольшими порциями». Если сделать кнопку для постепенной подгрузки дополнительных пунктов, пользователь куда быстрее найдет необходимый ему пункт.

7) Нет стрелок на кнопках и ссылках. Использование стрелок не всегда необходимо и уместно, но значительно облегчает навигацию при наличии цепочных ссылок. Так пользователь будет строить ассоциации с тем, куда он движется по ссылке: вперед или назад.

Сайт GameStop является новостным ресурсом на игровую тематику. При разработке навигации я учитывала те принципы, которые в наибольшей степени актуальны для новостных ресурсов, а также общие правила.

Структура навигации сайта – решетчатая. Это означает, что каждая страница связана как минимум с одной страницей каждого уровня (рисунок 1).

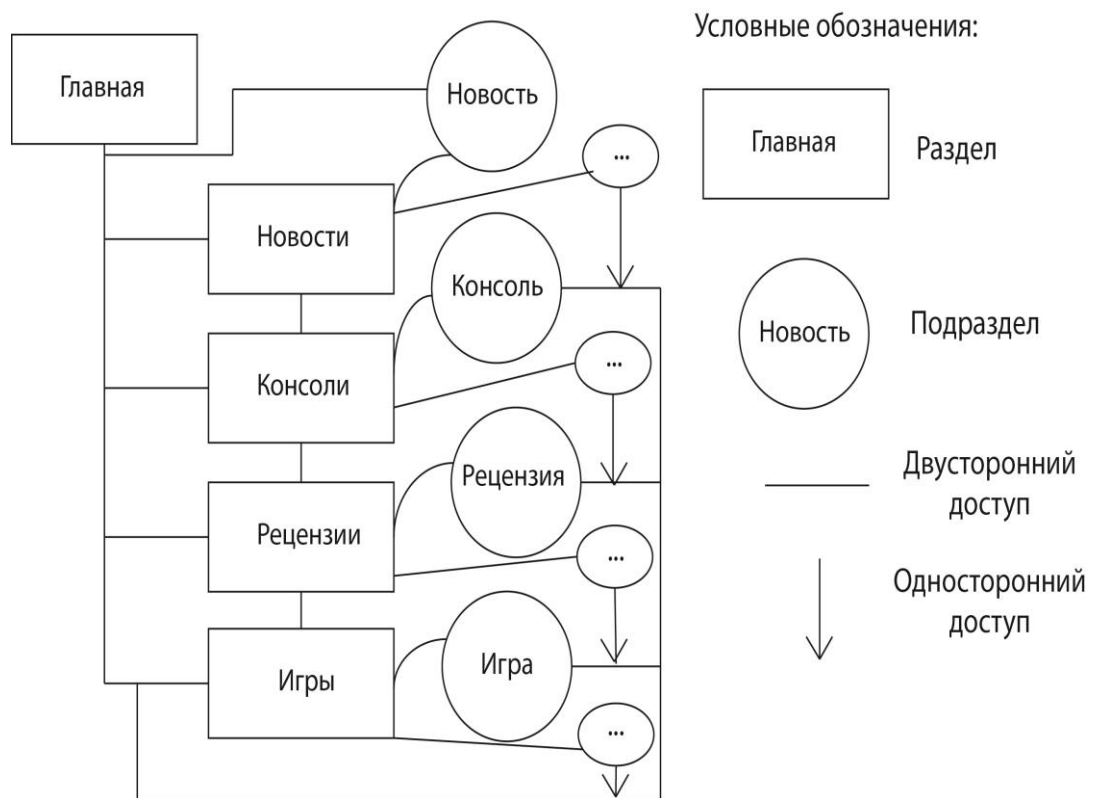


Рисунок 1 — Структура навигации веб-сайта GameStop

На всех страницах сайта меню фиксировано в хэдере. Так пользователь в любой момент может перейти на интересующий его раздел. Но на главной странице также присутствует кнопка

«Вернуться наверх». Данная кнопка сочетается с фиксированным меню именно на главной странице, т.к. информация на ней неоднородна и вероятность того, что пользователю нужно будет вернуться к началу страницы, велика. На других страницах подобная ситуация отсутствует, поэтому фиксированного меню и прокрутки страницы вполне достаточно.

Меню раскрывается в виде списка. Несмотря на то, что насайте, помимо главной страницы, есть всего четыре раздела, я было принято решение не размещать ссылки на разделы в хэдере друг за другом. Причин на это несколько: во-первых, в хэдере присутствуют логотип и строка поиска, и пункты меню загромождали бы пространство. Во-вторых, в раскрывающееся меню всегда можно добавить еще один пункт. Это удобно на тот случай, если в будущем на сайте появится еще один раздел.

Во всех превью статей присутствует ссылка «Читать дальше». К ней добавлено изображение стрелки, чтобы пользователь сразу же замечал эту ссылку и получал мгновенную ассоциацию, что она осуществляет переход вперед.

На страницах подразделов присутствуют «хлебные крошки». Несмотря на то, что цепочка переходов недлинная, возможность вернуться на предыдущую страницу и визуализация пути пользователя на сайте – это плюс для юзабилити.

На странице с новостными и другими блоками присутствует кнопка, загружающая следующие блоки. Таким образом пользователь можем сам контролировать, насколько длинной будет страница, и будет получать информацию «небольшими порциями».

Между всеми элементами сайта, включая навигационные, есть свободное пространство. В связи с тем, что на новостных сайтах всегда присутствует принцип «Информация превыше дизайна», количество информации на странице, как правило, большое. Поэтому особенно важно наличие свободного пространства, которое улучшит визуальное восприятие сайта и позволит пользователю лучше ориентироваться на сайте.

Таким образом, учитывая правила создания навигации и избегая наиболее распространенных ошибок, возможно создать наиболее приемлемую навигацию, удобную для пользователей.

УДК 621.357.74:628.54

Студ. В.С. Бекиш

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

ПРИНЦИПЫ РАЗРАБОТКИ ДИЗАЙНА ИНТЕРФЕЙСОВ МОБИЛЬНОГО УСТРОЙСТВА

Успех продукта (мобильного приложения) определяется совокупностью факторов, самый важный из которых — пользовательский опыт в целом. Однако продумывая общую концепцию, мы часто забываем о тех мелочах в дизайне, которые кажутся приятным, но не критичным бонусом. А ведь именно то, насколько тщательно мы продумали эти детали, может оказаться решающим для пользовательского опыта.

О визуальном отклике часто забывают, рассматривая общую картину дизайна, но именно он связывает все элементы воедино. Без него нормальное взаимодействие с пользователем невозможно. Представьте, что вы обращаетесь к человеку, а он вообще никак не реагирует – очевидно, что разговор не сложится. Так и с приложениями. Вы обязательно должны обеспечить пользователю какую-то реакцию на каждое действие – так он будет чувствовать, что контролирует ситуацию. Визуальный отклик:

- дает понять, что действие пользователя совершилось;
- сообщает результат этого взаимодействия в наглядной и понятной форме; подает пользователю сигнал, что он (или само приложение) выполнил действие успешно или неудачно.

Чтобы этого добиться, сделайте кнопки и прочие элементы управления осязаемыми. В физической реальности кнопки, панели управления и прочие предметы выдают какую-то реакцию, когда мы с ними взаимодействуем, и мы воспринимаем это как нормальное положение дел. Такой же отзывчивости пользователи ждут и от элементов дизайна приложений.

Загрузка анимации отображает состояние процесса в реальном времени, давая пользователю возможность быстро сориентироваться, что происходит.

Стивен Хубер в своем исследовании использования мобильных устройств, определил, что 49% людей для решения задач полагаются только на большой палец (рисунок 1). На рисунке ниже, диаграмма, показанная на экране мобильного телефона, является

ориентировочной схемой охвата, в которой цвета указывают, каких областей пользователь может достичь большим пальцем, чтобы взаимодействовать с экраном.

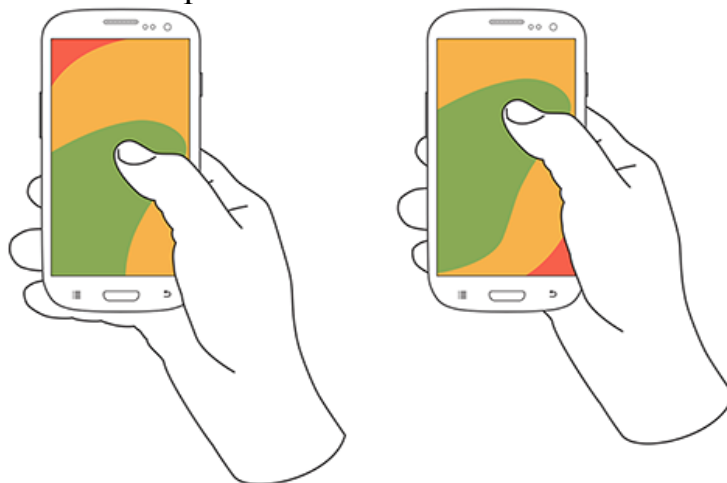


Рисунок 1 – Принцип большого пальца

Зеленый цвет указывает на область, которой пользователь может легко достичь; желтый – область, которая требует усилия; и красный – область, требующая от пользователя изменения способа удерживания телефона. Положение рук и сцепление должны влиять на размещение элементов управления в мобильном дизайне:

- Очень важно поместить меню верхнего уровня, часто используемые элементы управления и общие действия в зеленой зоне экрана, что бы до них было проще дотянуться большим пальцем.(картинка пример меню)
- Поместите негативные действия, такие, как удаление, в труднодоступную, красную зону, потому что вы не хотите, чтобы пользователь случайно на них нажал.

Элементы интерфейса должны быть ясно видны.

Обеспечьте достаточный цветовой контраст между элементами, чтобы пользователи со слабым зрением могли видеть и использовать ваше приложение без проблем.

Убедитесь, что цвет шрифта и фона достаточно контрастируют – это упущение может привести к недостаточной разборчивости. Рекомендуются следующие показатели контрастности для основного текста и текста изображения:

- Небольшой текст должен иметь коэффициент контрастности не менее 4,5: 1 к его фону.

- Большой текст (14 пт жирным шрифтом/18 пт обычным, и выше) должен иметь коэффициент контрастности не менее 3:1 к его фону
- Иконки, и другие важные элементы также должны использовать вышеуказанные рекомендуемые показатели контрастности (рисунок 2).

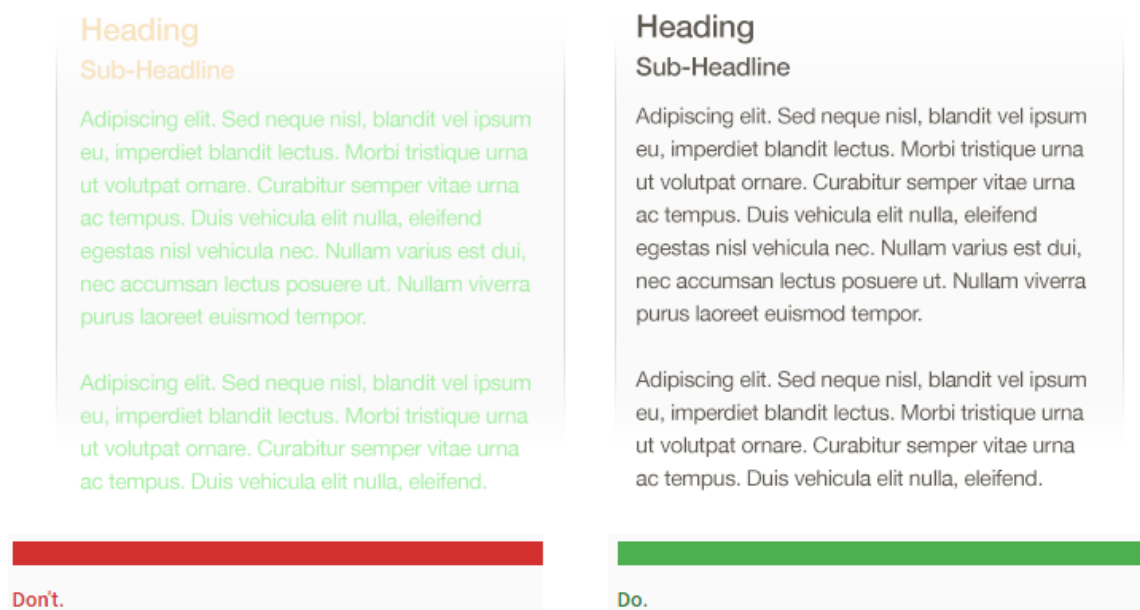


Рисунок 2 – Пример контрастности

Учитываем размер пальцев пользователя.

Да, пальцы многих пользователей гораздо больше, чем могли бы представить любители утонченного дизайна. Поэтому обязательно нужно адаптировать свою программу к пальцам разного размера.

Кнопкам просто нужно оставлять достаточно места. Если ваши кнопки слишком малы или расположены слишком близко друг к другу, некоторые люди просто не смогут попадать по ним. Как следствие, пользователи будут раздражаться и, может быть, прекращать работу с такой программой. Наши пальцы действительно большие. Их ширина составляет около 45–57 пикселей, что больше, чем рекомендует большинство руководств для тестовых устройств. Apple, например, рекомендует цель квадратной формы с размером стороны в 44 пикселя. А этого далеко не всегда достаточно.

Свободное пространство.

Свободное, или пустое пространство – это те области дизайна, где не размещены никакие элементы. Оно складывается из пространства вокруг картинок, полей, пробелов между буквами и строками текста. Хотя многие считают его бесцельной тратой ценного места на экране, на самом деле оно само по себе является важным элементом интерфейсом.

Сила пустого пространства происходит из того, что человеческие память и внимание не безграничны. Наша краткосрочная память способна удерживать очень небольшое количество актуализированной, готовой к употреблению информации (около 7 элементов или даже меньше) на достаточно короткий срок (10-15 секунд).

Большое количество пустого пространства может придать простой и гостеприимный вид даже самому неряшливому дизайну. Чем больше свободного места, тем меньше элементов и, соответственно, тем проще для пользователя бегло их просмотреть. Искусство применения пустого пространства состоит в том, чтобы предоставить пользователям столько информации (в блоках), сколько они смогут обработать, а затем убрать все лишние детали (рисунок 3).

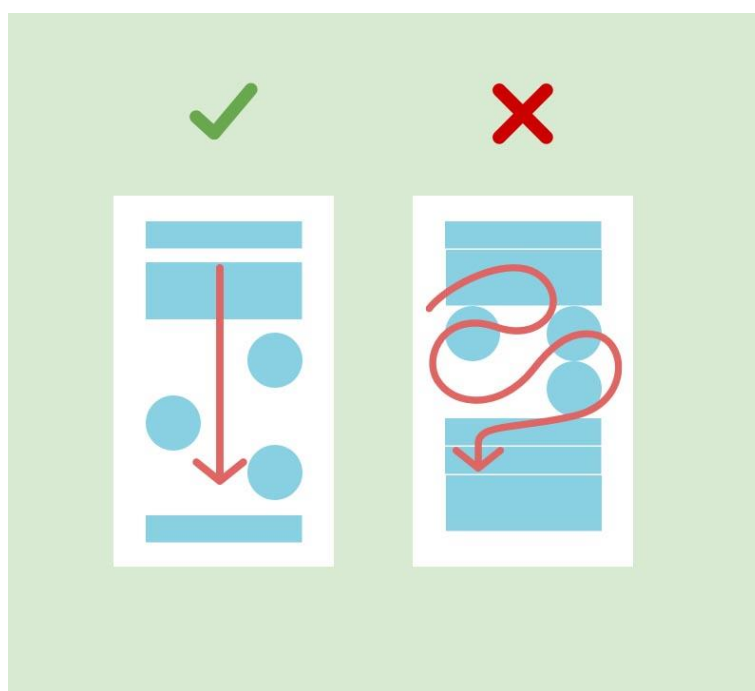


Рисунок 3 – Применения пустого пространства

Проясняем связи.

Закон близости объясняет, как наши глаза устанавливают связи между визуальными элементами. Он утверждает, что объекты, расположенные близко друг к другу, кажутся нам похожими. Мы можем использовать свободное пространство, чтобы выстраивать связи между элементами.

Когда мы разбиваем информацию на тематические группы, её становится проще пробежать глазами и воспринять.

Заключение.

«Детали — это не просто детали. Из них и состоит дизайн». — Charles Eames. Дизайн требует бережного отношения. Каждая мелочь в интерфейсе вашего приложения заслуживает пристального внимания, потому что пользовательский опыт складывается из гармоничного сочетания всех деталей.

УДК 004.738.5

Студ. Ю.С. Кулешова

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

ЖУРНАЛЬНЫЙ СТИЛЬ В ВЕБ-ДИЗАЙНЕ

На сегодняшний день, журнальный стиль веб-ресурсов является наиболее популярным и используемым.

Журнальный, или как его еще называют, газетный стиль имитирует печатный СМИ в Интернет – сети. Он имеет такие же качественные фото, вместительные заголовки, колончатое размещение текста. Для заголовков применяются разнообразные цвета, стили и шрифты. Фотографии публикуются только качественные и профессиональные. Каждая страница в обязательном порядке вмещает ссылки на другие рубрики.

В журнальном стиле главный акцент делается на текст. Даже изображение публикуются для того, что бы передать смысл текста, а не украшения или общего фона. В основном таким дизайном интересуются журналы и газеты, а так же электронные СМИ.

Журнальный стиль успешно зарекомендовал себя не только на подобных ресурсах, но и на сайтах другой направленности, в том числе и коммерческих. Этот стиль очень эффективно подойдет для создания портфолио, для галереи работ дизайнеров, фотографов и пр.

Также весьма популярен он и в создании Интернет магазинов. Одним словом, журнальный стиль, применим на самых разных ресурсах.

Но в первую очередь всегда нужно помнить, что для пользователя важнее всего удобство сайта, его простота и доступность. Позаботьтесь, что бы пребывание посетителя на сайте, было максимально комфортным и безопасным.

Особенности журнального веб стиля:

- Первое, что видим – это табличная верстка, вертикальность расположенной информации, часто в две или три колонки. Каждая колонка разделена либо полосами, гармонирующими со всем стилем сайта, либо пробелами. Присутствуют отступы, иначе поля слева и справа, как принято в тетрадках и любых книгах.
- Второе, одна страница может содержать совершенно по-разному оформленные блоки информации. Это может быть обыкновенный текст в колонке, текст в цветной рамке с другим фоном и другим стилем в заголовке, поясняющий текст на самих изображениях. Макетирование журнала определяет с помощью сетки расположение картинок и заголовков.
- Третье, заголовки выделяются полужирным начертанием, разными стилями шрифта и многообразием цвета.
- Четвертое, шрифты, используемые в тексте без засечек, размер средний.
- Пятое, очень много хороших высококачественных фотографий, которые являются яркими пятнами на странице веб-журнала и затмевают по цветовому решению другие графические элементы сайта.
- Шестое, присутствуют фоновые иллюстрации, по цвету гармонирующие с основным цветом веб-страницы журнала.
- Седьмое, навигация по максимуму представляет все подразделы сайта без повтора в футере, ссылки на темы рубрик есть на каждой странице сайта.
- Восьмое, журнальный стиль наполнен яркими красками, радостью, неким праздничным настроением.

- Девятое, попадая на сайт, выполненный в журнальном веб-стиле, чувствуешь привычное ощущение отдыха от просмотра бумажного журнала. Значит, своим посетителям мы передаем позитив, а не это ли цель создания сайта для людей?

Для данной темы доклада мной были разработаны макеты страниц сайтов на тему: интернет-магазин и блог. При разработке каждого макета учитывались все особенности журнального стиля.

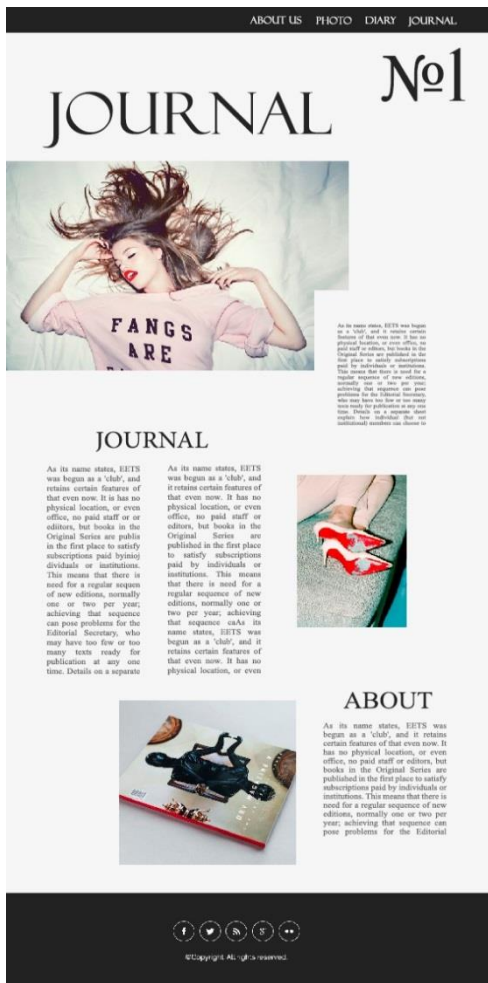


Рисунок 1 – Макет первого сайта

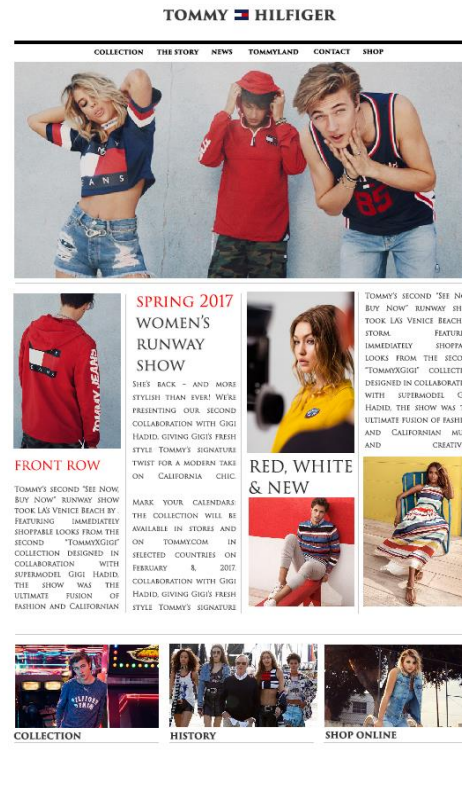


Рисунок 2 – Макет второго сайта

Таким образом, практика показывает, что журнальный стиль в веб-дизайне способен иметь успех и при создании прочих сайтов, в том числе тех, что относятся к коммерческим. Тут речь идет о сайтах-портфолио и сайтах, основное направление которых является онлайн-торговля. Именно поэтому этот стиль получил широкое применение и способен использоваться под самые разнообразные сайты.

УДК 004.925

студ. Д.И. Лосич

Науч. рук. Асс. Т.П. Брусенцова

(кафедра информационных систем и технологий, БГТУ)

ХУДОЖЕСТВЕННЫЕ СРЕДСТВА КОМПОЗИЦИИ – ПРОПОРЦИИ И ЗОЛОТОЕ СЕЧЕНИЕ, ИХ ПРИМЕНЕНИЕ В ДИЗАЙНЕ ИНТЕРФЕЙСОВ

«Математика прекрасна». Это может показаться абсурдным для людей, которые при одном только упоминании математики вздрагивают. Однако некоторые из самых красивых вещей в природе и нашей Вселенной — это проецирование математических свойств. Один из древних философов, Аристотель, сказал: «математическим наукам свойственно выстраивать все по порядку, в симметрии и ограничениях. Они являются главными формами прекрасного». Из-за своей природной красоты математика является частью искусства и архитектуры. Но она практически не применяется для дизайна сайтов и приложений. Это наблюдается из-за того, что многие не сопоставляют математику с дизайном. Хотя, наоборот, математика может быть *инструментом* для производства поистине волшебных конструкций.

ЗОЛОТОЕ СЕЧЕНИЕ «И» ЗОЛОТОЙ ПРЯМОУГОЛЬНИК.

Число 1.618 должно быть известно всем серьезным дизайнерам. Это число золотого сечения, которое встречается в природе, изобразительном искусстве, архитектуре. Морские ракушки, Мона Лиза и Парфенон демонстрируют использование золотого сечения. Наши лица и тела тоже пропорциональны золотому сечению. Это вездесущее сечение можно найти даже в звуках и временных интервалах. Если бы существовала математическая формула, раскрывающая красоту природы, то это было бы золотое сечение.

Золотое сечение, также известное как божественная пропорция, являет собой иррациональную математическую константу со значением примерно 1,618. Это означает, что если мы возьмем два числа и разделим одно на другое, то получим 1,618

Создание Золотого прямоугольника довольно просто, и начинается с основного квадрата. Основной квадрат позволяет сформировать длину «короткой стороны» прямоугольника. Этот квадрат разделяется пополам вертикальной линией, дающей два

прямоугольника. В одном прямоугольнике проводится линия из одного угла в противоположный. Эта линия переворачивается так, чтобы она легла горизонтально рядом с первым прямоугольником. С помощью новой горизонтальной линии достраивается прямоугольник.

Нормальным будет округлить его в большую или в меньшую сторону на точку или две, чтобы сделать эти числа соответствующими пикселям или точкам. Так что если есть область контента размером 640px, то боковая панель в 400px будет достаточно хорошо соответствовать Золотому соотношению, хотя на самом деле это соотношение - 1:1,6.

Простое правило для применения золотого сечения в тексте заключается в том, что последующий больший или меньший текст должен соответствовать числу 1,6.

На примере это выглядит так: если размер шрифта – 11, то подзаголовок должен быть написан в более крупном шрифте. Шрифт текста умножается на число золотого сечения, чтобы получить большее число ($11 \cdot 1,6 = 17$). Значит подзаголовок должен быть написан в 17 размере шрифта. А теперь заголовок или название. Подзаголовок умножается на пропорцию и получается 27 ($1 \cdot 1,6 = 27$).

Можно оставаться верным золотому сечению даже в веб-дизайне, то есть можно эффективно использовать золотое сечение и применить его к сеткам веб-страниц и макетам пользовательского интерфейса.

Возьмите общее число сетки пикселей за ширину или высоту и используйте его для построения золотого прямоугольника. Разделите наибольшую ширину или длину для получения меньших чисел. Возьмем обычную ширину разметки в 960 пикселей и поделим 960 на 1.618. Получим что-то около 594, что и будет являться шириной колонки с основным содержанием. Чтобы получить ширину второй колонки, вычитаем 594 из 960 и получаем 366. Собираем всё это вместе и получаем две колонки, построенные с применением золотого сечения. Золотое сечение применимо и к делению вертикали, если ширина сайта фиксирована. Берем высоту в 760 пикселей и делим на 1.618; получаем около 470 — высота колонки с содержанием сайта. Далее вычитаем 470 из 760 и результат в 290 пикселей и будет «подвалом» разметки. То, что осталось, может быть боковой панелью (или нижней панелью, если вы применили его к высоте). Теперь продолжайте использовать золотой прямоугольник для дальнейшего применения его к окнам, кнопкам, панелям, изображениям и тексту.

В качестве примера рассмотрим блог 404 (рисунок 1). Сам дизайн визуально привлекателен, обеспечивает спокойную и поддерживающую цветовую схему и имеет приятный состав.



Рисунок 1 – Блог 404

ЗОЛОТАЯ СПИРАЛЬ

Также можно использовать золотую спираль, чтобы определить, где разместить контент на вашем сайте. Если домашняя страница загружается с графическим контентом, как, например, на веб-сайте онлайн магазина или блога фотографий, вы можно воспользоваться золотым методом спирали, который используют многие художники в своих работах. Задумка в том, чтобы поместить наиболее ценный контент в центре спирали.

Контент со сгруппированным материалом тоже может быть размещен при помощи золотого прямоугольника. Это означает, что чем ближе спираль движется к центральным квадратам (к одному квадратному блоку), тем “плотнее” там содержимое.

Можно использовать эту технику, чтобы обозначить расположение заголовка, изображений, меню, панели инструментов, окна поиска и других элементов.

Использовать золотое сечения при построении разметки не сложно. А как быть с элементами сайта: полями, панелями, кнопками, картинками и текстом? Сделать это помогут золотые прямоугольники. Длина и ширина золотого прямоугольника пропорциональна золотому сечению. Естественно, не каждый элемент интерфейса будет иметь прямоугольную форму, но те, что будут, можно организовать пропорционально золотому сечению. Для эффективной работы понадобится сетка из золотых

прямоугольников. Она поможет подогнать размеры объектов под размер золотого прямоугольника.

Чтобы создать золотой прямоугольник следует пользоваться спиралью Фибоначчи (рисунок 2). Построение начинается с квадрата со стороной в 1 пиксел. Далее создается еще один и ставится справа от первого. После строится квадрат со стороной 2 пиксела и располагается под предыдущими двумя. Следующий квадрат будет со стороной 3 пиксела, и он будет расположен слева от конструкции. Наконец, создается квадрат со стороной 5 пикселей и размещается сверху. Процесс можно продолжать до бесконечности.

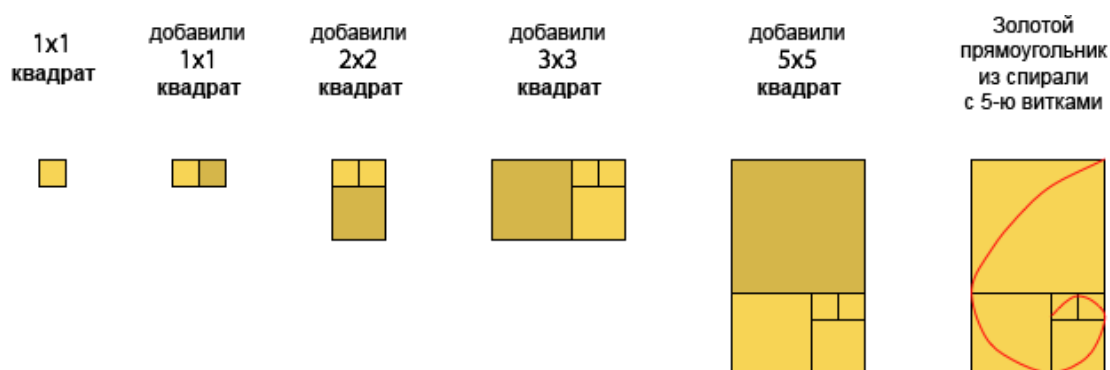


Рисунок 2 – Построение золотого прямоугольника

Теперь, если полученный прямоугольник копировать множество раз по вертикали и горизонтали, получится сетка для построения вертикальных золотых прямоугольников. Для создания горизонтальных прямоугольников и сетку потребуется повернуть горизонтально. Важно помнить, что для правильной работы сетки требуется одинаковое количество прямоугольников и по вертикали, и по горизонтали, в противном случае золотого сечения не получится. Используя золотые сетки удастся создать красивые элементы интерфейса, которые будут соответствовать золотому сечению.

Хоть математика и может быть *инструментом* для производства поистине волшебных конструкций, но тем не менее, не стоит полагаться на математику для каждого своего творения. Дело в том, что математика должна рассматриваться как помощник, а не панацея от всех бед. Конечно, веб-дизайн не должен быть организован в соответствии с Божественной пропорцией. Однако в некоторых случаях это может улучшить не только коммуникацию всего дизайна, но и подачу подробной информации в макетах.

УДК 004.58

Студ. Т. А. Мельникова
Науч. рук. доц. Т. П. Брусенцова
(кафедра информатики и веб-дизайна, БГТУ)

ОРГАНИЗАЦИЯ ОБРАТНОЙ СВЯЗИ С АУДИТОРИЕЙ СЕТЕВОГО ИЗДАНИЯ

В наши дни обратная связь пользователей с разработчиками очень важна. Обратная связь – это не только отправка писем на e-mail, это различные опросы, предложения, формы, тесты.

В данной теме можно разобрать 3 основных вопроса:

- 1) В каких ситуациях уместно спрашивать мнение пользователей о сайте или сервисе;
- 2) Как обратиться к клиентам, чтобы они отреагировали;
- 3) Виды обратной связи.

В каких ситуациях уместно спрашивать мнение пользователей о сайте или сервисе.

Расскажу о собственном опыте в роли клиента. Недавно я совершала покупку в интернет-магазине велосипедов. Я оставила заказ на покупку велосипеда, но не была уверена в нём и попросила в графе «Комментарии к заказу» мне перезвонить.

Мне перезвонили через 5 минут и спросили, что меня интересует по данной модели, я задала кучу вопросов, на которые консультант терпеливо ответил. После окончания разговора мне на почту пришло письмо. В нем я прочитал: «Только что Вы общались с сотрудником центра нашей компании. Мы высоко ценим Ваше мнение и опыт и очень заинтересованы в Вашем отзыве, который будет способствовать повышению качества обслуживания клиентов. Пожалуйста, уделите несколько минут, чтобы ответить на наши простые вопросы...». Если бы со мной как-то не так поговорили или не ответили на мои вопросы, я бы высказал все, что думаю. И менеджеры получили бы возможность немедленно принять меры, чтобы исправить ситуацию и, возможно, удержать клиента.

В данной ситуации было уместно спрашивать мнение пользователя, но также я хочу привести пример, когда мнение клиента можно не учитывать.

Приведу второй пример.

Вы, наверняка, не раз слышали плохие отзывы о продуктах Apple от людей, которые не могут себе позволить приобрести их. Они рассказывают о плохом качестве, неудобстве, высокой стоимости, и о массе других факторов не в пользу Apple. Но ключевым моментом

является то, что они никогда не пользовались этими продуктами. Такая обратная связь для компании не будет полезна.

Как обратиться к клиентам, чтобы они отреагировали

Как мы уже выяснили ранее, не только отправка сообщений на имейл является обратной связью, но и различные опросы, предложения тоже относятся к этому. Поэтому, чтобы посетители вашего сайта хотели с вами общаться добровольно, не нужно навязывать им очередной опрос. Достаточно разместить на сайте небольшую форму, которая содержит в себе рейтинговую систему или 2-3 вопроса о вашем продукте или сайте. Тогда пользователь не будет слишком много тратить времени на это и с радостью ответит на ваши вопросы или поставит честную оценку.

Виды обратной связи

№1 — Форма обратной связи на вашем сайте (Рисунок 1).

A screenshot of a feedback form titled "Обратная связь" (Feedback). The form is contained within a dark grey border with a close button (X) in the top right corner. The text inside the form reads: "Обратная связь", "Оставьте Ваши данные и мы свяжемся с Вами", "Ваше имя", "Контактный телефон или электронная почта", "Отправить заявку", and "Мы не занимаемся рассылкой рекламных сообщений, а так же не передаем контактные данные третьим лицам."

Рисунок 1 – Форма обратной связи

Данный вид обратной связи позволяет клиенту напрямую связаться с продавцом или разработчиком сайта или сервиса. Вы можете таким способом спросить более подробно про продукт или сервис, которым вы пользуетесь или покупаете. Также можете написать таким способом отзыв.

№2 — Раздел FAQ

Раздел FAQ или раздел вопросов-ответов (рисунок 2) на Вашем сайте – очень хороший способ узнать пробелы в своих продуктах.

Если у клиентов очень часто возникают одни и те же вопросы по поводу вашего продукта — значит это повод вынести эти вопросы в отдельный пост, это сэкономит вам очень много времени, а также вашему клиенту.

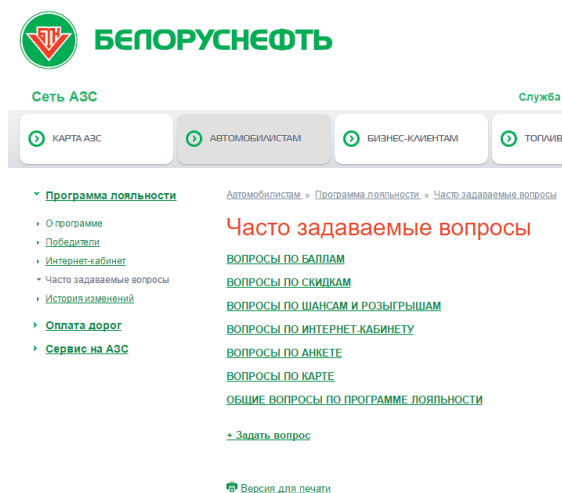


Рисунок 2 – Часто задаваемые вопросы

№3 — Опрос

Опросы и анкетирование помогут вашему сайту стать лучше. С помощью этого вида обратной связи вы можете задавать вопросы напрямую клиенту, который зашёл на ваш сайт или сервис:

1. Что можно улучшить в нашей работе?
2. Как вы оцените наш продукт (сервис)?
3. Откуда вы узнали о нас?
4. Посоветуете ли вы нашу компанию своим знакомым?

Как правило, если это будет оформлено в интересном виде, то пользователь очень охотно ответит на ваш опрос (Рисунок 3). Также можно показать пользователю результаты опроса в отдельном блоке

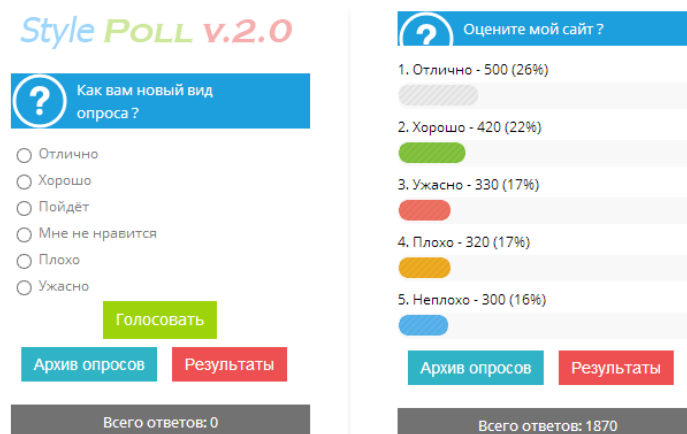


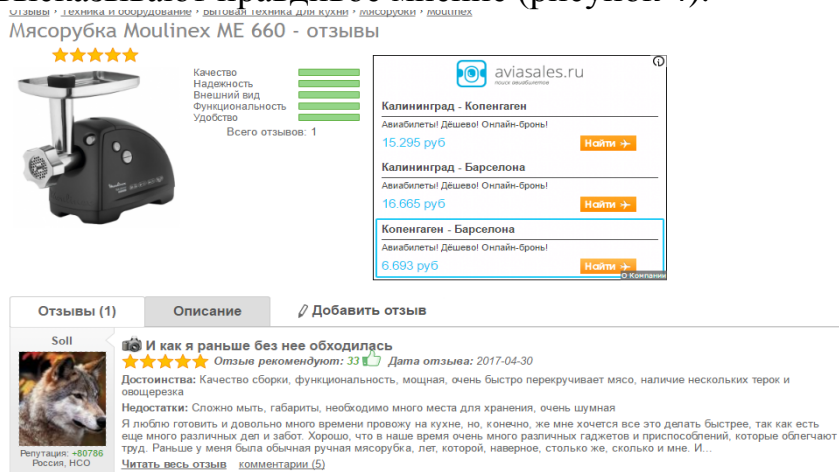
Рисунок 3 – Опрос на сайте и результаты

№4 — Мониторинг интернета и социальных сетей

Многие пользователи не всегда хотят оставлять свои отзывы именно на самом сайте, очень часто они пишут свои отзывы в социальных сетях или на отзывных сайтах, таких как irecommend.ru или otzovik.ru.

Вам необходимо следить за этим, т.к. как правило люди таким способом высказывают правдивое мнение (рисунок 4).

Мясорубка Moulinex ME 660 - отзывы



Качество
Надежность
Внешний вид
Функциональность
Удобство

Всего отзывов: 1

Калининград - Копенгаген
Авиабилеты! Дешево! Онлайн-Бронь!
15.295 руб [Найти](#)

Калининград - Барселона
Авиабилеты! Дешево! Онлайн-Бронь!
16.665 руб [Найти](#)

Копенгаген - Барселона
Авиабилеты! Дешево! Онлайн-Бронь!
6.693 руб [Найти](#)

Отзывы (1) Описание [Добавить отзыв](#)

Soll **И как я раньше без нее обходилась**
★★★★★ *Отзыв рекомендуют: 33* *Дата отзыва: 2017-04-30*
Достоинства: Качество сборки, функциональность, мощная, очень быстро перекручивает мясо, наличие нескольких терок и овощерезка
Недостатки: Сложно мыть, габариты, необходимо много места для хранения, очень шумная
Я люблю готовить и довольно много времени провожу на кухне, но, конечно, же мне хочется все это делать быстрее, так как есть еще много различных дел и забот. Хорошо, что в наше время очень много различных гаджетов и приспособлений, которые облегчают труд. Раньше у меня была обычная ручная мясорубка, лег, которой, наверное, столько же, сколько и мне. И...
[Читать весь отзыв](#) [комментарии \(5\)](#)

Рисунок 4 – Отзыв клиента

№5 — Звонки клиентам

Важно иметь доверительные отношения с клиентом. Очень хорошим видом обратной связи является телефонный звонок (рисунок 5).

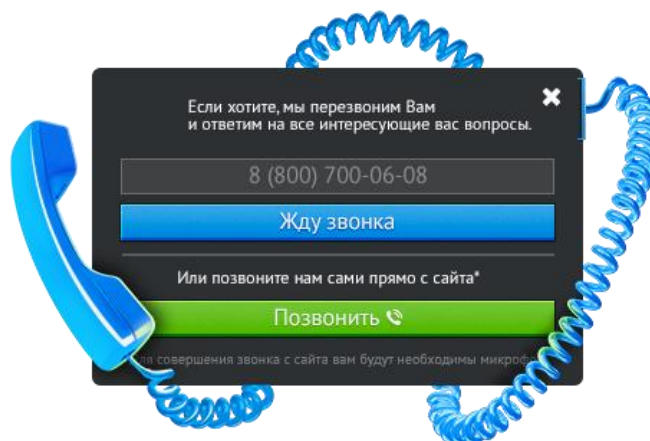


Рисунок 5 – Обратный звонок

Как только вы звоните клиенту, он чувствует, что вы ему не безразличны и начинает вам доверять. Также можно разместить на своём сайте услугу «Обратный звонок», чтобы клиент сам не искал ваш номер телефона.

Правильно продуманная обратная связь на сайте сможет увеличить конверсию вашего сайта.

УДК 004.5:004.925.5

Студ. А. А. Сугако

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

ВЛИЯНИЕ ЦВЕТОВОЙ КОМПОЗИЦИИ НА ЭФФЕКТИВНОСТЬ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Если обратить внимание на интерфейсы современных веб-страниц, то можно наткнуться на невероятное разнообразие решений.

Для того, чтобы создать наиболее оптимальное решение нужно уделить большое внимание не только интерфейсу, но и его дизайну. Главная задача любого сайта - донести информацию, поэтому любые дизайнерские решения необходимо черпать из функциональной составляющей сайта. Основное внимание уделяется тому, кто будет основной аудиторией, на кого ориентирована информация, услуга или товар. Важными факторами при этом может стать возраст, пол, образование, культурная принадлежность. Также нужно определить какое настроение и какие эмоции должен вызывать сайт.

При выборе цветов следует акцентировать свое влияние на том что каждый цвет вызывает определённое настроение. Например, яркие тёплые цвета (красный, оранжевый, жёлтый) наполняют человека энергией, пробуждают его, а холодные тёмные оттенки (зелёный, голубой, фиолетовый), наоборот, расслабляют и успокаивают.

В своей работе я рассмотрю цветовые предпочтения трех категории сайтов:

- интернет-представительства;
- информационные ресурсы;
- веб-сервисы.

К сайтам интернет-представительств относят *сайты-визитки и корпоративные сайты*, их главной задачей является сформировать благоприятный имидж владельца сайта или компании, для выполнения этой задачи можно использовать цветовые решения с *синим цветом* который вызывает эмоции спокойствия, безопасности, открытости, надёжности, помогает сконцентрировать внимание на стабильности, профессионализме, а также стратегическом лидерстве. Он является весьма распространенным цветом в бизнес-среде и широко используется на корпоративных сайтах с целью создать ощущение уверенности, силы и надёжности.

Однако следует уделить внимание его оттенку, т.к. светлые тона ассоциируются с дружелюбием, а тёмные – с печалью.

Если вы разрабатываете сайт-визитку для отдельных творческих личностей, то обычно используют *бирюзовую* или *сине-зеленую гамму цветов* и их оттенки. Также для данного типа сайтов будет очень актуален *голубой цвет* – это один из самых любимых цветов у 40% пользователей, особенно у мужчин, что связано со стабильностью, профессионализмом и доверием.

Если ваш сайт посвящен кулинарной тематике, то лучшим решением будет оранжевый цвет, т.к. он лучше остальных вызывает аппетит. Хорошим решением будет задействовать близкие цвета желтый и красный, они также используются в оформлении пунктов быстрого питания, так как стимулируют улучшение пищеварения и активацию деятельности желудка. Но ни в коем случае не следует использовать синий цвет, эволюционная теория предполагает, что он ассоциируется с ядом.

Для создания сбалансированной и стабильной атмосферы можно использовать различные *оттенки зеленого*.

Для того чтобы создать ощущение изысканности, элегантности и стиля можно использовать сочетания *черно-белого* в минималистичном дизайне.

Однако самое выгодное решение для таких сайтов это основной цвет логотипа или использование фирменных цветов, лучшее сочетание из возможных.

Промо-сайты и *сайты интернет-витрины* представляют товар для его продажи. На таких сайтах необходим *красный цвет*, он является самым лучшим цветом для стимулирования продаж. *Оранжевый* и *желтый цвета* хорошо продают продукты на тему спорта и детей. Автомобильные концерны предпочитают использовать *серебряный* в сочетании серо-синей, белой и красными красками. Дизайн, выполненный в данном цвете, показывает, что сайт спокойно ждет своего потребителя, который сможет оценить товар по достоинству, при этом цвет не является слишком броским и вызывающим. Люди с аналитическим складом ума, которые не склонны совершать покупки спонтанно, выберут сайты с использованием в дизайне *синего цвета*, а значит стоит обратить на него внимание при разработке интернет-магазина.

Коричневый цвет не желательно применять на подобных сайтах, он создает препятствие продажам.

Информационные ресурсы – это различные тематические сайты, интернет-порталы, блоги, каталоги сайтов данного типа интернет-сайтов характеризуется большим количеством информации по какой-либо конкретной тематике. На таких сайтах главное не перегрузить человеческое зрение и привлечь внимание к контенту. Выбор цвета напрямую будет зависеть от тематики.

Использование на сайте акцентов *красного цвета* хорошо подходит для игровых, молодежных, женских сайтов, ресурсов по развитию бизнеса. *Синий цвет* чаще всего используют на сайтах, связанных с инновациями, технологиями, так как он вызывает чувство надежности и уверенности. *Голубой цвет* хорошо подходит для сайтов на темы путешествий, здоровья, личностного роста. *Фиолетовый цвет* связан с магией и психологией, поэтому он подходит для сайтов духовного и личностного роста, ресурсов астрологической, магической и мистической тематик. *Зеленый цвет* в дизайне успокаивает, отождествляет собой жизнь, рождение, природу, естественность. Он приятен для глаз и не напрягает зрение. Это цвет подходит для темы медицины. Его с удовольствием используют при разработке сайтов о природе, животных, растениях, здоровье.

Для тематических сайтов в качестве контраста ярким тонам хорошим решением является выбор *серого цвета* как фона, он вызывает ощущение чистоты, разумности и стабильности. При этом, серый цвет нейтрален, не акцентирует внимание на себе. Универсальный цвет, который можно использовать на любом сайте.

Говоря о информационных ресурсах несомненно нужно выделить *белый цвет*, он является лучшим цветом для фона под основной темный текст, так как при чтении он не напрягает глаза. Сайты, выполненные в белых тонах очень удобны для навигации, о них складывается хорошее первое впечатление.

Если Ваш сайт имеет основную целевую аудиторию женского или мужского пола, вам следует знать, что существуют большие различия между цветовыми предпочтениями у женщин и мужчин. Нит Патель выявил какие цвета предпочтительны для каждого пола.

В проведенном им опросе 35% женщин сказали, что синий – их любимый цвет, затем идет фиолетовый (23%) и зеленый (14%). 33% женщин признались, что оранжевый и коричневый их самые нелюбимые цвета, кроме этого к самым нелюбимым женщины отнесли еще и серый (17%). Мужчины не любят фиолетовый, оранжевый и коричневый. Им ближе синий, зеленый и черный.

Говоря о *веб-сервисах* можно выделить поисковые и почтовые системы, интернет-форумы, фото-, видео-, аудио- хостинги, доски объявлений, социальные сети

Данная категория сайтов отличается от предыдущих тем, что они более активно взаимодействуют с пользователем, и на них всегда представлен большой объем информации в различном виде. Поэтому идеальным решением для них будет являться выбор светлого фона и различных ярких акцентов.

Самыми распространенными цветами для призыва к действию являются: *красный, зеленый и желтый* они работают на интуитивном уровне, поэтому *желтый цвет* хорошо использовать в качестве предупреждений, *зеленый* для кнопок, *красный* для всплывающий окон или отмены действия.

Для социальных сетей используется *синий цвет* т.к. он вызывает доверие у пользователей.

ИНТЕРНЕТ-ПРЕДСТАВИТЕЛЬСТВА		ИНФОРМАЦИОННЫЕ РЕСУРСЫ	ВЕБ-СЕРВИСЫ
Корпоративные сайты Сайты-визитки	Промо-сайты Интернет витрины	Тематические сайты, интерне-порталы, блоги, каталоги сайтов	Поисковые и почтовые системы, форумы, фото- видео- аудио- ностинги, доски объявлений, социальные сети

Рисунок 1 - Схема цветовых решений.

Если использовать цвет в правильном направлении, в нужное время, для нужной аудитории, и для правильной цели, можно добиться очень высоких результатов, что приведет к максимальной эффективности вашего интерфейса.

УДК 004.514:7.017.4

Студ. А.В. Соловьева

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

УПРАВЛЕНИЕ ВНИМАНИЕМ ПОЛЬЗОВАТЕЛЯ НА ОСНОВЕ СТРАТЕГИИ ЭРГОНОМИКИ ЦВЕТА И КОНТРАСТНОСТИ

Наиболее эффективная стратегия управления вниманием и восприятием пользователя в Интернет-СМИ связана с использованием композиции цвета на веб-странице издания. Это ключевой элемент дизайна, который позволяет дизайнерам определять ценности, свойства и приоритеты элементов относительно друг друга.

Основное воздействие цветовое решение оказывает на пользователя с учетом соотношения выбранной палитры для основных элементов по отношению к фоновым. Таким образом, единичные цветовые акценты приобретают собственный вес и значение в рамках веб-страницы.

Контраст. Если произнести это слово, то первая мысль, которая посетит нас — это контраст цвета. И это правда, но где-то на четверть. Помимо цвета, контраст имеет место в формах, в лицах и взглядах, в размерах, в движениях.

Контраст выступает в качестве образующего элемента эргономически успешной цветовой композиции за счет использования дополняющих друг друга цветовых пар на основе принципов контрастирования и единства. Использование приемов контраста способствует не только формированию эмоционального отклика пользователя на организацию пространства, но и оказывает значительное влияние на восприятие других элементов, расположенных на странице.

Контраст цвета привлекает внимание – это самый простой прием. Цвет используется для выделения основного действия, пользователь сразу переключается на требуемый объект (рисунок 1).



Рисунок 1 - Контраст цвета

Контраст формы — на сложных объектах акцент внимания происходит в первую очередь. Важно отличать фон от объекта, потому что фокус внимания происходит достаточно быстро и неосознанно (рисунок 2).

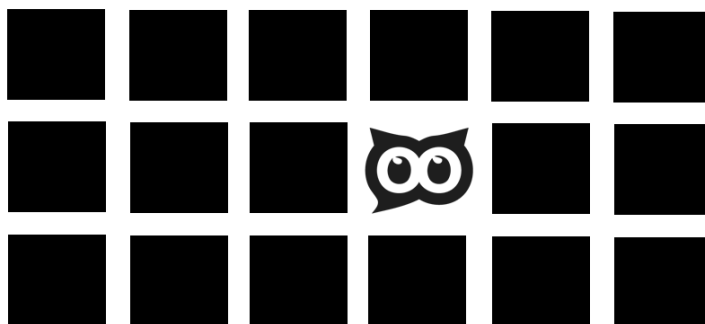


Рисунок 2 - Контраст формы

Лицо и взгляд — «Лицо — естественная ловушка для глаз». Лицо является гарантом того, что пользователь обязательно обратит внимание на нужный объект. Часто этот прием можно увидеть в рекламе (лица детей, актеров). Большое значение имеет направление взгляда. На примере ребенка видим, что взгляд сторону концентрирует внимание на блоке текста (рисунок 3).



Рисунок 3 - Контраст взгляда

Контраст размера — подсознательно объекты более крупные воспринимаются как наиболее важные. Например, большая кнопка «купить» и маленькая «нет, спасибо» используется в интернет-магазинах (рисунок 4).

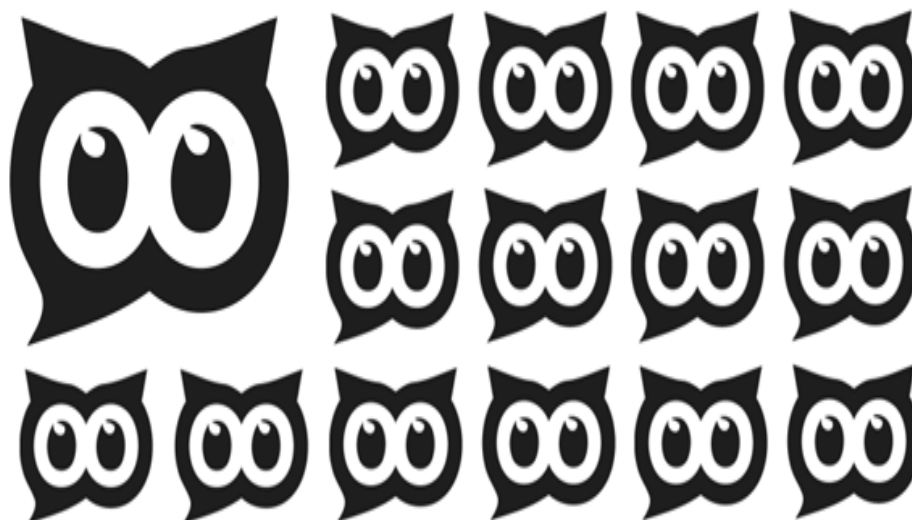


Рисунок 4 - Контраст размера

Контраст движения — движущиеся объекты почти невозможно игнорировать. Мгновенная реакция заложена в человека с далеких времен, когда еще движение воспринималось либо как опасность, либо как еда. Бежит? Схватить или убежать! Таким образом, анимация в управлении вниманием — хорошая вещь. Сюда относится анимация плавных переходов, скроллинг, анимация пиктограмм, реакция на действия пользователя.

Контраст повсюду: открытый и закрытый, полный и пустой, симметричный и ассиметричный, крупный и мелкий, текстура и флэт, заполненный и прозрачный.

Итак, следует помнить, что пользователи редко замечают и оценивают цвет фона, панели навигации, отдельных деталей, но это не значит, что цвет при этом не воздействует на них. Просто это происходит подсознательно.

Контраст является основой графического дизайна. Когда вы понимаете, что контраст — это не только множество красивых цветов и размер текста, вы можете создавать дизайн миллионами способов, как вам захочется. Различия привлекают наш взгляд, а контраст дает нам возможность активизировать восприятие отличий. И если все сделать правильно, это трудно заметить.

Дизайн есть во всем, даже если в него забыли вложить порядок и подчинение правилам. Даже если это зеленые точки на красном фоне. Потому что наши возможности безграничны.

ЛИТЕРАТУРА

1. The psychology of color in web-design [Электронный ресурс]. – Режим доступа: <http://www.vandelaydesign.com/the-psychology-of-color-in-web-design/>. – Дата доступа: 14.04.2017
2. Как использовать контраст в веб-дизайне интерфейсов [Электронный ресурс]. – Режим доступа: <http://sketchapp.me/kak-ispolzovat-kontrast-v-dizajne-interfejsov/>. – Дата доступа: 14.04.2017

УДК 004.5:519.8

Студ. М. А. Новиков

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

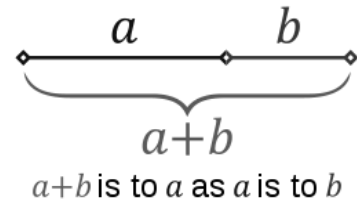
ИСПОЛЬЗОВАНИЕ МАТЕМАТИЧЕСКИХ ЗАКОНОМЕРНОСТЕЙ В ПРОЕКТИРОВАНИИ ИНТЕРФЕЙСОВ

«Математика прекрасна». Это может показаться абсурдным, для людей, которые при одном только упоминании математики вздрагивают. Однако некоторые из самых красивых вещей в природе и нашей Вселенной – это проецирование математических свойств, от самых маленьких до крупнейших галактик. Один из древних философов, Аристотель сказал: «математическим наукам свойственно выстраивать все по порядку, в симметрии и ограничениях, они являются главными формами прекрасного».

Из-за своей природной красоты, математика является частью искусства и архитектуры. Но она практически не применяется для дизайна сайтов и приложений. Это наблюдается из-за того, что многие не сопоставляют математику с дизайном. Хотя, наоборот, математика может быть инструментом для производства, поистине волшебных конструкций. Тем не менее, вы не должны полагаться на математику для каждого своего творения. Дело в том, что вы должны рассматривать математику как вашего помощника. Для наглядности мы создали пару сайтов, которые представляют математические принципы, обсуждаемые в этой статье. Мы также создали рекомендации, которые можно использовать в дизайне.

Золотое сечение «и» Золотой прямоугольник

Золотое сечение, также известное как божественная пропорция является собой иррациональную математическую константу со значением примерно 1,618033987. Золотое сечение (золотая пропорция, деление в крайнем и среднем отношении) – деление непрерывной величины на две части в таком отношении, при котором меньшая часть так относится к большей, как большая ко всей величине (рисунок 1)



Построить золотой прямоугольник достаточно просто. Во-первых, надо построить квадрат. Затем провести линию от середины одной стороны к противоположному углу и использовать эту линию в качестве радиуса дуги, которая определяет высоту прямоугольника. Наконец, завершить прямоугольник, достроив секцию, где угол, в который проведен радиус, является правым нижним углом секции, а левый верхний угол ограничен дугой (рисунок 2).

Рисунок 1 – Формула Золотого сечения

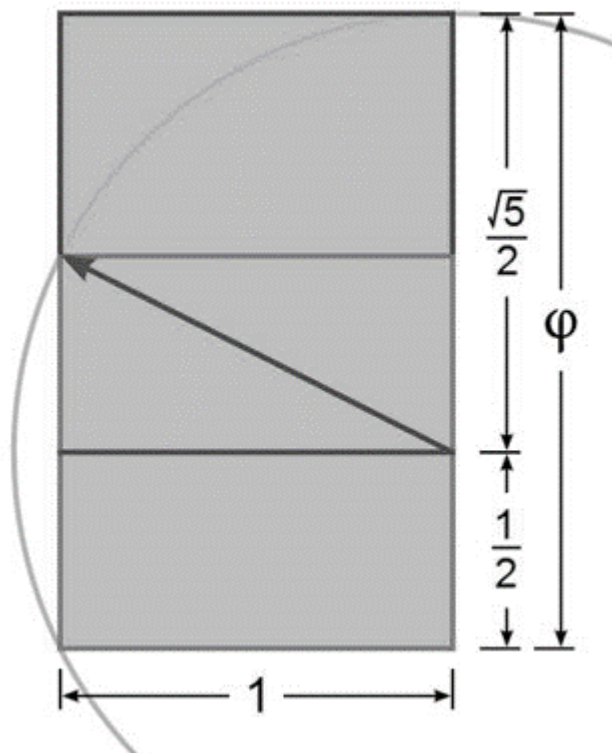


Рисунок 2 – Золотой прямоугольник

В качестве примера рассмотрим минималистский дизайн ниже (рисунок 3). Он имеет шесть золотых прямоугольников, по три прямоугольника в каждой строке. Прямоугольники имеют размеры 299 x 185 пикселей. Таким образом, стороны этих прямоугольников образуют золотое сечение, то есть $299/185 = 1,616$. Обратите внимание, что большое количество свободного пространства, окружающего Золотой прямоугольник создает спокойную и простую атмосферу, в которой каждый блок служит своей цели. Хотя используется всего несколько цветов и все блоки очень похожи, навигация является очевидной и простой.

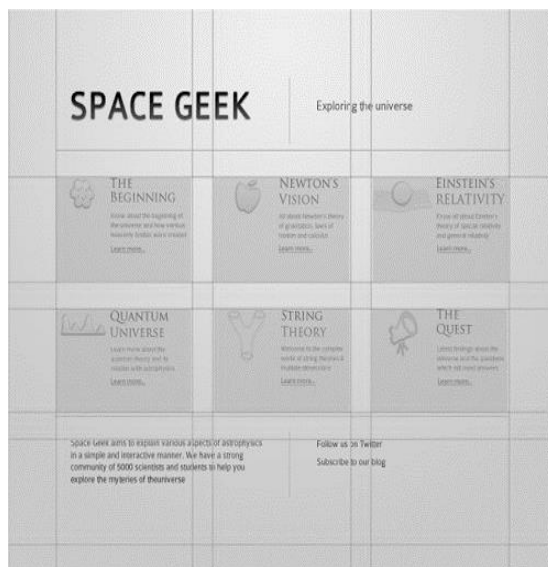


Рисунок 3 – Дизайн на основе золотых прямоугольников

Дизайн Фибоначчи

Как следует из названия, дизайн Фибоначчи основан на последовательности чисел Фибоначчи. По определению, два первых числа Фибоначчи равны 0 и 1, и каждое последующее число равно сумме двух предыдущих. Некоторые источники опускают 0, и начинают последовательность с двух единиц. Итак, два первых числа Фибоначчи равны 0 и 1, и каждое последующее число равно сумме двух предыдущих. Чем больше числа в последовательности Фибоначчи, тем ближе они связаны друг с другом в соответствии с «золотым сечением». Последовательность Фибоначчи выглядит следующим образом:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

В музыке, числа Фибоначчи, иногда используются для настроек, а в изобразительном искусстве, чтобы определить длину или размер содержания или элементов (рисунок 4).

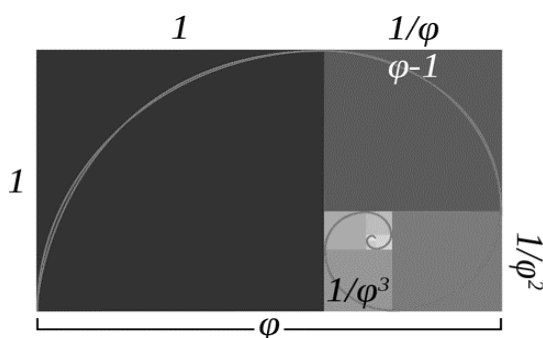


Рисунок 4 – Пример последовательности Фибоначчи

Основная идея такой конструкции состоит в использовании Фибоначчи при решении вопроса о размерах и содержании основной и боковой областях. И Фибоначчи и Золотое сечение равнозначны в данном контексте и пусть дизайнеры полагаются то, что им удобнее.\

В целом, макет довольно легко построить с использованием последовательности Фибоначчи. Вы выбираете определенную ширину базе первого блока — например, 90px. Затем, при определении размера контейнера, нужно умножить базовую ширину на номер блока из ряда Фибоначчи (1,2,3,5,8...). В зависимости от расчетов вы получаете значения, которые являются ширинами блоков, для вашего макета. Давайте посмотрим на примере. Далее минималистический макет, созданный на основе веб-дизайна по Фибоначчи (рисунок 5).

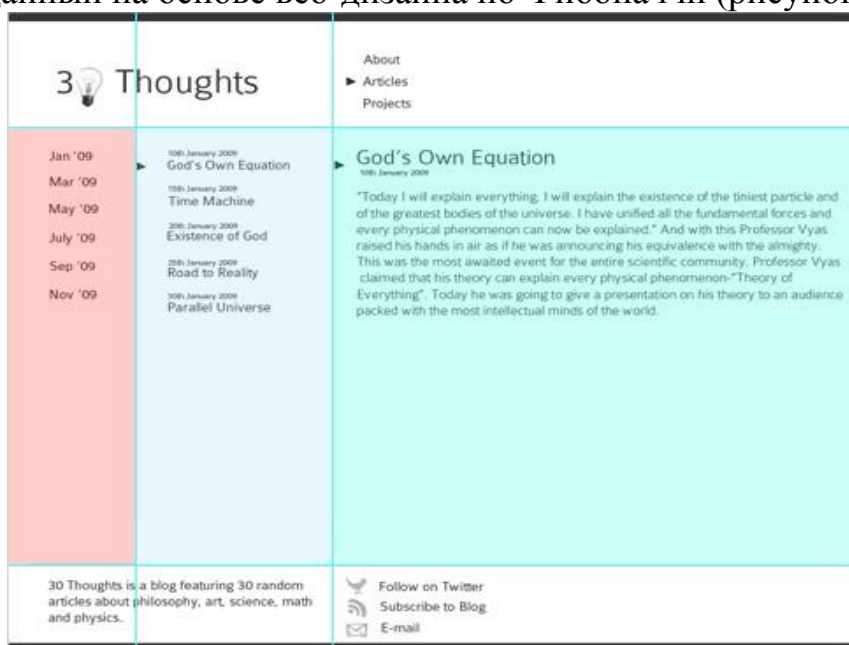


Рисунок 5 – Дизайн на основе рядов Фибоначчи

Некоторые из самых красивых вещей в природе и нашей Вселенной – это проецирование математических свойств, от самых

маленьких до крупнейших галактик. Из-за своей природной красоты, математика является частью искусства и архитектуры. Но она практически не применяется для дизайна сайтов и приложений. Это наблюдается из-за того, что многие не сопоставляют математику с дизайном. Хотя, наоборот, математика может быть инструментом для производства, поистине волшебных конструкций.

УДК 004.5

Студ. Е.С. Котик

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

ИСПОЛЬЗОВАНИЕ FLAT-ДИЗАЙНА ПРИ РАЗРАБОТКЕ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

Flat-дизайн - современный стиль пользовательского интерфейса, а также графического дизайна, отличающийся минималистичностью. Для flat-дизайна характерны использование минимума элементов и отсутствие различных эффектов текстуры, к примеру: смешанных цветов, градиентов, бликов и так далее [1].



Рисунок 1 – Пример flat-дизайна

Flat противопоставляется скевоморфизму (скевоморфизм — это принцип дизайна, когда какому-то одному продукту придаётся облик другого, т.е. когда различные элементы интерфейса копируются с реальных объектов – прим. перевод.), а также насыщенному дизайну. Однако, стоит сказать, что flat-дизайн вовсе не такой простой, каким он кажется на первый взгляд. Он включает в себя некоторые особенности скевоморфизма, что помогает сосредоточить пользователя на контенте, без отвлечения на визуальные эффекты. Flat-дизайн подчеркивает простоту элементов, в то же время делая интерфейс более отзывчивым, приятным и легким в использовании.

Разберём основные моменты, на которые стоит обратить внимание при создании flat-сайта.

Простота. При создании flat-дизайна не стоит использовать большое количество элементов при его наполнении, должно оставаться достаточно много «свободного» места. Дизайн должен выглядеть минималистично, просто и чисто. Благодаря такому принципу внимание пользователей будет концентрироваться на контенте, не отвлекая его на посторонние визуальные эффекты.

Тени. В основном flat отвергает использование любых эффектов. Однако, есть один эффект, характерный для flat-дизайна. Этот эффект стал трендом и характерной особенностью flata – длинные тени. У них есть некоторые типичные характеристики, делающие этот эффект узнаваемым, а именно: 45-градусный наклон и большой размер (тень может быть в несколько раз длиннее самого предмета. В результате, длинные тени придают flat некоторый эффект глубины). Тени изображают относительное возвышение между материальными элементами. Также тени никогда не аппроксимируются красящим материалом.

В основном тени используются для выделения одного объекта над другими, его возвышения. При этом часто также используется его масштабирование, ведь несколько элементов во flat-дизайне не могут одновременно занимать одну и ту же точку в пространстве.

Толщина. Ширина и высота элементов во flat-дизайне может быть различна, но вот толщина при создании сайта у всех элементов должна быть одинаковой. Только при соблюдении данного правила будет достигнут истинный flat-дизайн.

Цвета. Одна из наиболее специфичных особенностей flat-дизайна, помимо теней, это использование цвета. Большинство

цветов, которые использует flat-дизайн в своих элементах состоят всего из нескольких базовых цветов.

Цвет в flat-дизайне отличается яркостью, насыщенностью, сочностью. Цветовая схема flat не ограничивается несколькими особенными цветами. Она содержит множество оттенков, вариаций. Это могут быть как комплементарные схемы, так и составные.

Шрифт. Типографика играет большую роль в flat-дизайне. Часто текст становится главным элементом композиции. Во flat обычно используются простые шрифтовые начертания, делающие весь дизайн в целом чистым и читабельным.

Чаще всего в flat-дизайне принято использовать верхний регистр и контрастные цвета, это делает текст более разборчивым.

На рисунке 1 приводится пример современно flat-дизайна со всеми описанными принципами выше.

Плюсы и минусы. Таким образом, мы можем выделить основные плюсы flat-дизайна:

- популярность. Flat -дизайн превратился в тенденцию, собирая все больше позитивных отзывов\ от дизайнеров и веб-дизайнеров, и вовсе не кажется, что он теряет свои позиции. Наоборот, он распространяется все больше и больше, обретая некоторые новые формы и особенности, становясь все более креативным.

- простота. Flat -дизайн простой, минималистичный и чистый. Flat в вебе помогает пользователям сосредоточиться на контенте, а не отвлекаться на визуальные эффекты. Это также работает для интерфейсов мобильных приложений: чистый дизайн с большими кнопками делает использование мобильных устройств совершенным.

- яркость. Цвет – другой классный плюс в flat-дизайне. Яркие и насыщенные цвета выглядят привлекательными и чистыми, а отсутствие градиентов делает дизайн стильным. Более того, такие чистые цвета делают его более позитивным, презентабельным, flat-дизайн создает нужное настроение.

У flat-дизайна есть еще множество достоинств, но никакой дизайн не идеален, и мы не можем идеализировать его. Здесь некоторые недостатки flat-дизайна, которые мы обязаны упомянуть:

- неотзывчивость. Иногда отсутствие важных деталей или визуальных эффектов делает процесс создания дружеского для пользователя интерфейса сложным, и это в целом делает весь дизайн неотзывчивым. Не всем пользователям flat кажется комфортным, потому что может быть сложным найти элементы на веб-странице,

на которые ты должен кликнуть или нажать на экране мобильного телефона, потому что они не интерактивны.

- проблемы с типографией. Как упоминалось ранее, не всякий шрифт может подойти flat-дизайну. Иногда такой насыщенный шрифт с острыми краями выглядят действительно сбалансировано и стильно. Однако, в том случае, если шрифт выбран неверно, это может разрушить весь дизайн. Ты должен действительно хорошо чувствовать какие шрифты подходят для flat, а какие нет. Недостаток опыта делают выбор шрифта очень трудным.

- отсутствие эффектов и минимализм. Из-за ограничений в использовании эффектов, цветов и шрифтов, flat может выглядеть слишком простым и холодным. Его минималистичность может также стать его главным недостатком – другой flat-дизайн в итоге выглядит точно так же, как и ваш. Поэтому очень сложно сделать ваши иконки или веб-страницы отличными от чужого дизайна, потому что вы используете те же самые упрощенные формы, ограниченные цветовые палитры и аналогичные шрифты. В результате, со временем flat-дизайн может стать скучным.

Вывод. Подведя итоги, мы можем сказать, что в данный момент flat-дизайн является ведущим стилем в веб-пространстве. Придерживаясь именно его, добавляя свои «изюминки», ваш сайт будет современным.

ЛИТЕРАТУРА

1. Top Ten Tips on Creating Flat Design Graphics // ссылка <https://design.tutsplus.com/tutorials/10-top-tips-on-creating-flat-design-graphics--cms-25888>

УДК 004.51

Студ. Н. Н. Чобот

Науч. рук. асс. Т.П. Брусенцова

(кафедра информатики и веб-дизайна, БГТУ)

ИСПОЛЬЗОВАНИЕ ПРИНЦИПОВ ЭМОЦИОНАЛЬНОГО ДИЗАЙНА ПРИ ПРОЕКТИРОВАНИИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА САЙТА «РЮКЗАК»

Эмоциональный дизайн опирается на психологию и творческий подход, его цель – подарить пользователям новые впечатления и создать ощущение, что к ним обращается человек, а не машина. «Эмоциональный дизайн» — дизайн, который дарит эмоции [1].

Дизайн интерфейса – это дизайн для человека, для пользователя. Потребности пользователя можно выстроить наподобие пирамиды Маслоу (рисунок 1).



Рисунок 1 – Пирамида потребностей Маслоу

Красивый дизайн очень привлекает, но ненадолго, если сайт не является функциональным и надёжным.

Индивидуальность — одна из составляющих принципов эмоционального дизайна. Чтобы построить эмоциональную связь с аудиторией, нужно продемонстрировать индивидуальность своего бренда, и аудитория почувствует с ним связь, как с живым человеком. Иногда ощущение присутствия людей появляется, благодаря совершенно абстрактным вещам, например, пропорциям. Пифагор и древние греки осознали это, открыв золотое сечение – пропорцию, свойственную многим творениям природы, включая человеческое тело.

Контраст – мощный дизайнерский инструмент. Убирая с главной страницы все лишнее, нуждающиеся в новых пользователях сайты увеличивают свои шансы. Так и в сайте журнала Рюкзак, главная страница, основанная на визуальном контрасте, показывает, чему посвящен сайт (рис. 2). Пользователю даётся возможность сосредоточиться и выбрать тот пункт меню, который ему нужен. Дизайн не придерживается разметки. Насыщенные цвета делают его больше похожим на коллаж из бумаги, чем на сайт. Это вызывает у аудитории положительные эмоции и делает сайт незабываемым.



Рисунок 2 – Главная страница сайта «Рюкзак»

Простота дизайна помогает потенциальным клиентам быстро проанализировать затраты и выгоды. Это еще одно действие, которое мозг совершает каждый раз, выделив контрасты. Но если внимания требует все вокруг, пользователь ничего не заметит. По закону Хика (рис. 3) время, необходимое для принятия решения, прямо пропорционально количеству альтернатив. Хотя, человеческий мозг — удивительный инструмент, его способность быстро воспринимать большой объем информации ограничена. Каждый раз, когда дизайнер добавляет что-либо к интерфейсу, пользователям становится сложнее выделять закономерности и контрасты.

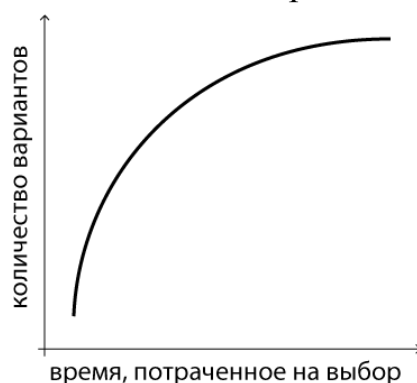


Рисунок 3 – Закон Хика

Главная цель эмоционального дизайна — упростить коммуникацию между людьми. Если дизайнер справляется с этой

задачей, компьютер отходит на задний план. Для этого дизайн следует представлять человеком, готовым общаться с аудиторией.

Создание образа для сайта. Дизайн-образ сайта «Рюкзак».

Название бренда: Рюкзак.

Обзор: логотип (рис. 4) является лицом бренда и его воплощением индивидуальности. Ровные формы говорят о продуманности сайта. Рюкзак привлекает своей простотой и изящностью, позволяет чувствовать себя комфортно. Текст производит впечатление грации, изящности, стиля.

РЮКЗАК

Рисунок 4 – Логотип сайта «Рюкзак»

Индивидуальность: Рюкзак серьезный, деловой, доводит дело до конца. Любит точно формулировать цели и строить конкретные планы. Предпочитает быть впереди остальных: опережать знаниями, освоенными технологиями. Привлекает эрудицией.

Свойства бренда: надежность, неформальность, стиль, простота, функциональность.

Карта индивидуальности: можно поместить разные черты индивидуальности в систему координат XY. Ось X сообщает, насколько индивидуальность дружелюбна, а ось Y позволяет определить степень доминирования или подчинения этой личности (рис. 5).

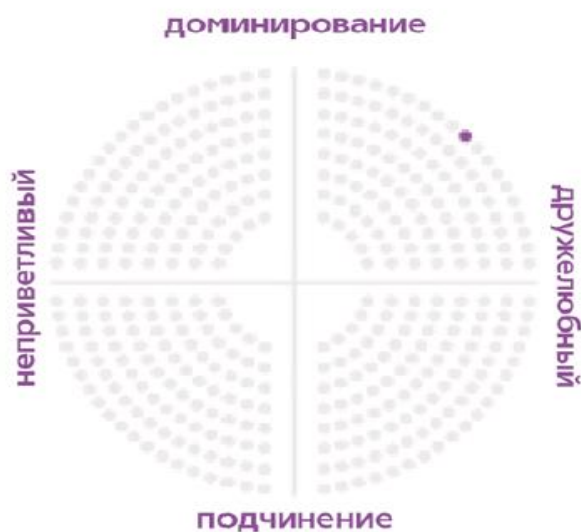


Рисунок 5 – Карта индивидуальности образа сайта «Рюкзак»

Визуальный лексикон:

Цвет. Цвета чистые, но не как в детских телепередачах.

Типографика. Рюкзак легок на подъем, прост и эффективен, и типографика это демонстрирует. Простые шрифты отличаются размером, плотностью и цветом, передают иерархию сообщений.

Общий стиль. Элементы интерфейса простые, интуитивно понятные.

Методы связи: На сайте Рюкзак есть процентная полоса, которая заполняется вместе с прогрессом прочтения.

Индивидуальность — способ взаимодействия с аудиторией. Она дает пользователям понять стиль бренда, формирует эстетику и интерактивный дизайн сайта.

Эффект предшествования — это явление, когда реакция человека на воздействие того или иного стимула определяется предшествующим действием того же самого или подобного стимула. Например, в играх всегда интересно открыть новый раунд, а в интерфейсе — перейти на новую страницу и увидеть другое оформление.

Дизайнеры имеют уникальную возможность — помогать пользователям следовать интуиции. Используя инструменты дизайна, такие как верстка, цвет, линия, текст или контраст, дизайнер подаёт информацию так, что пользователь принимает решение, руководствуясь скорее инстинктом, нежели разумом. Задача — воздействовать на чувства аудитории, создать ощущение, что выгода значительно превышает затраты.

Таким образом, эмоциональный дизайн — это не просто способ произвести положительное впечатление и тем самым преодолеть временные трудности. Его задачи гораздо шире: он помогает справляться со сложными ситуациями, такими как падение сервера, потеря информации или атака вирусов, влияющих на работу сайта. Ошибки случаются, и у любого человека что-то может пойти не так. Но правильная реакция и запас доверия, накопленный в результате эмоционального взаимодействия с аудиторией, помогут пережить тяжелые времена.

ЛИТЕРАТУРА

1. Аарон Уолтер «Эмоциональный веб-дизайн» – Манн, Иванов и Фербер: ISBN 54457094774, 9785457094772, 2012, – с. 191.

УДК 004.031.42:004.92:004.738.5

Студ. А. Н. Щербакова
Науч. рук. ассист. Т. П. Брусенцова
(кафедра информатики и веб-дизайна, БГТУ)

ИНТЕРАКТИВНАЯ ИНФОГРАФИКА В СОВРЕМЕННОЙ ИНТЕРНЕТ СФЕРЕ

В настоящее время инфографика все больше и больше набирает обороты, занимая важное место в печатных и интернет СМИ. Она помогает публике понять значение информационного сообщения и его смысл более быстрым и понятным способом.

Инфографика (infographics) — визуальное изложение сложных данных в очень простой и привлекательной форме с помощью креативных и визуально привлекательных изображений.

Инфографика вызвана помочь автору доступно донести сложную информацию и большой объём данных читателю. С помощью инфографики пользователю удобнее и легче воспринимать информацию.

Появилась инфографика в 1982 году как результат эволюции в информационной сфере, постоянно развивалась, стала одним из самых модных, актуальных и влиятельных видов визуальной коммуникации и получила огромное распространение. Инфографика применяется там, где сложные данные должны быть представлены быстро и четко. Это значительно облегчает процесс разработки и восприятия нужной информации.

Далеко не каждое изображение с данными можно назвать инфографикой. Необходимо соблюдать определенные правила при ее создании, в противном случае можно получить иную интерпретацию той информации, которую вы хотели донести, и несоответствующую ожиданиям реакцию аудитории на нее.

Можно выделить четыре основных правила, которых рекомендуется придерживаться при создании инфографики. Это содержательность, смысл, легкость восприятия и аллегоричность.

Из всех правил содержательность является первоначальным и самым важным.

Вторым правилом инфографики является смысл. На этапе проектирования и создания инфографики его легко потерять. Например, из-за несогласованности действий команды проектировщиков и разработчиков. Если дизайнер при создании инфографики увлечется визуализацией, то это может привести

к полному изменению смысла сообщения, которое хотел передать аудитории автор.

Третьим правилом является легкость восприятия. Оно, в большей степени, относится к визуальной составляющей, а здесь легко допустить много ошибок. Так, если рассмотреть пример создания инфографики в виде диаграмм, то основным вопросом при разработке дизайна является правильный подбор ее типа. Например, разницу между 52% и 62% на круговой диаграмме различить практически невозможно. А вот столбчатая гистограмма для этих целей отлично подходит.

Аллегоричность является четвертым правилом создания инфографики. Привлекательная возможность инфографики – превратить скучную, комплексную информацию в графическую аллгорию, с помощью которой даже непрофессиональная аудитория мгновенно сможет понять суть посылки, тематику и основную идею.

Для создания инфографики могут использоваться таблицы, диаграммы, различные графические элементы и т.д.

Вершина качества инфографики – интерактивная инфографика. Она позволяет представить большой объем разношерстной информации в организованном виде, который будет удобен для просматривающего. Существует несколько видов и определений интерактивной инфографики.

Первое: интерактивная инфографика – это инфографика, которая предполагает участие читателя в управлении отображаемыми данными. Интерактивность создается с применением технологий Flash, JavaScript или HTML 5.

Второе: любая инфографика, которая позволяет в онлайн режиме изменять данные, отображаемые в графической форме. Пример интерактивной инфографики представлен на рисунке 1 (рисунок 1).

Данная инфографика представляет собой интерактивную карту мозга, которая стимулирует пользователей нажимать на рисунок мозга и читать о функциях его отделов. Данная информация могла бы занять слайдов 30, однако интерактивная инфографика позволяет представить это одним изображением.

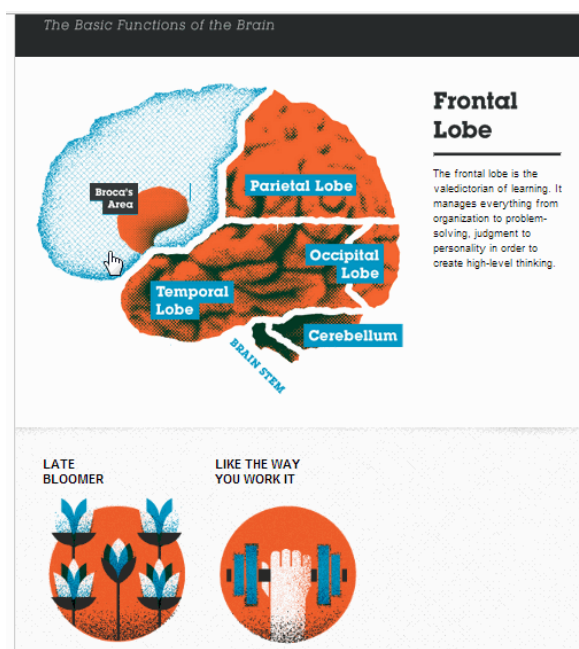


Рисунок 1 – Пример интерактивной инфографики

Есть три главные ступени для успешного создания инфографики в печати и в интернете:

- 1) четко понимать, что за тип информации будет представлен: пространственный, хронологический, количественный или смешанные комбинации;
- 2) представлять себе соответствующий образ информации как единого целого, то есть гораздо больше, чем совокупность карт, схем, чертежей и диаграмм;
- 3) выбрать нужный тип презентации – статический, динамический или интерактивный.

Руководствуясь этапами создания инфографики, была создана инфографика для сайта БГТУ: «Как попасть в санаторий-профилакторий БГТУ?».

1 этап – выбор темы.

Студенты БГТУ постоянно задаются вопросом «Как попасть в санаторий-профилакторий БГТУ?», поэтому в качестве темы была выбрана данная.

2 этап – сбор информации.

Сбор информации зависит от темы. В качестве информационного источника использовался сайт БГТУ.

3 этап – построение «скелета» инфографики (блоков текста с предполагаемой информацией). Информация была разбита на 4 блока.

4 этап – разработка **графического эскиза**.

5 этап – **художественное оформление** инфографики.

В результате получилась работа, представленная на рисунке 2 (рисунок 2).

Как попасть в санаторий-профилакторий БГТУ?

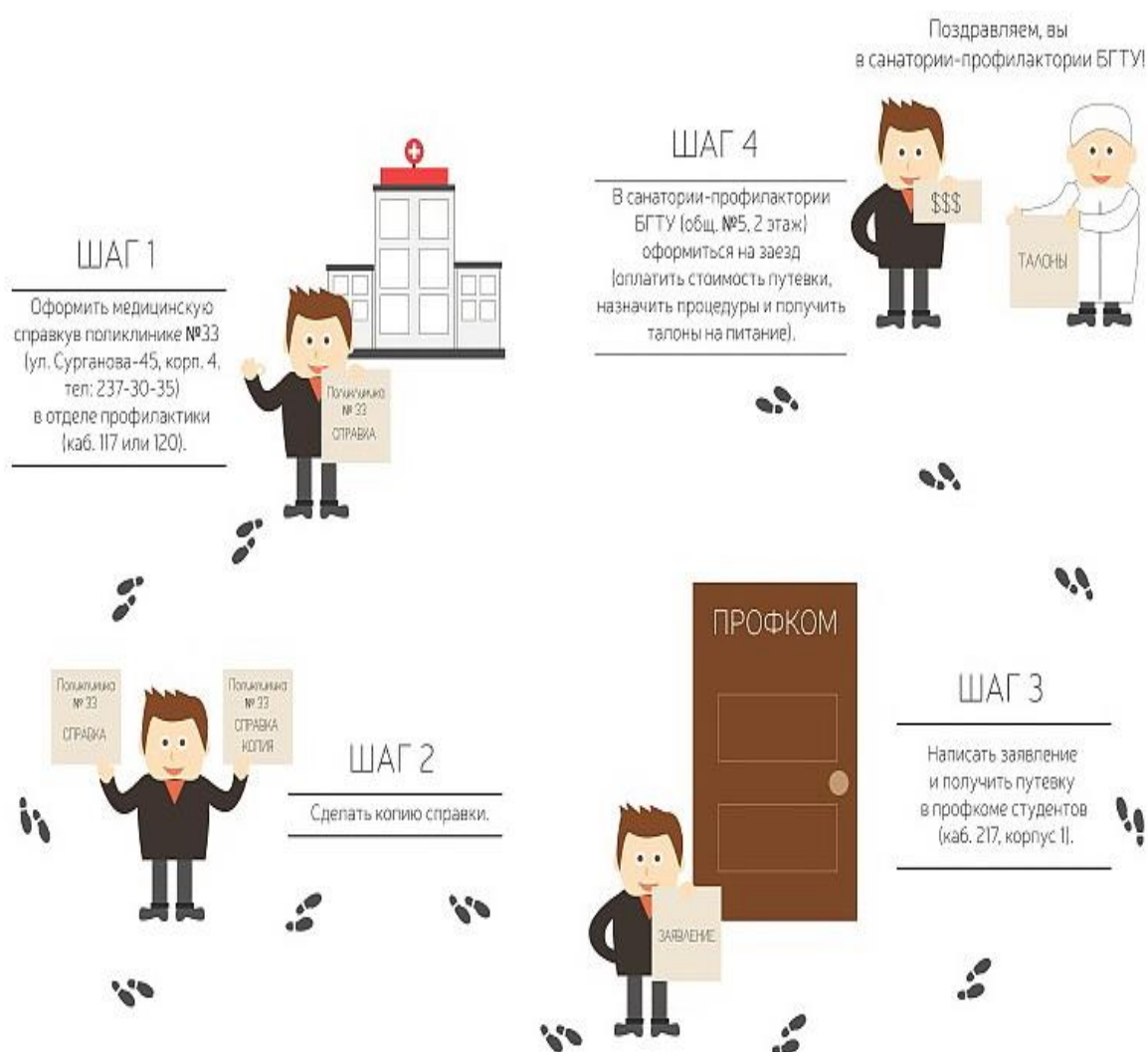


Рисунок 2 – Инфографика «Как попасть в санаторий-профилакторий БГТУ»

Инфографика позволяет представить большой объем текстовой информации в графическом виде, что облегчает восприятие и понимание материала, ведь известно, что мозг на 90% воспринимает визуальную информацию, а скорость обработки графики в 60 тысяч раз быстрее скорости обработки текста.

ЛИТЕРАТУРА

1. Интерактивная инфографика [Электронный ресурс]. – Режим доступа: <http://comagency.ru/interaktivnaja-infografika>. – Дата доступа: 11.04.2017
2. Этапы создания инфографики [Электронный ресурс]. – Режим доступа: <https://infogra.ru/lessons/6-etapov-sozdaniya-infografiki>. – Дата доступа: 10.04.2017

УДК 004.738.5

Студ. В.Ю. Тонкович

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

РАЗНОВИДНОСТЬ БЛОГОВ. РАЗРАБОТКА ДИЗАЙНА БЛОГА

Блог (англ. *blog*, от *web log* — интернет-журнал событий, интернет-дневник, онлайн-дневник) — веб-сайт, основное содержимое которого — регулярно добавляемые записи, содержащие текст, изображения или мультимедиа. Для блогов характерны недлинные записи временной значимости, упорядоченные в обратном хронологическом порядке (последняя запись сверху). Отличия блога от традиционного дневника обуславливаются средой: блоги обычно публичны и предполагают сторонних читателей, которые могут вступить в публичную полемику с автором (в комментариях к записи или своих блогах).

Есть разные виды ресурсов. Их различают по способам оплаты, по авторству, содержанию и по целям создания.

По способу оплаты все блоги делят на:

- автономные;
- созданные на бесплатной платформе.

Если в перспективе получение заработка — лучше создавать ресурс автономный, то есть на платном хостинге.

По авторству интернет-дневники делят на:

- корпоративные;
- коллективные;
- авторские.

Авторский, личный блог ведет один человек. Автор делится своими мыслями или собственным опытом.

Правила ведения и форма коллективного блога установлены группой участников, которые пишут посты для него. Такие сообщества постоянно развиваются, расширяя аудиторию читателей и авторов. Ценной информацией делится эксперт, новые посты

комментируются в группах. Для ведения нанимают специалиста или выполняют работу всем коллективом.

Корпоративные блоги создают сотрудники компании. Цель – повысить популярность организации, привлечь новых клиентов и сплотить коллектив.

По целям создания различают:

- тематические;
- общетематические;
- имиджевые;
- бизнес-ресурсы

Тематические ресурсы посвящают только одной теме: или музыке, или только политике, или путешествиям, или увлечению

Общетематические – блоги «обо всем». В сети их достаточно, некоторые довольно популярны, но лучше определить для себя одну тему и вести ее.

Имиджевые блоги ведут и определенные люди, и целые компании для раскрутки собственного бренда. Рассказывает автор о своем опыте, серьезных проблемах и предлагает способы их решения. Помогая другим, он одновременно обретает популярность. Подобные блоги подходят для поиска новых бизнес-партнеров либо клиентов.

Для продвижения продукции или услуг создают бизнес-сайт. Здесь все понятно: все сообщения на них касаются только определенных вопросов по продукции или оказанию услуг.

По содержанию выделяют блоги

- текстовые;
- видео или влоги;
- подкасты;
- фотоблоги.

Текстовый – статьи, хотя авторы часто используют видео и снимки в зависимости от тематики.

Популярность влогов растет быстро: видеоконтент удобен. Записывать видео очень просто.

Подкасты – блоги, где посетители могут загружать постоянно обновляемые аудиофайлы.

Фотоблоги очень оригинальны и популярны. Главное на них – фотоснимки, а не тексты. Небольшие заголовки или комментарии лишь сопровождают фото.

Есть и более редкие разновидности блогов: тайпкаст, тлог или микроблог. Отличие тайпкастов от других в том, что автор пишет тексты на пишущей машинке, а на блоге размещает только им

отсканированные статьи. Публикуют различные виды информации на блоге. Но для одной записи – только один формат: видео, текст либо ссылки. Это главное требование. В микроблоге публикуют лишь небольшие сообщения, ссылки или видео.

Дизайн блога: какой он должен быть

1. Гибкий формат. Полная свобода, чтобы обсудить различные темы. Даже если блог имеет центральную тему, они могут время от времени отрываться от нее, чтобы исследовать другие идеи, которые их интересуют;

2. Записи могут быть как короткими, так и длинными, как вам нравится;

3. Записи не обязательно должны содержать текст. Видео и фото блоги могут быть привлекательными и популярными;

4. Блоги можно использовать для становления кого-то в качестве эксперта в своей области, тем самым увеличивая возможности для бизнеса;

Независимо от типа блога он включает:

- Список категорий, обычно в боковой панели;
- Архивы статей по датам;
- Отдельные статьи с комментариями;
- Страница «Обо мне» и/или «Контакты».

Это не плохая структура работает и довольно распространённая. Но это не означает, что все должны её придерживаться.

Мною был разработан дизайн блога, который содержит следующие необходимые элементы (рисунки 1,2,3):

- *Главная страница с последними записями.*

Работа дизайнеров заключается не только в том, чтобы удостовериться, что текст достаточного размера для чтения, а меню легко определяется, или кнопки достаточно большие для нажатия. Они решают проблемы, чтобы быть уверенным, что пользователям легко и быстро найти то, что они ищут.



Рисунок 1 - Гл. страница

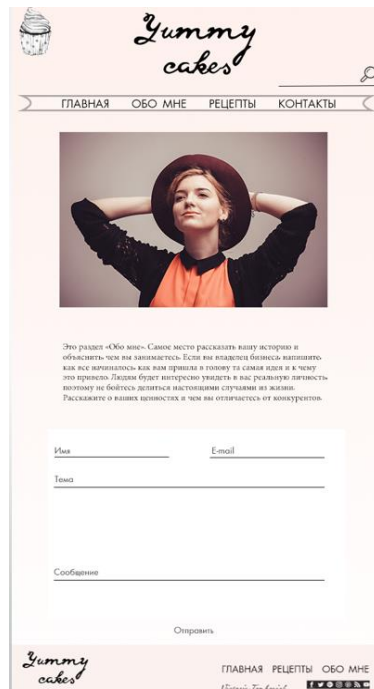


Рисунок 2 - Стр. «Обо мне» Рисунок 3- Стр. со статьей



- Боковая панель.

Одна из проблем боковой панели в том, что она слишком часто отвлекает от содержимого. Это может быть красочное анимированное объявление, что еще хуже, если оно со звуком и без кнопки его отключения. Или сама боковая панель конфликтует с остальной частью сайта. Порой, владельцы блогов умудряются добавить слишком много информации в боковую панель своего блога. Это приводит к тому, что боковая панель намного длиннее, чем статья.

Поэтому боковые панели не следует фаршировать дополнительной информацией и виджетов, и они не должны доминировать на странице. Их предназначение — средства навигации.

- Домашняя страница.

Чаще всего только постоянные читатели начинают с просмотра домашней страницы. А многие люди выстраивают домашнюю страницу своего блога так, будто это будет первое, что увидит каждый пользователь. Это ошибка.

- Хронологическая навигация.

На большинстве блогов у опубликованных постов стоит дата публикации. Это позволяет пользователям перемещаться по содержанию сайта по месяцам или годам.

Тем не менее, хронологическая навигация не всегда необходима. Если вы пишете о вещах, которые не меняются в спешке, как прошедшие исторические события или породы животных, можно совсем отказаться от навигации по времени.

- *Комментарии.*

Комментарии могут оказать неоценимую обратную связь, а участие и взаимодействие пользователей может привести ваших читателей к вам на сайт снова и снова.

ЛИТЕРАТУРА

1. Актуальные тенденции веб-дизайна «Niklenburg» [Электронный ресурс] / Режим доступа: <http://niklenburg.com/> — Дата доступа: 09.04.2017.

2. Свободная энциклопедия «Wikipedia» [Электронный ресурс] / Режим доступа: <http://ru.wikipedia.org/> — Дата доступа: 13.04.2017.

УДК 004.928

Студ. Д.С. Лозюк

Науч. рук. асс. Т.П. Бруснецова

(кафедра информатики веб-дизайна, БГТУ)

АНИМАЦИЯ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

Все люди понимают и многие любят мультипликацию, но анимацию специалисты не умеют использовать за редким исключением. Никто не понимает, зачем она, проектировщики и дизайнеры интерфейсов воспринимают ее, как украшательство. Это заблуждение.

Зачем нужна анимация интерфейса?

Основное предназначение анимации — сделать интерфейс удобнее. Но как повысить юзабилити, используя анимацию? В статье я попробую разобраться, какие взаимодействия удобны людям и почему, а так же привести примеры удачного использования анимации в интерфейсах.

Что такое кнопка в интерфейсе? Это элемент, активирующий полезное действие. Отправка заявки на получение кредита происходит при нажатии на соответствующую кнопку.

Одновременно с этим, кнопка — это отсыл к аналоговому. вспомните свой телефон или магнитофон с кнопками. Кнопка — это один из первых способов, применяемых дизайнерами интерфейсов для привычной имитации взаимодействия с объектами реального мира. Когда

человек нажимает на аналоговую кнопку, он ожидает получить определенный результат. Кнопка в цифровом мире — привычный инструмент, знакомый из реальной жизни, с помощью которого достигается полезное действие.

Базовые принципы имитации реального мира

1. Физическая модель

При разработке интерфейса, дизайнер должен сформулировать физическую модель, согласно которой будут двигаться и взаимодействовать все элементы.

Например, необходима физическая модель сайта — презентация. Отталкиваемся от работ — это главное. Каждая конкретная работа является стопкой слайдов. Человек может хватать или свайпить стопки слайдов горизонтально. Выбранную работу можно просмотреть, начав листать «экраны» вертикально, раскрыв все слайды конкретной стопки, т. е. конкретной работы. При вертикальном свайпе (листании) мы наглядно показываем, что слайды находятся в стопке, один над другим. Это достигается правильной интерактивностью, которая основана на принципах анимации интерфейса и имитации объектов реального мира. Так в общем виде формулируется физическая модель.

2. Контекст / связь

Связь между различными состояниями интерфейса способствует созданию иллюзии живого взаимодействия.

Вы решили почитать книгу перед сном. Обычную бумажную книгу. Что вы делаете, когда прочитали весь текст разворота? Скорее всего, вы берете и переворачиваете страницу. Процесс перелистывания — это пауза в чтении. Хорошо это или плохо? С позиции полезного действия книги — это плохо, с позиции человека — это привычно. Для дизайнера интерфейса «хорошо», зачастую, означает «удобно». Но полезное действие не отменяется, оно всегда на первом месте. Перелистывание страницы — это привычное действие, позволяющее видеть связь между страницами.

Плохо, если в конце страницы вам нужно нажать на кнопку и страница мгновенно изменится на другую. Без связи и процесса перелистывания сложнее понять, что происходит и почему.

Мгновенное изменение состояния невозможно встретить в жизни.

Старый способ загрузки веб-страниц не является естественным для реального мира. Действительно, людей «заставили привыкнуть к плохому», т. к. технологии не позволяли имитировать живые взаимодействия качественно.

Фактически, 2 варианта:

Старый способ загрузки. Когда страница перезагружается и человек какое-то время видит пустой экран браузера.

Современный контекстный способ загрузки. Показать процесс перехода от одного экрана к другому, согласно заранее продуманной физической модели. В данном случае человек «проваливается» в работу. В процессе видна связь одного экрана с другим, человек точно понимает, куда и как он попал. Этот способ обеспечивает юзабилити качественного нового уровня по сравнению со старым способом загрузки.

3. Анимация, имитирующая живое взаимодействие

Одно из предназначений анимации интерфейса заключается в создании иллюзии взаимодействия с объектом реального мира.

1. Анимация как окно в реальный мир

Людям привычно посмотреть утром погоду за окном. Стандартное iOS-приложение «Погода» превращает бездушный кусок стекла в настоящее окно в реальный мир, в котором в любую секунду видна текущая погода. В данном случае реалистичная анимация оправдана.

2. Свойства реального мира

Анимация используется для имитации взаимодействий, с которыми вы сталкиваетесь в жизни. Речь идет о динамических характеристиках, таких как инерция, эластичность, вес и пр.

Инерция прокрутки ленты любого популярного приложения. Разработчики должны понимать, что в жизни практически не встречается равномерное прямолинейное движение. Все объекты реального мира перемещаются с ускорениями и замедлениями.

Тяжелые объекты имеют большую инерцию, чем легкие, это влияет на тип анимации и изинг (от англ. *easing* — ослабление, смягчение). Изинг — математическая модель ускорения или замедления. Для одного типа анимации применяется экспоненциальный тип замедления (замедление — это анимация типа *out*), для другого — синусоидальное ускорение (анимация типа *in*) и т. п. Для каждого конкретного случая скорость, длительность анимации, изинги и типы анимаций выбираются индивидуально. Это обеспечивает имитацию живого взаимодействия.

Примеры некоторых моих работ

Многю были разработаны некоторые примеры анимации в пользовательском интерфейсе (рисунок 1, рисунок 2).

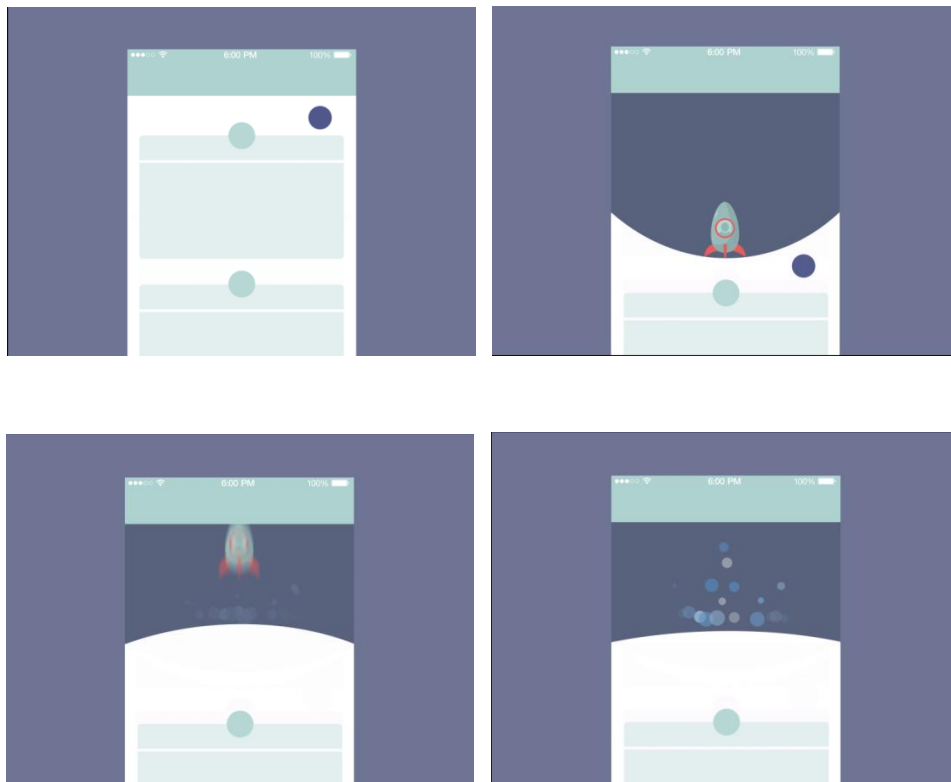


Рисунок 1 — Раскадровка анимации загрузки страницы

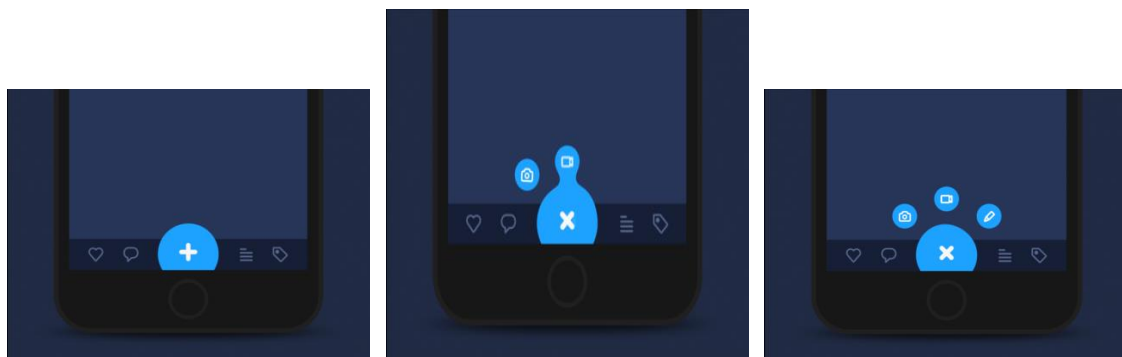


Рисунок 2 — Раскадровка анимации пользовательского меню

Заключение

Дизайн интерфейса, анимация и технологии — 3 составляющие качественного продукта, работающие только в связке. На сегодняшний день в компаниях, претендующих на экспертизу в области юзабилити, должны быть специалисты, разбирающиеся в анимации объектов реального мира, имеющие опыт работы с интерфейсами и тесно

взаимодействующие, как с дизайнерами, так и с техническими специалистами.

ЛИТЕРАТУРА

1. Тайминг в анимации / Whitaker H., Halas J (русская редакция Ф.Хитрука). 2002. –№1. – С. 26–70.
2. THE ANIMATOR'S SURVIVAL KIT / Richard Williams, 2009 – 342 с.

студ. В.С. Хворост

Науч. рук. асс. Т. П. Брусенцова

(кафедра информатики и веб-дизайна, БГТУ)

ПСИХОЛОГИЧЕСКИЕ АСПЕКТЫ ВЗАИМОДЕЙСТВИЯ ЧЕЛОВЕКА И КОМПЬЮТЕРА

Изучение психологических аспектов взаимодействия человека и компьютера, а также поиск эффективных методов применения информационных технологий приобретают в настоящее время особую актуальность. Чтобы сформировать сообщение, которое нужно донести пользователю, дизайнеры используют такие инструменты как шрифт, графика, формы, линии, изображения и цветовые оттенки. Дизайн сайтов в наше время имеет больше отношение к изучению психологии восприятия и поведения пользователя, чем к овладению творческими художественными способностями. Знание психологических принципов и законов восприятия помогает создавать не только красивые, но и юзабельные интерфейсы.

Понятие ментальное моделирование плотно вошло в работу UX-дизайнеров и разработчиков. При создании сайта либо приложения, проектировщик должен понимать, как ведут себя пользователи и что для них будет понятно на интуитивном уровне. Таким образом Ментальное моделирование — это процесс изучения поведения пользователя и применение этих поведенческих моделей при создании чего-либо виртуального. В ходе проектирования дизайнер или разработчик должен постоянно проверять свою работу следующими вопросами: понятно и ясно ли общее послание от сайта (раздела)? Нет ли раздражающих элементов?

Рассмотрим применение принципов ментального моделирования на примере макета сайта «kompiki.by», деятельность которого направлена оказание услуг по ремонту компьютеров (рисунок 1).



Рисунок 1 - Макет сайта «Kompiki.by»

Обратите внимание на изображение главной страницы, какое послание она несет? Указано ли на странице, что это ремонт компьютеров? Написаны ли преимущества компании? На посадочной странице нет главного рекламного текста для своих потребителей.

Зачастую при проектировании дизайна сайта необходимо привлечь внимание пользователя к конкретной точке, это может быть слоган либо конкретное изображение. Для этого проектировщик использует другой шрифт, цвет или размер элемента. Эффект человеческой памяти, при котором объект, выделяющийся из ряда однородных, запоминается лучше других называется эффектом Ресторфф или эффектом изоляции. На рисунке 2 представлена главная страница сайта кузнечной мастерской. Внимание пользователя привлекается к тексту: «Ручная художественная ковка». Это достигается выбором гарнитуры шрифта, его размера, а также размещением текста по центру).

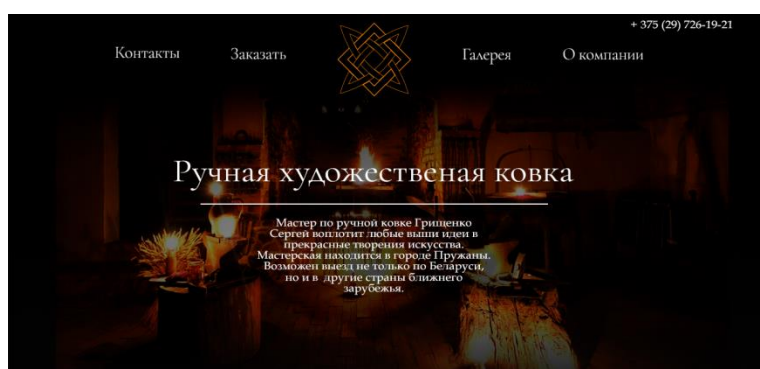


Рисунок 2 - Страница сайта кузнечной мастерской

Гештальт-психология исследует зрительное восприятие объектов относительно друг друга. Объекты, расположенные рядом, воспринимаются как единая логическая группа, а не в качестве отдельных элементов. Это гештальт-принцип близости. Принцип

сходства: похожие объекты воспринимаются как единое целое или как часть одной группы. Принцип замкнутости: форма воспринимается целой, даже если не все линии замкнуты. На рисунке 3 вы видите 4 прямоугольника, хотя линии, создающие их, не являются прямоугольниками по своей сути это просто линии. Только из-за активной обработки разумом мы склонны видеть прямоугольник, а не набор разъединенных линий.

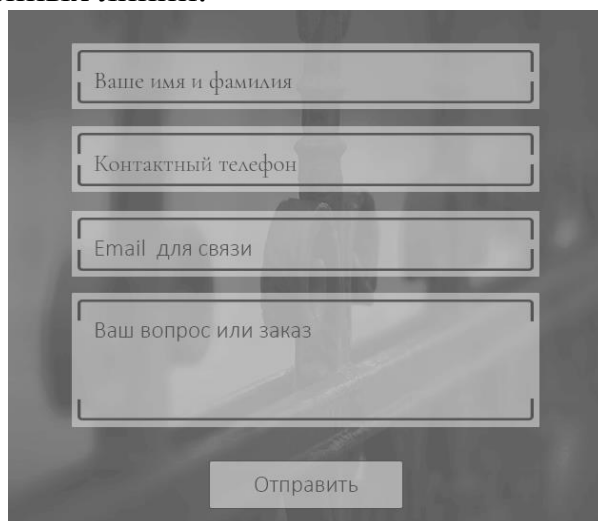


Рисунок 3 - Принцип замкнутости

Принцип непрерывности: зрительное восприятие естественным образом движется от одного объекта к другому. Фигура и фон: когда глаз видит объект как объект, он отделяет его от окружающего пространства (фона). Восприятие любого изображения начинается с того, что глаз отделяет фигуры (объекты) от фона.

Висцеральные реакции это психологические механизмы, происходящие из глубин подсознания. На висцеральном уровне восприятия эмоциональные сигналы из окружающей среды интерпретируются автоматически. Для построения веб-ресурса на основе висцеральных реакций дизайнеру необходимо использовать красивые фото и красочные образы, чтобы эмоционально зацепить аудиторию. В разделе отзывы сайта кузнечной мастерской используется висцеральный дизайн, чтобы передать счастье и удовлетворение клиента после заказа товара на сайте. Молодая улыбающаяся девушка оставляет положительные эмоции у посетителей сайта (рисунок 4).

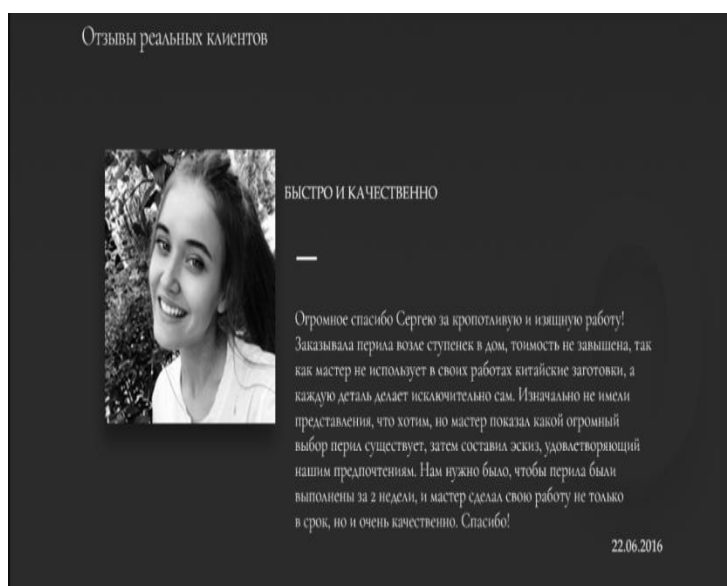


Рисунок 4 - Красивые фото

Психология форм. Люди ассоциируют различные формы с чувствами, эмоциями и настроениями. На эту тему было проведено много исследований. И было выявлено, что круги, овалы, эллипсы – вызывают позитивные эмоции, связанные с сообществом, дружбой, связями, отношениями, единством, женственностью. В то время как прямоугольники и треугольники означают стабильность, баланс, силу, профессионализм, эффективность, мужественность.

В качестве логотипа для кузнечной мастерской был выбран символ Бога- кузнеца. Четкие линии квадрата, символизирующие стабильность, мужественность гармонично переплетаются с более мягкими немного овальной формы дугами, что добавляет нотки доверия и дружелюбности в логотип (рисунок 5).



Рисунок 5 - Логотип кузнечной мастерской

Давно известно, что мозг человека обрабатывает визуальную информацию в 60 000 раз быстрее текстовой. В проектировании и дизайне активно используют сочетание визуальных и вербальных элементов, это явление получило название теория двойного кодирования. В частности использование принципов этой теории выражается в сопровождении текста иллюстрирующими изображениями. В разделе, поясняющем порядок работы с клиентами в кузнечной мастерской, изображения являются буквальными иллюстрациями, наглядно отражающими каждый этап работы с клиентами.

На примере двух макетов сайтов, созданных мной, был произведен обзор небольшой части психологических аспектов взаимодействия человека и компьютера. И наглядно отражено, как хороший дизайнер может умело использовать каждый принцип психологического восприятия и влиять на эмоции и решения людей.

УДК 004.41

студ. А. В. Ярошук

Науч. рук. асс. И. А. Миронов

(кафедра информатики и веб-дизайна, БГТУ)

ПРОГРАММНЫЙ МОДУЛЬ ВЕБ-ПОРТАЛА УНИВЕРСИТЕТА «ЛИЧНЫЙ КАБИНЕТ СТУДЕНТА»

В данной статье рассматриваются основные моменты разработки персонализированного рабочего пространства (личный кабинет) студента, предназначенное для своевременного получения актуальной информации об учебном процессе, о финансовых отношениях и иных видах взаимодействия студентов с университетом, а также для предоставления доступа к необходимым для обучения материалам, ресурсам и информационным системам.

Личный кабинет будет интегрирован в корпоративный информационно-образовательный портал университета в качестве одного из сервисов.

Для получения доступа к интерфейсу личного кабинета пользователи проходят процедуру аутентификации, которая выполняется на основе данных учетных записей из единой базы данных студентов и сотрудников для доступа к информационным ресурсам вуза. Права пользования личным кабинетом предоставляются всем студентам, находящимся в процессе обучения на текущий момент времени.

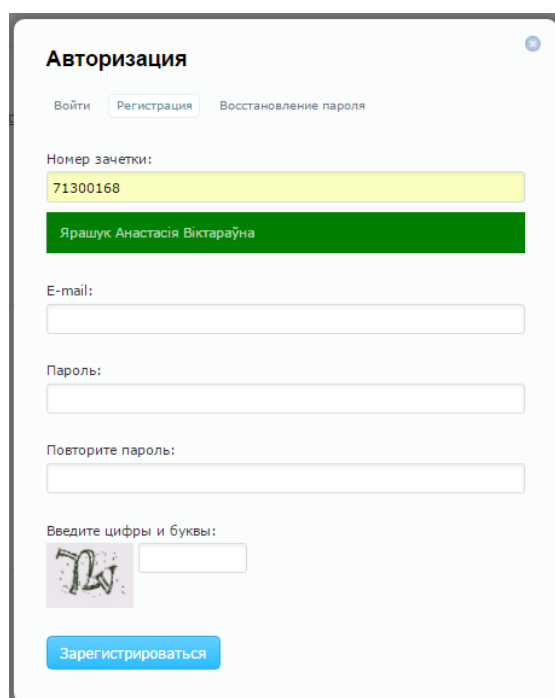


Рисунок 1 – Проектирование страницы авторизации студента

После аутентификации во вкладке «Профиль» появится факультет, курс и группа студента. Также студент сможет загрузить свою фотографию в профиль и другую различную персональную информацию.

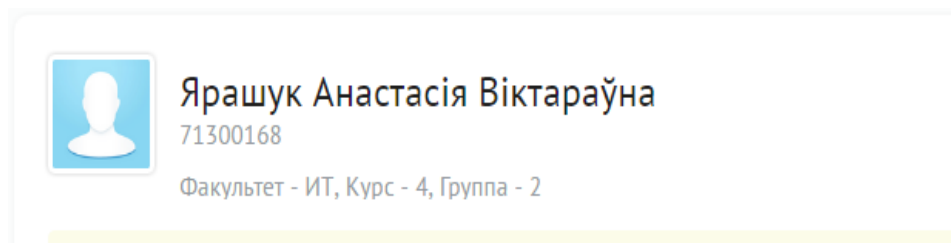


Рисунок 2 – Шаблон части профиля студента

В личном кабинете студента будут находиться следующие модули:

- Расписание;
- учебные планы;
- список дисциплин;
- обсуждения;
- электронные учебно-методические комплексы;
- текущая успеваемость (результаты сессий);
- документы;

– внеучебные достижения студентов и др.

Модуль «*Расписание*» предоставляет функции просмотра актуального расписания занятий на текущую неделю, включая возможность перехода к предыдущим и последующим временным периодам в рамках текущего семестра. Помимо периодических лекционных и практических занятий, в личном кабинете можно посмотреть расписание экзаменов, консультаций, неперiodических мероприятий - факультативов, мастер-классов, перенесенных занятий. Дополнительно реализован функционал просмотра расписания преподавателей и дисциплин.

Модуль «*Учебный план*» содержит раздел «График учебного процесса», который информирует о продолжительности и последовательности частей учебного процесса, сгруппированных по семестрам и курсам в рамках периода обучения.

Модуль «*Список дисциплин*» находится информация о дисциплинах которые изучаются в каждом семестре студентом.

Модуль «*Электронные учебные курсы*» содержит электронную информацию по изучаемым предметам.

Модуль «*Успеваемость*» отображает итоговые оценки студентов по дисциплинам учебного плана. Для каждой дисциплины определяются части итоговой оценки: типы контроля, набранное студентом количество баллов по каждому из типов контроля, результирующее количество баллов и итоговая оценка по дисциплине. Модуль «*Успеваемость*», так же, содержит информацию об академической задолженности студентов.

Модуль «*Документы*» содержит все шаблоны документов, договоров и др.

Студенты, обучающиеся на контрактной основе, а также студенты, проживающие в общежитии, будут иметь возможность получить:

- информацию об оплате обучения (оплата за каждый год обучения студента);
- информацию об оплате общежития (оплата за каждый месяц проживания в общежитии студента);
- информацию об оплате за дополнительные образовательные услуги.

Модуль «*Обсуждения*» позволит студентам задавать интересующие вопросы, касающийся организации учебного процесса и студенческой жизни, и оперативно получить ответ на них от сотрудников пресс-службы, проректора по учебной работе,

воспитательной работе, деканов факультетов и их заместителей, а также сотрудников других подразделений университета. Возможна будет также организация опросов администрацией университета среди студентов по наиболее интересующим их вопросам.

Студенты различных факультетов смогут добавлять друг друга в друзья. Обмениваться информацией как между собой, так и с преподавателями (отправлять друг другу сообщения, документы, изображения).

В основе разработки персонализированного рабочего пространства используется система управления содержимым с открытым исходным кодом, созданная на языке PHP. Данный проект реализован с применением ООП и построен по принципу архитектуры Model-View-Controller, что позволяет применяться его в качестве PHP фреймворка.

УДК 625.539.3

студ. В.О. Берников

Науч. рук. доц. А.П. Лашенко

(кафедра информатики и веб дизайна, БГТУ)

ОПРЕДЕЛЕНИЕ ДЕФОРМАЦИЙ ДОРОЖНЫХ ОДЕЖД НЕЖЁСТКОГО ТИПА С УЧЁТОМ РЕОЛОГИЧЕСКИХ СВОЙСТВ МАТЕРИАЛОВ

Для реализации алгоритма по определению деформаций дорожных одежд нежесткого типа с учётом реологических свойств дорожно-строительных материалов использовалась технология windows forms и язык программирования C#. На основании литературных источников было установлено, что для материалов, используемых в дорожном строительстве, с достаточной точностью для практических целей, может быть принята следующая зависимость (1):

$$En \frac{d\varepsilon}{dt} + H\varepsilon = n \frac{d\sigma}{dt} + \sigma, \quad (1)$$

где E – мгновенный модуль упругости, H – длительный модуль упругости, n – коэффициент времени релаксации, зависящий от упруго-вязких свойств материала и вида нагружения, σ – напряжение, ε – деформация.

Расчетной математической моделью дорожной одежды и земляного полотна может служить многослойное квазистатическое упруго-вязкое полупространство, на поверхность которого действует нагрузка, равномерно распределенная по площади круга. Каждый i -

слой характеризуется пятью параметрами: E_i , H_i , n_i , коэффициентом Пуассона μ_i и толщиной h_i .

В пределах каждого слоя искомая функция $\varphi_i(r, z)$ непрерывна и для любого i -го слоя может быть представлена аналитической зависимостью (2) вида:

$$\varphi_i(r, z) = \int_0^\infty \{ A + B[\alpha(\eta - 1) + 2\mu_i] + \sum_{k=2}^\infty [C_k[(1 - 2\mu_i)(1 - e^{-2\lambda_k}) + \lambda_k(1 + e^{-2\lambda_k})] + D_k[2\mu_i(1 + e^{-2\lambda_k}) - \lambda_k(1 - e^{-2\lambda_k})]] \} e^{-\alpha\eta} J_0(\rho\alpha) d\alpha, \quad (2)$$

где $\eta = z/h$; $\rho = r/h$; $\gamma_i = h_i/h$; $\lambda_k = (h_{k-1} - z)/h$; h_i – суммарная толщина слоев, лежащих выше i -го, $J_0(\rho\alpha)$ – функция Бесселя первого рода нулевого порядка. Коэффициенты A , B , C_k , D_k , входящие в (3) представляют собой неопределенные функции, зависящие от нагрузки, параметра α и времени действия нагрузки. Для определения данных функций используются следующие граничные условия:

$$\begin{cases} (a_1 - a_2)A - [a_1(1 - 2\mu_1) - a_2(1 - 2\mu_2)]B - 4a_2(1 - \mu_2)C = 0 \\ (a_1 - a_2)A + [2(1 - \mu_1)a_1 - 2(1 - \mu_2)]B - 4a_2(1 - \mu_2)D = 0 \\ A + (1 - \alpha)B + \alpha a_3 C + (a_3 - \alpha a_4)D = -P\beta h^3 \alpha^{-3} J_1(\beta\alpha) \\ A - \alpha B + (a_3 + \alpha a_4)C - \alpha a_3 D = 0 \end{cases} \quad (3)$$

Найденные коэффициенты были подставлены в (4) для нахождения просадки дорожной одежды:

$$\begin{aligned} w_i = & -\frac{1 + \mu_i}{E_i h^2} \int_0^\infty \{ A + B(2 + 2\mu_i - \eta) + \sum_{k=2}^\infty \{ C_k [(1 - 2\mu_i)(1 - e^{-2\lambda_k}) - \lambda_k(1 + e^{-2\lambda_k})] - D_k [2(1 - 2\mu_i) \\ & \times (1 + e^{-2\lambda_k}) - \lambda_k(1 - e^{-2\lambda_k})] \} \} \alpha^2 e^{-\alpha\eta} J_0(\rho\alpha) d\alpha \\ & - \frac{(1 + \mu_i)(E_i - H_i)}{E_i^2 h^2 n_i} \int_0^i \int_0^\infty \{ A + B(2 + 2\mu_i - \eta) \\ & + \sum_{k=2}^\infty \{ C_k [(1 - 2\mu_i)(1 - e^{-2\lambda_k}) - \lambda_k(1 + e^{-2\lambda_k})] \\ & - D_k [2(1 - \mu_i)(1 + e^{-2\lambda_k}) - \lambda_k(1 - e^{-2\lambda_k})] \} \} \\ & \times \alpha^2 e^{-\frac{H_i(t-\tau)}{E_i n_i} - \alpha\eta} J_0(\rho\alpha) d\alpha d\tau \end{aligned} \quad (4)$$

Была выполнена визуализация зависимостей просадки дорожной одежды от скорости груженого автопоезда (рис.1) и времени действия нагрузки (рис. 2) соответственно при помощи библиотеки ZedGraph.dll.

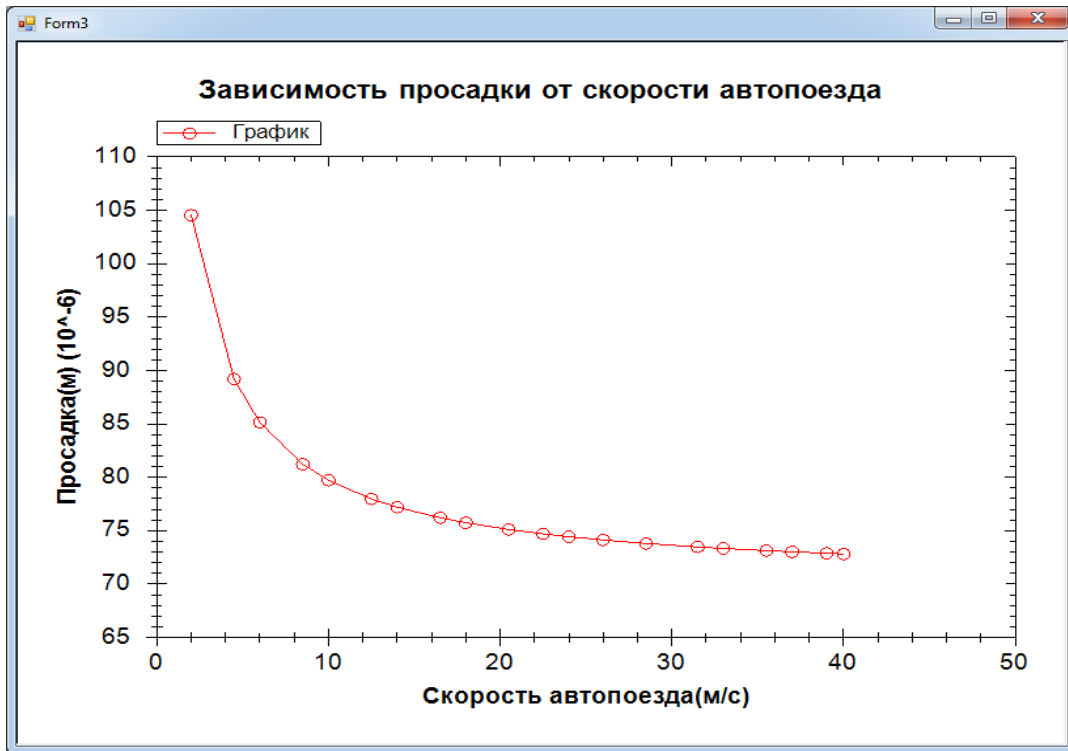


Рисунок 1 - Зависимость просадки дорожной одежды от скорости автопоезда

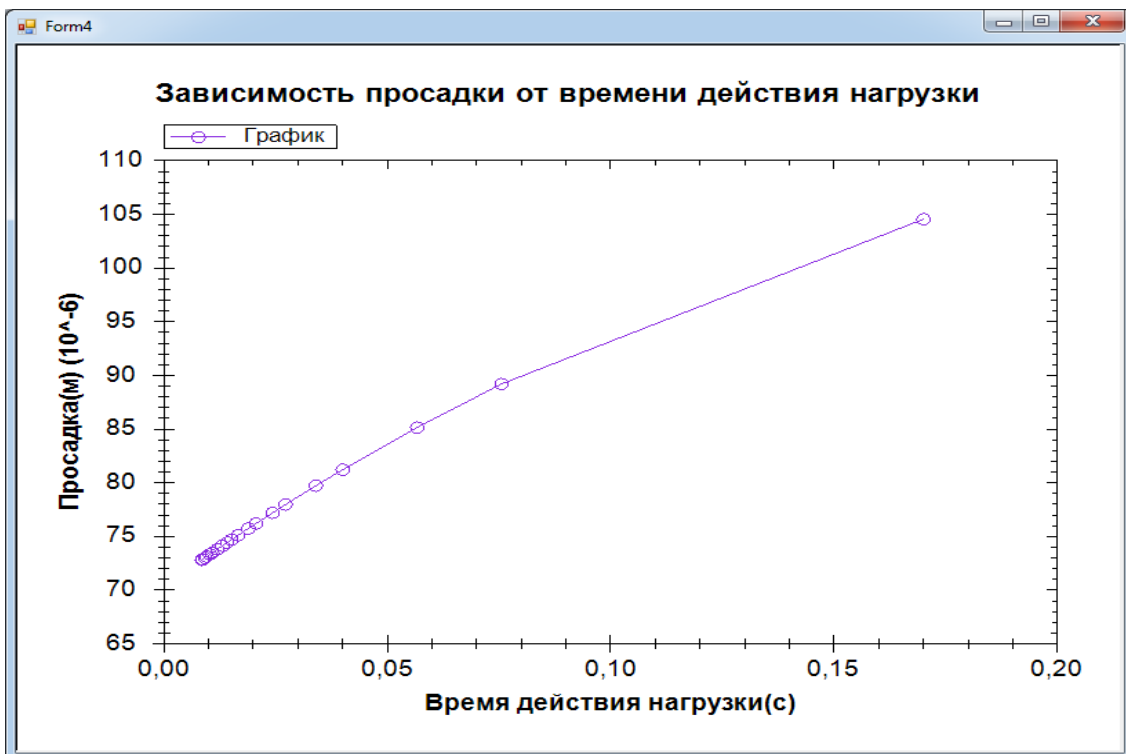


Рисунок 2 -Зависимость просадки дорожной одежды от времени действия нагрузки

ЛИТЕРАТУРА

1. Лащенко, А. П. Решение задачи определения напряжений и деформаций дорожных одежд и земляного полотна с учетом реологических свойств материалов/ А. П. Лащенко. – Минск: Рукопись, 1983. – 196 с.
2. Леонович, И. И. Применение реологических моделей к расчету дорожных одежд/ И. И. Леонович, С. С. Макаревич, А. П. Лащенко. – Минск: БелСЭ, 1971. – 183 с.
3. Лащенко, А. П. Определение деформаций слоистых систем с учетом реологических свойств материалов/ А. П. Лащенко. – Труды БГТУ, 2012, №3: Лесная и деревообрабатывающая промышленность. – 25 с.

УДК 625.539.3

студ. А.В. Фёдоров

Науч. рук. доц. А.П. Лащенко

(кафедра информатики и веб дизайна, БГТУ)

ИСПОЛЬЗОВАНИЕ ТЕОРИИ ГРАФОВ ПРИ РЕШЕНИИ ИНЖЕНЕРНЫХ ЗАДАЧ

Теория графов как один из разделов дискретной математики наряду с математическим моделированием является в настоящее время одним из интенсивно развивающихся разделов современной математики. Это связано, в первую очередь, с широким использованием компьютера, как средства решения научных и прикладных задач.

В статье рассматриваются алгоритмы, связанные нахождением решения поиска оптимального пути с использованием теории графов.

Алгоритм Флойда.

Является динамическим алгоритмом для поиска кратчайших расстояний между всеми вершинами взвешенного ориентированного графа.

$O(n^3)$

Оптимальная структура матрица смежности (весов дуг)

Недостатком данного алгоритма является высокая сложность

Алгоритм Дейкстры.

Используется для нахождения пути в графе из одной вершины до всех остальных. Алгоритм работает исключительно для графов с ребрами положительного веса.

$O(n \cdot \log(m))$

Оптимальная структура список списков дуг

Данный алгоритм применим при решении этой задачи

Алгоритм Беллмана-Форда.

В отличие от алгоритма Дейкстры, этот алгоритм применим также и к графам, содержащим рёбра отрицательного веса. При этом граф не должен содержать отрицательных циклов.

$O(n*m)$

Оптимальная структура список всех дуг

Данный алгоритм применим при решении этой задачи

Алгоритм *А.

Данный алгоритм достигает более высокой производительности за счет введения в работу алгоритма эвристической функции.

$O(\log(h(n)))$

Оптимальная структура список списков дуг

Недостатком данного алгоритма является то, что решение может быть лишь локально оптимальным.

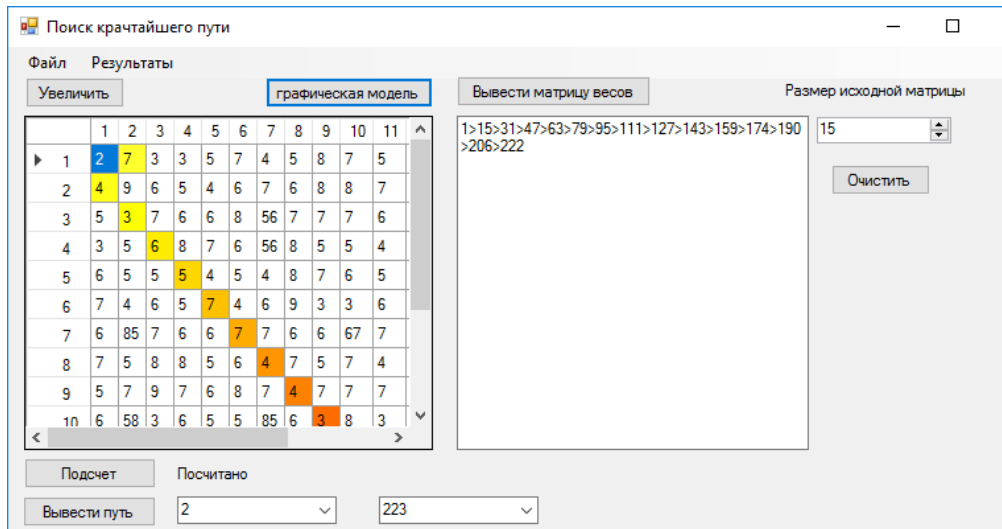


Рисунок 1 - Демонстрация работы программы

Для выделения кратчайшего пути на матрице смежности соответствующие ячейки окрашиваются в цвета от желтого к красному (Рис 1). В системе RGB красному соответствует 255,0,0, а для желтого был выбран 255, 255 ,45. Ниже приведе фрагмент кода.

```
Color[] c = new Color[a.Length];
for (int i = 0; i < a.Length; i++)
{
    int tempColor = 300 - 300 * i / (a.Length - 1);
    int green=0,blue=0;
```

```

if(tempColor<=255)
    green = tempColor;
else
{
    green = 255;
    blue = tempColor-255;
}
C[i] = Color.FromArgb(255, green, blue);//
}

```

C – массив цветов для окраски требуемых ячеек.

Color.FromArgb(255, green, blue); - функция, которая возвращает цвет, соответствующий заданным значениям в системе RGB.

Для сохранения результатов расчета используются следующие матрицы:

static short[,] costs; - матрица расстояний

static short[,] next; - матрица путей.

Поскольку вычислительные средства обрабатывают только цифровую информацию, условия строительства в каждой точке территории можно с требуемой степенью точности описать рядом чисел x , y , n_1 , n_2 , ..., n_k , где x и y координаты данной точки в прямоугольной системе координат; n_1 , n_2 , ..., n_k , конечный ряд чисел, характеризующий условия дорожного строительства в данной точке. Так как критерием оценки условий строительства служат приведенные затраты, то конечный ряд чисел представляет в общем виде экономическую функцию:

$$w = f(n_1, n_2, \dots, n_k), \quad (1)$$

В общем виде все аргументы можно сгруппировать по следующим четырем группам:

$$v = f(n_1, n_2, \dots, n_k); \quad (2)$$

$$c = f(n_{k+1}, n_2, \dots, n_i); \quad (3)$$

$$d = f(n_{i+1}, n_2, \dots, n_m); \quad (4)$$

$$k = f(n_{m+1}, n_2, \dots, n_n), \quad (5)$$

где v – функция оплачиваемых земляных работ; c – функция стоимости выполнения единицы земляных работ; d – некоторая функция, учитывающая достоверность информации об условиях

строительства и надежности принятого проектного решения в любой точке системы координат; k – функция стоимости строительства дорожной одежды. Тогда исходную функцию (1) можно представить в виде:

$$w = f(v, c, d, k) \quad (6)$$

Работа по созданию цифровой модели проводится в два этапа.

Первый этап заключается в построении инженерно-геологической карты. Работа выполняется по материалам аэрофотосъемки с использованием материалов инженерно-геологической съемки территории. Кроме того, производится накладка на исследуемую территорию изысканных и построенных трасс автомобильных дорог. Результатом первого этапа является карта категорий местности по условиям дорожного строительства.

Второй этап основан на построении цифровой модели по уже имеющейся карте категорий местности.

ЛИТЕРАТУРА

1. Бабков В. Ф., Андреев О. В. Проектирование автомобильных дорог: учебник для вузов / В. Ф. Бабков, О. В. Андреев. – М.: Транспорт, 1979. – 367 с.
2. Харари Ф. Теория графов / Ф. Харари. – М.: Мир, 1973. – 262 с.
3. В. В. Смелов, Л. С. Мороз. Алгоритмы на графах и их реализация на C++ / В. В. Смелов – Минск : БГТУ, 2011. – 96-100с.

УДК 005.745:378.66(063)(476)

Студ. У.Л. Бусько

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

ПОЛЕЗНЫЕ СОВЕТЫ ПРОЕКТИРОВЩИКАМ ИНТЕРФЕЙСОВ

Что такое «интерфейс сайта»? В переводе с английского «interface» — это средства взаимодействия. Интерфейс сайта представляет собой визуальное его отображение на экране монитора компьютера и является одним из наиболее важных элементов не только интернет-портала, но и любого приложения, программного обеспечения. Именно пользовательский интерфейс отвечает за то,

насколько удобно пользователю будет взаимодействовать с сайтом или приложением, соответственно, захочет ли он использовать этот ресурс и в дальнейшем.

Web-сайты существуют для посетителей. Посетители приходят на них ради нужной информации. Смогут ли они ее получить, зависит не только от внешних характеристик, юзабилити, а также и от интерфейса.

Изучив интерфейсы современных web-сайтов, можно смело сказать, что качество сильно варьируется.

Один и тот же интерфейс может быть как совершенно ужасным, так и приемлемым. Качество одного и того же интерфейса можно оценить по-разному в зависимости от целей посетителей.

В наши дни существует огромное количество советов по созданию пригодного, понятного, привлекающего интерфейса, как от любителей, так и от профессионалов. Не обязательно придерживаться всех советов, необходимо помнить только самые важные из них.

15 советов по созданию опрятного пользовательского интерфейса:

1) Не скрывайте основную информацию за поиском. Ваши клиенты не будут тратить время на поиск основной информации, которая должна быть видна сразу.

2) Рисуйте отзывчивый (адаптивный) дизайн. Ваш дизайн должен легко подстраиваться под разные разрешения экранов, нигде не ломаясь и не искажая основную концепцию. При создании дизайна сайта нужно продумывать и мобильную версию. Если же мобильной версии не будет, то обычно разрабатывают интерфейс мобильного приложения.

3) Сделайте удобную навигацию. Навигация должна находиться в максимально доступном и видимом месте и помогать перемещаться в нужном направлении. На сегодняшний день существует два вида меню: зафиксированная навигация, которая остается видна при скроллинге страницы; скрытая или выплывающее меню, при нажатии на соответствующую иконку.

4) Используйте большие и жирные шрифты. На маленьких экранах мелкий и тусклый шрифт трудно заметить и, уж тем более, извлечь полезную информацию. Отличным пример будет, когда мы загружаем наш сайт на мобильном телефоне. Довольно часто случается ситуация, когда сайт адаптивный, но шрифт слишком мелкий. Таким образом, очень сложно воспринимать информацию. Пользователю необходимо увеличить, чтобы разобрать, что же там

написано. Чтобы не повторять таких ошибок, необходимо выбирать соответствующий кегль и стиль шрифта.

5) Используйте простые и читабельные шрифты. Самое основное это информация и ее правильная подача. Никому не нужен ваш декоративный шрифт, в котором с трудом можно распознать буквы. Исключением будут: заголовки на странице, логотипы.

6) Если вы создаете дизайн для мобильных приложений, то он должен быть для пальцев. Вы должны понимать, что существуют люди разного телосложения. Адаптируйте все кнопки и интерактивные области вашего дизайна таким образом, чтобы человек с пальцами условно-среднего размера мог легко нажать на них.

7) Подбирайте правильную цветовую схему. Как и любой другой дизайн, дизайн интерфейсов должен притягивать внимание и, одновременно, оставаться простым и легким. Информация должна быть и привлекательной, и максимально разборчивой.

8) Поставьте себя на место вашей аудитории. Например, вы создали интерфейс для поиска недвижимости. Пройдитесь по нему и попробуйте найти себе жилье. Если какие-то элементы дизайна вам вовсе негодились — уберите их.

9) Не ставьте ссылки слишком близко друг к другу. Очень трудно попасть на ссылку, если рядом слишком близко находятся другие. Соблюдайте вертикальные и горизонтальные отступы.

10) Делайте одинаковый дизайн для сайта и его мобильной версии. Рисуйте идентичный дизайн, с идентичными элементами, как для мобильной версии, так и для десктопной версии сайта.

11) Ваш дизайн должен быть «живым». Большинство элементов вашего дизайна должны реагировать на те или иные события. Например, при клике на кнопку должна быть отдача. Человек должен понимать и иметь визуальное подтверждение того, что клик совершился.

12) Используйте быстрые кнопки связи. Если ваш бизнес локальный, используйте различные кнопки для связи с вами по телефону или почте. Клиент может не найти нужную информацию, но может быстро позвонить вам, чтобы получить то, за чем он пришел к вам.

13) Используйте изображения в нужных размерах. Не стоит использовать очень большие изображения для всех возможных разрешений экранов. Используйте точные размеры изображений для каждого экрана. Этим вы и сэкономите трафик вашей аудитории и будете показывать изображения в наилучшем качестве.

14) Привяжите ваш дизайн к эмоциям. Эмоции — одна из основных составляющих, которые заставят вашу аудиторию совершать те или иные действия. Ваш дизайн должен вызывать позитивные эмоции.

15) Постоянно держите связь с вашей аудиторией. Изучайте аналитику, тесно общайтесь с вашей аудиторией. Очень часто пользователи могут вам подсказать полезные вещи, которые упростят взаимодействие с вашим интерфейсом.

Надеюсь, эти советы помогут Вам начать мыслить как истинный дизайнер. Разработка чистого пользовательского интерфейса требует гораздо больше работы чем, кажется. Вы должны поставить себя на место пользователей и посмотреть на сайт их глазами. Это может занять месяцы тренировок, но рост креативности при этом огромен.

ЛИТЕРАТУРА

1 <https://perfecto-web.pro/ru/blog/design/20-beginner-tips-for-mobile-design.html> 20 советов дизайнеру мобильных интерфейсов

УДК 005.745:378.66(063)(476)

Студ. А.Д. Ткаченко

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

ОНБОРДИНГ – ПЕРВОЕ СВИДАНИЕ С ПОЛЬЗОВАТЕЛЕМ

В мире программного обеспечения под онбордингом понимается совокупность способов обучить пользователя-новичка -донести до него, чего и как он может добиться, используя ваше приложение. Как правило, это всевозможные туры по системе, контекстные подсказки, советы дня и т. д. Онбординг - это ваша первая встреча с пользователем. Другими словами, говорите пользователю только то, что ему необходимо знать как можно меньшим количеством слов. Следуйте простому правилу «меньше слов, больше дела». Также можно представить онбординг в виде руководства по установке вашего приложения или сайта. Ваша главная задача — расположить к себе юзера и приучить его пользоваться вашим продуктом.

Вполне логичен вопрос: когда нам нужен онбординг? В большинстве случаев, когда ваш сайт или приложение отличаются от того, к чему привыкли ваши пользователи. Это часто происходит на

мобильных девайсах - если вы вводите новое действие. Например, проводить пальцем по экрану, чтобы увидеть скрытый предмет.

Существуют различные приемы онбординга. Приём «обзорной экскурсии» показывает пользователю основные элементы вашего приложения и выделяет его главные функции. Способ хорош тем, что он сразу избавляет пользователя от первоначального недопонимания.

Недостатки: пользователь узнает, как что работает, но на самом деле он почти ничего и не сделает. Его не просят совершить действие или начать пользоваться продуктом.

Приём «сделай что-нибудь». Попросив пользователя совершить первое действие в приложении, вы сразу же вовлекаете его в процесс. Это распространено в приложениях, которые начинают работать только после совершения юзером каких-либо действий.

Недостатки: в результате пользователи могут разобраться в работе приложения не так хорошо, как нужно.

Приём «настройка». «Настройка» - это приём онбординга, который не учит юзера пользоваться продуктом, а позволяет ему настроить свой собственный аккаунт. Этот подход подразумевает, что понимание работы приложения - не самая сложная часть. А вот «настройка» - это уже настоящее слабое место, требующее поддержки.

Недостатки: В исключительных случаях пользователи всё равно могут сталкиваться с проблемами при работе с вашим сервисом. Этот приём работает при наличии понятного UI.

Приём «всё и сразу». В данном случае сразу после регистрации, пользователи получают общий обзор приложения, во время которого они могут настраивать его, а вы побуждаете совершить действие.

«Непрерывный онбординг» - это тактика, с помощью которой сайт удерживает юзера в процессе использования веб-приложения. На LinkedIn, например, есть понятный чеклист с задачами. Чаще всего они предлагают вам добавить информацию в ваш профиль и призывают «поддерживать» ваши связи. Достижение заполненного на 100% профиля является стимулом и признаком «непрерывного онбординга».

Предварительные настройки приложения. Иногда для начала работы нам надо получить от пользователя какую-то информацию. Социальные сети просят пользователей предоставить как можно больше информации о себе: демографические данные, места учебы и работы, интересы и т. д. Анализируя эту информацию, они смогут

предложить им больше релевантного контента. А значит, с большей вероятностью удержат на сайте и покажут больше рекламы.

Всплывающие подсказки. Наверное, это самый простой способ рассказать о назначении отдельных элементов вашего интерфейса. Вы можете показывать их рядом с ключевыми элементами отдельного экрана, когда пользователь впервые видит его. Обязательно давайте пользователю возможность закрыть такие подсказки.

Слои. Техника является логическим продолжением предыдущей. Новый для пользователя экран приложения мы затеняем полупрозрачной черной «занавеской» и поверх нее со стрелками, текстом и иллюстрациями рассказываем, что к чему.

Видео. Самый очевидный способ научить кого-то делать что-то – наглядно продемонстрировать весь процесс.

Несколько советов, как реализовать онбординг так, чтобы увлечь пользователя.

Используйте стартовые страницы. Пустая стартовая страница при первом запуске приложения – важная часть онбординга. Пустые экраны тонко дают юзеру понять, что ему нужно что-то сделать. Смысл сайтов и приложений заключается в их контенте. Если на странице контента нет – это еще не повод оставлять ее без внимания. Ваша задача – показать пользователю, как заполнить эту страницу и чего ждать от нее дальше. Помогите им освоиться и расскажите, как будет выглядеть страница дальше. Расскажите, про первое возможное действие – оно должно быть очевидным.

В Instagram, например, страница нового пользователя абсолютно пуста. В других профилях есть фото, комментарии и лайки, но аккаунт нового пользователя пуст – ни постов, ни фолловеров. Чтобы подтолкнуть к действию, Instagram превращает это «пустое место» в обучающую страницу: там, где должны быть фото, написано «У вас еще нет постов. Нажмите на камеру, чтобы поделиться первым фото или видео» с указателем на кнопку камеры.

Радуйтесь вместе с пользователем. Момент, когда пользователь решает какую-то задачу с помощью приложения, идеально подходит для установления эмоциональной связи между ним и продуктом. Дайте ему понять, что он классно справляется, и похвалите за победу.

Обучайте постепенно. Старайтесь делать так, чтобы пользователь больше работал с приложением, чем обучался работе с ним. Помогайте ему в трудных и неочевидных сценариях, но не заставляйте сначала выучить все функции, а только потом начинать пользоваться. Используйте «прогрессивный» онбординг. Подавайте

информацию порциями и в контексте текущей пользовательской задачи.

Не объясняйте очевидного. Не стоит объяснять пользователям, что крестик в правом верхнем углу закрывает окно, а клик на иконку с мусорным ведром приведет к удалению строки из списка. Общеупотребимые метафоры используются в интерфейсах десятилетиями, поэтому назначение этих элементов очевидно.

Расставляйте приоритеты. Не пытайтесь рассказать обо всех функциях вашего приложения в мобильных экранах онбординга и туре по системе. Выберите не более пяти ключевых возможностей и отведите для каждой отдельный слайд.

Если ваша идея сложна, сформулируйте её точнее. Поймите, какой приём сработает на вашем сайте. Если вы представляете пользователю совершенно новый концепт, преподнесите его как можно понятнее, всё равно оставаясь лаконичными. Чем дольше идёт онбординг, тем важнее показать, что завершение онбординга совсем близко.

ЛИТЕРАТУРА

1. <https://habrahabr.ru/post/238965/> Онбординг в мобильных и веб-интерфейсах. Как правильно рассказать пользователям о возможностях вашего продукта.

2. <http://uxgu.ru/onboarding-techniques/> 5 отличных способов привлечь новых пользователей.

УДК 621.357.74:628.54

Студ. К. С. Бердник

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

МЕНТАЛЬНЫЕ МОДЕЛИ КАК ОПОРА ДЛЯ ДИЗАЙНЕРСКИХ РЕШЕНИЙ

Создание веб-страницы — это процесс взаимодействия между веб-дизайнером и пользователем.

Начну с примера. Пользователь набрал в поисковой системе интересующий его вопрос и получил на него огромное количество результатов. Он переходит на понравившийся ему ресурс и попадает не на начальную страницу сайта, а в самую середину. Вот здесь и начинает работать ментальная модель. Пользователь знает, что,

кликнув по логотипу сайта, он должен попасть на главную страницу сайта. Таким образом, если пользователь кликнет по логотипу и не попадает на главную страницу, то ментальная модель ломается. Это и есть пример работы ментальной модели пользователя.

Ментальные модели формируются на основании человеческого восприятия. Это термин ввёл в 1943 году в своей книге «Природа объяснения» шотландский психолог Кеннет Крейк. Существует много определений для «ментальной карты». Ментальная модель – это интуитивное понимание принципов работы объекта или системы, основанное на прошлом опыте человека, имеющейся информации и здравом смысле. Ментальная модель – это представление человека об объекте, с которым он взаимодействует.

Как отмечает Кеннет Крейк, ментальные модели по своей природе являются неполными и постоянно развивающимися. Они никогда не дают абсолютно точное представление об объекте, но они представляют сложные явления в упрощенном виде.

Ментальная модель формируется благодаря наблюдению, погружению в ситуацию и культуры.

Наблюдение — это основной способ получения ответов на интересующие нас вопросы. С помощью органов чувств мы получаем информацию о мире вокруг, познаём объекты.

Во время наблюдения человек находит схожие характеристики окружающих его объектов, затем начинает рассматривать объекты в рамках своего представления о них. Таким образом, каждый человек создаёт свои собственные неповторимые и уникальные ментальные модели с самого рождения и по мере взросления усложняет их.

Ментальные модели непостоянны. Наблюдая за объектом в течение определенного периода времени, мы можем заметить его изменения и этим изменить существующую ментальную модель.

Погружение – это такой тип пользовательского опыта, при котором пользователь полностью поглощён взаимодействием с системой. Погружение стимулирует наши чувства и способствует выходу ощущений пользователя на другой уровень.

Опыт погружения люди получают во время увлекательной компьютерной игры с дополненной реальностью. Интерактивные объекты настолько органично вписываются в существующую у человека ментальную модель взаимодействия с реальностью, что процесс взаимодействия с системой поглощает его.

Сложно определить ментальную модель пользователя для его веб-опыта. Компании пытаются обеспечить пользователю всё более

приятный опыт взаимодействия. Для этого они постоянно добавляют новые элементы интерфейса, которые, по их мнению, заинтересуют пользователей. Вследствие чего нормы и паттерны пользователя постоянно меняются, а его пользовательские ожидания только растут.

Изменение норм пользователей для его веб-опыта — закономерный процесс. Примером служит скевоморфизм, который получил широкое распространение. Пользователи интуитивно понимали принципы работы и назначение элементов интерфейса, т.к. они выглядели как привычные объекты окружающей их среды. Скевоморфизм вышел из моды, но остались некоторые элементы, поведение которых интуитивно понятно благодаря сложившейся ментальной модели.

Культура играет важную роль в формировании человеческого восприятия. Люди, принадлежащие к различным культурам, часто имеют различные ментальные модели. Языковой барьер, социальные нормы, раскладка клавиатуры, проблемы с распознаванием символов и знаков, формат валюты и дат, единицы измерения и даже юридические требования. Все эти трудности отталкивают от сайта.

Обучение – ещё один способ формирования ментальной модели. Рассмотрим его на примере. Пользователь заходит на сайт с целью найти нужную информацию. У него уже сформировалось предположение о том, где её найти. Но изучив карту сайта, он изменил сложившуюся ментальную модель. Т.е. при помощи карты сайта научился правильно использовать веб-ресурс.

Необходимо различать концептуальную модель и ментальную модель пользователя. Люди используют свои ментальные модели для того, чтобы предсказать, спрогнозировать, как система, предмет или интерфейс будут себя вести, и для того, чтобы понять, что с ними делать. Кто-то спроектировал интерфейс, заложив в него определённую логику, и этот интерфейс сообщает вам как с ним нужно взаимодействовать.

Есть ряд популярных человеко-ориентированных подходов к проектированию интерфейсов, они помогают нам использовать ментальные модели пользователей для создания хорошего дизайна, который удовлетворит их потребности.

Первый подход – это персонажи. Алан Купер, известный первопроходец в разработке программного обеспечения, предложил когда-то концепцию персон. После опроса семи или восьми пользователей он создал персону, которая была основана на результатах опроса. Она являлась собирательным образом этой

группы пользователей. Такой подход помог объединить людей с разным менталитетом и воспринимать их как потенциальных пользователей проектируемого продукта (рисунок 1).

Персонажи:



Анастасия Кондратьева
Возраст: 30 лет.
Семейное положение: замужем, есть сын (3 лет).
Профессия: преподаватель русского языка и литературы в школе.
Группа персонажи: №4.

Анастасия очень любит своего мужа и сына. Всегда с радостью о них заботится, любит готовить и удивлять новыми блюдами. Типичная «хранительница домашнего очага».

Данный сайт посещает с целью повышения своих навыков в сфере уюта и комфорта семьи.

Задачи:

- поиск новых рецептов;
- «женские хитрости» для домашнего уюта;
- тематика здоровья.

Потребность:

- новая информация в сфере приготовления блюд, домашнего уюта и здоровья.

Потребности к сайту:

- простая поиска необходимой информации;
- подробное описание (с фото, иллюстрациями) рецептов;
- большое количество информации и идей по интересующим темам.



Александра Игоревна Степанова
Возраст: 66 лет.
Семейное положение: замужем, есть дети и внуки.
Профессия: пенсионер.
Группа персонажи: №3.

Александра Игоревна пенсионерка с большим количеством свободного времени. Любит своих детей и внуков, с радостью рассказывает их дела, подарки и сюрпризы. В свободное время любит вязать и вышивать. Как многие бабушки, вкусно готовить и любит баловать внуков выпечкой. Следит за своим здоровьем и здоровьем своей семьи. Берит в гороскопы. Активный пользователь сети Интернет и ПК.

Данный сайт необходим для:

- поиска новых шаблонов для вязания и вышивки;
- поиска новых рецептов и их обсуждения;
- вопросы здоровья.

Потребности:

- в новой информации в сфере интересующего хобби;
- обсуждение различной информации с другими пользователями интернет-портала.

Рисунок 1 - Персонажи для женского интернет-издания

Эмпатия – способность понимать и чувствовать контекст, эмоции, цели и мотивацию другого человека. Длительное погружение в роль пользователя даст понимание, которое можно использовать при проектировании. Поставив себя на место кого-то другого, можно оценить контекстные сигналы из окружающей среды, которые позволяют понять, как вещи связаны друг с другом.

Когда нужно изучить пользователей в их естественной среде, контекстные исследования являются эффективным методом для получения необходимой информации. Это метод наблюдения за пользователями в конкретной ситуации, который помогает понять их боль и способность принимать решения.

Mind mapping — это технология отображения структуры сайта, с указанием подчиненности и связей каждого элемента с другими элементами структуры. На русский язык термин переводится как «карта мыслей», «интеллект-карта», «карта памяти», «ментальная карта», «ассоциативная карта», «ассоциативная диаграмма» или «схема мышления».

Ментальная карта создается как помощь в структурировании идей по созданию сайта, так и для выражения структуры сайта.

Мой ресурс — это интернет-издание для женщин. Поэтому структура моего сайта буде именно такой, как показано на рисунке 2.

В середине карты — логический центр сайта, отправная точка. От центра отходят множество веток-разделов, каждая из которых может иметь свои веточки. Я дополнительно увеличила информативность карты введя цветовое кодирование сегментов, т.е. сгруппировала.

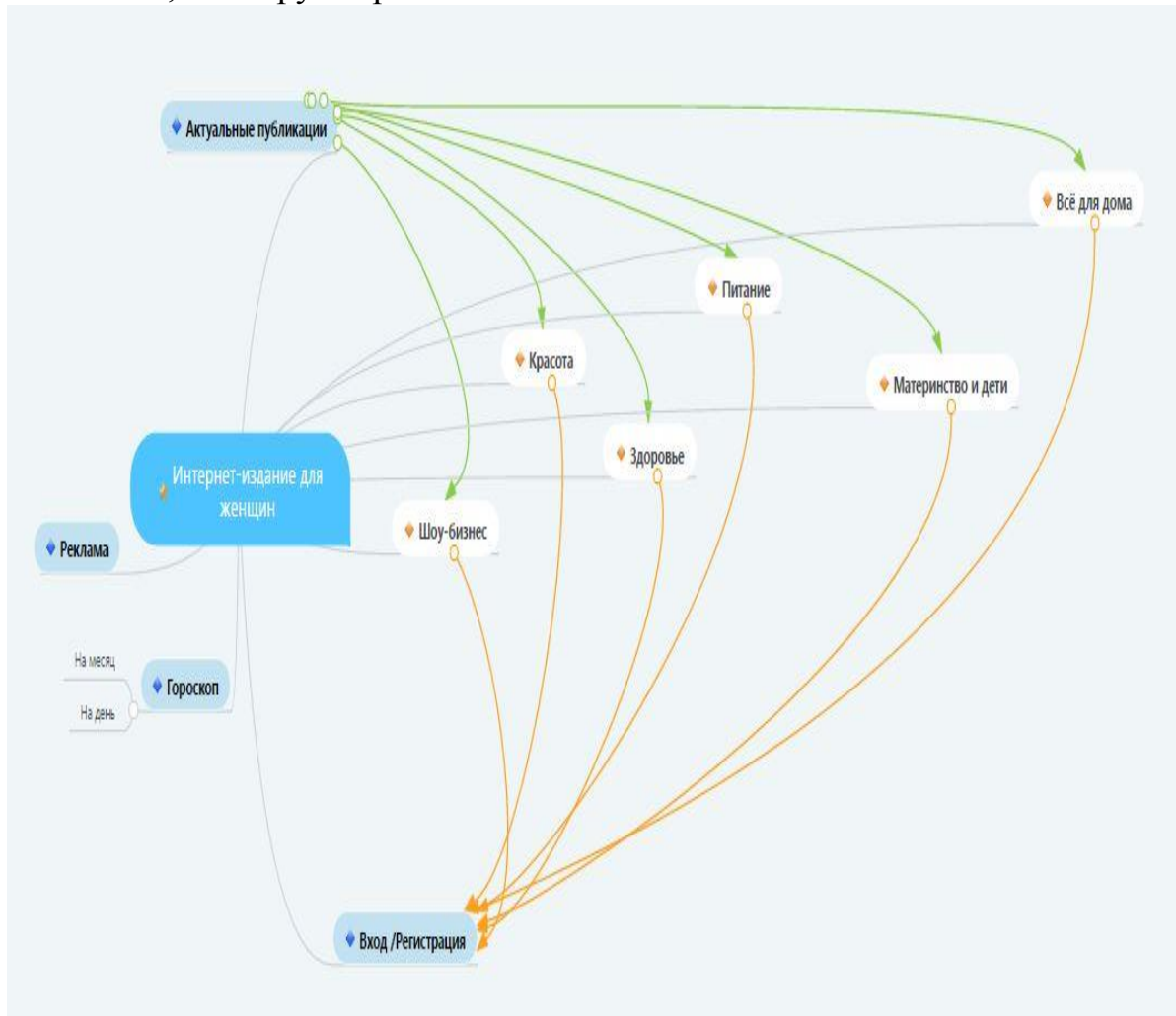


Рисунок 2 - Ментальная карта женского интернет-издания

Существует множество специальных программ, которые позволяют создавать ментальные карты. К ним относятся Concept Draw Mind Map, Mind Manager, Free Mind и другие. Инструментом для создания самой простой ментальной карты может послужить лист бумаги. Так же удобно использовать для этого Adobe Illustrator, CorelDRAW или же Adobe Photoshop.

В заключении хочу сказать, что в проектировании и создании дизайна не всегда будет возможность и финансирование на проведение исследования пользователей и на детальную проработку ментальных моделей. Однако помнить об этом обязательно.

ЛИТЕРАТУРА

1 Уэйншенк Сюзан. 100 главных принципов дизайна / Уэйншенк Сюзан. – Питер, 2013. – 272 с.

УДК 004.056

Студ. Ю. В. Ревинская, К. С. Бердник

Науч. рук. проф. П. П. Урбанович

(кафедра информационных систем и технологий, БГТУ)

РОЛЬ КРИПТОГРАФИИ В ОБЕСПЕЧЕНИИ БЕЗОПАСНОСТИ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ

Изначально криптография использовалась только для безопасного хранения или передачи документов. Сегодня область применения криптографии существенно расширилась. Основные изменения связаны с активным использованием асимметричных алгоритмов шифрования. Симметричное шифрование практикуется в основном для защиты сведений от несанкционированного доступа во время хранения.

Криптографический метод защиты самый надежный, так как охраняется непосредственно сама информация, а не доступ к ней. Процесс криптографического закрытия данных может осуществляться как программно, так и аппаратно.

Для современной криптографии характерно использование открытых алгоритмов шифрования, предполагающих использование вычислительных средств. Известно более десятка проверенных алгоритмов шифрования, которые при использовании ключа достаточной длины и корректной реализации алгоритма являются криптографически стойкими.

Сегодня на рынке представлено немало продуктов, предназначенных для безопасного хранения конфиденциальной информации с помощью криптографии. Анализ криптографических продуктов лучше всего начинать с реализованных в них алгоритмов шифрования. При этом предпочтение стоит отдавать таким, в которых реализованы известные технологии. Другой важной характеристикой является производительность. Очевидно, что процесс криптографического преобразования информации уменьшает скорость передачи и приема данных сервером и слишком длительные задержки могут привести к определенным трудностям.

Третий момент, на который необходимо обратить внимание при выборе криптографической защиты, связан с ключом шифрования. Значимы три фактора. Во-первых, длина ключа. Во-вторых, необходимо узнать, как именно осуществляется процесс создания ключей. В-третьих, способ хранения ключа шифрования.

При выборе криптографического ПО следует уделить особое внимание компании-разработчику. Практика показывает, что лучше приобретать продукты у известных фирм, которые давно присутствуют на рынке.

В связи с распространением такой информационной системы, как облачное хранилище данных, в котором информация хранится на многочисленных распределенных в сети серверах, первоочередной задачей является защита и сохранность размещенной информации.

Все существующие модели облачных вычислений подразделяется на три основных типа в зависимости от предоставляемого доступа, уровня безопасности и других возможностей: частные, общего пользования и гибридные (рисунок 1).

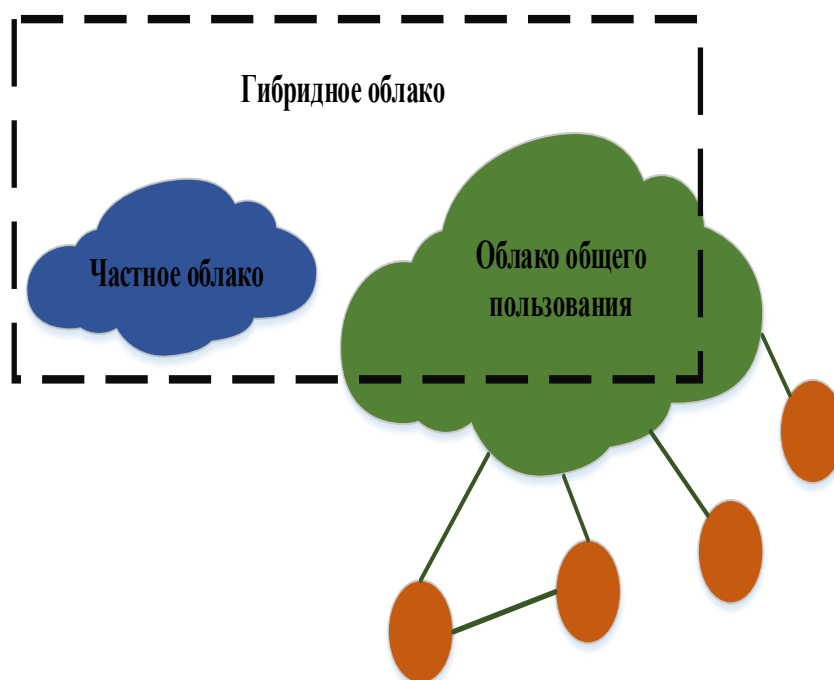


Рисунок 1 - Модели облачных вычислений

Архитектуру облачных вычислений можно разделить на сервисы интеграции (Integration as a Service, IaaS); платформы, предоставляемые в виде сервиса (Platform as a Service, PaaS);

программное обеспечение, предоставляемое в виде сервиса (Software as a Service, SaaS) [1].

На практике ключевым признаком при выборе одного из приведенных уровней облачной архитектуры является безопасность. Рынок облачных сервисов регулярно растет и расширяется, появляются, как и многие уязвимости нового уровня, так и средства защиты. Поэтому для защищаемой системы обычно определяют перечень классов угроз:

- защита периметра и разграничение сети. При использовании облачных вычислений защита более уязвимой части облака определяет общий уровень защищенности;

- динамичность виртуальных машин. Динамичность влияет на разработку целостности системы безопасности;

- уязвимости и атаки внутри виртуальной среды;

- защищенность данных и приложений. В облаке происходит процесс депериметризации. Чтобы сохранить защищенность, методы должны стать информационно-центричными;

- доступ системных администраторов к серверам и приложениям;

- несанкционированный доступ к среде виртуализации. Угроза возможна вследствие нарушения изоляции среды, предоставленной клиенту в рамках облачной услуги;

- защита бездействующих виртуальных машин;

- влияние традиционной безопасности на производительность.

На основе представленных угроз можно привести следующую классификацию основных типов атак [2]:

- традиционные атаки на ПО. Уязвимости операционных систем, модульных компонентов, сетевых протоколов, для защиты которых достаточно установить межсетевой экран, firewall, антивирус, систему предотвращения вторжений (IPS) и другие компоненты;

- функциональные атаки на элементы облака. Для каждой части облака необходимо использовать: для прокси — эффективную защиту от DoS-атак, для веб-сервера — контроль целостности страниц, для сервера приложений — экран уровня приложений, для системы управления базами данных — защиту от SQL-инъекций, для системы хранения данных — бэкапы (резервное копирование), разграничение доступа;

- атаки на клиента. Это такие атаки, как CrossSiteScripting, «угон» паролей, перехваты веб-сессий, «человек посередине» и

многие другие. Наиболее эффективной защитой является правильная аутентификация и использование шифрованного соединения (SSL) с взаимной аутентификацией;

- атаки на гипервизор. Это может привести к тому, что одна виртуальная машина сможет получить доступ к ресурсам другой;

- атака на виртуальные машины при их переносе с одного узла на другой;

- атаки на системы управления. Это может привести к появлению машин-невидимок. Решение: контроль доступа с определенных IP адресов или обязательное подключение через VPN.

Самой популярной моделью облачных вычислений являются SaaS-приложения. Однако такая модель является наименее защищенной в облачной инфраструктуре. Поэтому первоочередной проблемой, которая должна решаться при проектировании систем такого типа — безопасное хранение данных пользователей.

Для комплексного решения предложена концепция построения SaaS-систем, основанная на применении мультиарендной архитектуры с поддержкой версионности баз данных (БД), механизмом безопасной передачи информации между приложением-арендодателем и приложений-арендатором, а также управление доступом на основе токенов, ролей и привилегий.

Данные клиентов сохраняются в реляционной БД с поддержкой технологий внутреннего шифрования. При шифровании используется симметричный ключ, защищенный сертификатом, который хранится в загрузочной записи базы и доступен при ее восстановлении.

При авторизации пользователя в системе отсылается запрос приложению-арендодателю на получение строки подключения к своей БД. Для безопасной передачи данных по незащищенному каналу связи используется алгоритм, основанный на криптографическом протоколе Диффи-Хеллмана и асимметричном алгоритме шифрования RSA.

Если в рамках функционирования SaaS-системы необходимо сохранять какую-либо информацию в облачном хранилище, то эти данные шифруются с помощью поточного шифра, например, RC4.

Данная концепция универсальна и может быть основой SaaS-системы любой тематики. Защита информации является не разовым мероприятием, а непрерывным процессом. Повышение производительности вычислительной техники и появление новых видов атак на шифры ведет к понижению стойкости известных

криптографических алгоритмов. Таким образом, используемые криптографические средства должны постоянно обновляться.

ЛИТЕРАТУРА

2 Макаров С.А. Облачные вычисления / С.А. Макаров. – Москва: LAP LambertAcademicPublishing, 2012. – 104 с.

3 Безопасность облачных вычислений [Электронный ресурс] / PCmag. – Режим доступа: <http://q99.it/nMAhKZo>. – Дата доступа 10.04.2017.

УДК 004.921

Студ. А.А Козловский

Науч. рук. проф. П.П.Урбанович

(кафедра информационных систем и технологий, БГТУ)

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ СБОРА, АНАЛИЗА И ИСПОЛЬЗОВАНИЯ ДАННЫХ В ИНФОРМАЦИОННОЙ СИСТЕМЕ НА ОСНОВЕ СВЕТОДИОДНЫХ ДИСПЛЕЕВ

В связи с развитием информационных технологий, распространением беспроводного интернета и его доступности, для бизнеса наступила веха развития реализаций бизнес процессов, а также контроль их качества.

Так как у пользователей быстро растут требования к предлагаемым им сервисам, критически важными стали инструменты, позволяющие контролировать предлагаемый продукт, а также уменьшать время отклика от клиента к производителю и наоборот. Наиболее заинтересован в этом бизнес, построенный по модели SaaS – программное обеспечение как услуга, и производства, напрямую связанные с этим бизнесом [1].

С учетом изложенного, предложено решение, позволяющее значительно упростить управление, контроль, а также взаимодействие с клиентом при работе со светодиодными дисплеями. Данные дисплеи в основном используются в рекламных целях. Как правило, это один большой монитор, собранный из большого количества маленьких панелей, подключенных к специальному контроллеру, посылающему им необходимые сигналы (рисунок 1).



Рисунок 1 – Вид светодиодных дисплеев

Основная сложность эксплуатации подобных систем заключается в своевременном определении каких-либо неисправностей и, как следствие – продолжительное нефункционирование всего дисплея.

Как правило, данные дисплеи арендуются у компаний, размещающих их по всему миру.

В ходе патентного поиска для дисплеев был выбран прототип устройства из патентной базы Российской Федерации. Патент № 2316139 на изобретение «Светодиодный дисплей» имеет индекс МПК H04N9/12. Изобретение относится к аппаратным средствам персонального компьютера и может быть использовано в качестве плоско панельного дисплея в мониторе персонального компьютера и в телевизоре.

Основное преимущество светодиодных экранов, обеспечивающее их все более широкое распространение – длительный срок службы (до 100 тыс. часов, или более 10 лет непрерывной работы), высокая надежность, при необходимости – исключительно высокая яркость и всепогодность, что позволяет использовать их в уличных условиях при солнечном свете.

Светодиодный экран относится к классу активных экранов, т.е. поверхность экрана является как модулятором, так и источником света, в отличие от проекторов, видеокубов и других устройств, в которых источник света отделен от экрана. При прямом попадании солнечного света изображение светодиодного экрана не теряет контрастность, особенно при использовании серых фильтров, ослабляющих внешний свет.

Светодиодные дисплеи – излучающие приборы с некоторым количеством светодиодов, помещенных в общий влагозащищенный и

свет изолированный корпус. Один кластер соответствует одному пикселю изображения. Технология используется для изготовления больших уличных экранов [2].

Для разработки программного обеспечения управления дисплеем была выбрана платформа .NET. Используемая платформа Microsoft .NET имеет множество встроенных технологий для интеграции информационных систем и приложений, таких как службы Web, WCF, XML и другие, при этом ASP.NET существует как ее часть. Наличие таких многочисленных решений дает возможность выбора оптимальной технологии для каждого отдельного случая. Это обеспечивает отменную производительность, масштабируемость и, самое главное – безопасность. В качестве базы данных была использована СУБД PostgreSQL. Также были использованы такие технологии как JavaScript, HTML 5.0, ASP.NET MVC 5, jquery, google maps API, OSM API.

Окно созданного веб-приложения представлено на рисунке 2.

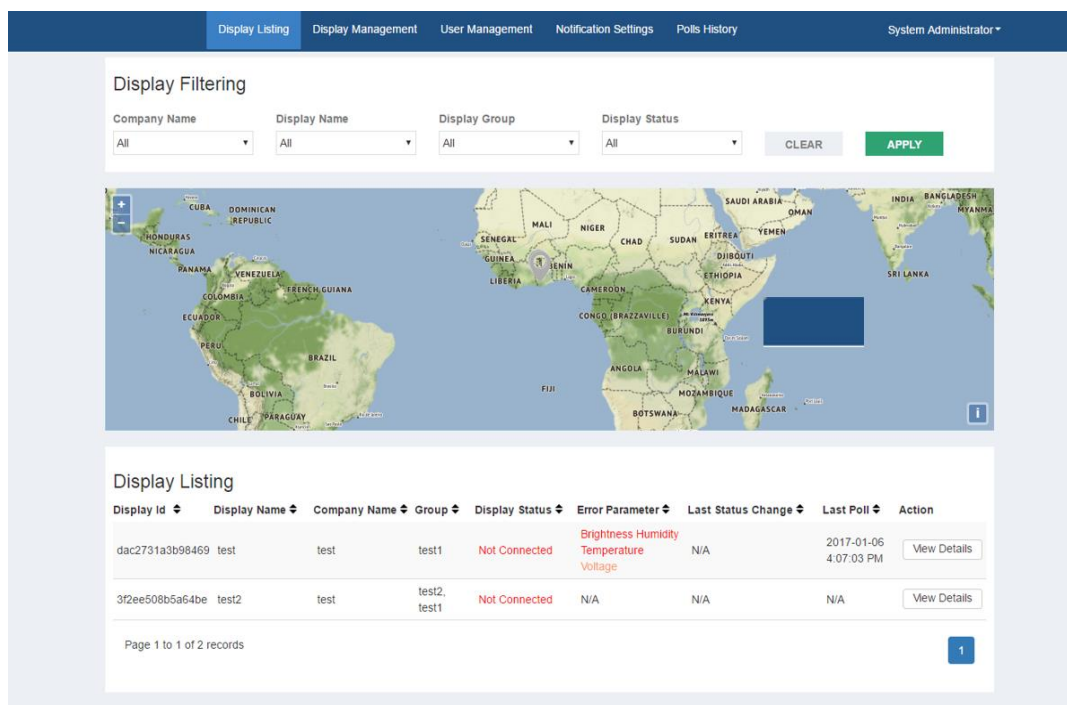


Рисунок 2 – Окно веб-портала для мониторинга дисплеев

Основной функционал веб-портала:

- авторизация и аутентификация, роли администраторов и клиентов,
- мониторинг дисплеев,

- создание и управление виртуальными дисплеями,
- создание групп дисплеев для более удобного управления,
- отправка уведомлений в случае неполадок,
- настройка параметров уведомления,
- просмотр истории дисплеев.

ЛИТЕРАТУРА

1. Wikipedia. Электронная энциклопедия [Электронный ресурс]: SAAS модель. – Режим доступа: <https://ru.wikipedia.org>. – Дата доступа: 23.04.2017.

2. ГОСТ Р 52870-2007 «Средства отображения информации коллективного пользования. Требования к визуальному отображению информации и способы измерения».

УДК 003.26

Студ. Е. С. Котик; студ. В. С. Хворост

Науч. рук. проф. П.П. Урбанович

(кафедра информационных систем и технологий, БГТУ)

КРИПТОВАЛЮТА БИТКОИН

Биткоин — это цифровая валюта, которую создали, чтобы решить все проблемы онлайн-платежей [1]. Может показаться, что существующая система и так хороша, но все наши нынешние покупки проходят через банки, которые берут себе долю транзакций и опираются на наше доверие их компетентности. Многие пытались придумать платёжную систему без такого посредника. Но есть проблема: как доказать, что ты оплатил покупку? Или что у тебя вообще есть нужная сумма, если нет поручителя? Проблема настолько серьёзная, что у неё даже есть название — проблема двойной траты.

В 2008 году анонимный программист, который представился как Сатоши Накамото нашёл решение этой проблемы. С. Накамото разместил на популярном криптографическом блоге статью с описанием платёжной системы, являющейся валютой. Он предложил, чтобы вместо банка или кредитной компании, которые записывают все транзакции в одну общую книгу, нужно чтобы все пользователи записывали все транзакции одновременно. В результате любая попытка обмануть сеть будет замечена и платёж будет отменен. Ни один конкретный пользователь, государственный аппарат или банк не могут требовать комиссию за платежи или управлять их движением. Получается более дешёвый, быстрый и простой метод

тратить деньги даже через государственные границы. Спустя несколько месяцев за биткоины уже продавали товары, их стали принимать интернет-магазины, за них стало возможным купить еду и даже оплатить учёбу в университете. Однако есть и проблемы. В то время, как одни подключились рано и получают прибыль, иные прогорают на этом молодом непостоянном рынке. Также люди открывают компании и скупают массы биткоинов, но поскольку по своей задумке биткоинов в обороте будет ограниченное количество, в будущем это может вызвать проблемы.

В своей основе биткоин — это цифровой файл, в котором записаны имена и балансы, как в регистре. Каждый компьютер в сети биткоин видит копию этого файла. Эти числа не имеют ничего физического, они ценны лишь потому, что люди готовы обменивать реальные товары и услуги ради большего числа напротив своего имени и уверены, что того же хотят остальные. Отправляя деньги, пользователь сообщает в сеть сумму, на которую уменьшится его баланс и увеличится баланс получателя, узлы или компьютеры в сети биткоин копируют эту транзакцию в свои регистры и передают транзакции следующим узлам. Если дополнить это цифровой защитой, получится система, позволяющая группе компьютеров вести регистр. И хотя по описанию она похожа на регистр, который ведет банк, тот факт, что регистр ведет группа, а не один владелец, создает ряд отличий. Во-первых, в отличие от банков, в которых каждый знает только о своих транзакциях, в биткоине все знают обо всех транзакциях. Также банку можно доверять или, как минимум, его «засудить», в случае махинаций. В биткоине же кругом незнакомцы, поэтому доверять нельзя никому. К счастью, система биткоина продумана таким образом, что доверие и не нужно. Особые математические функции защищают каждый аспект системы. Система биткоин работает следующим образом.

На базовом уровне, отправляя деньги Ивану, Алина распространяет сообщение с именами и суммой: «Отправить Ивану 5 биткоинов от Алины». Каждый узел, получая сообщение, обновляет свою копию регистра и передает сообщение о транзакции дальше.

Но откуда узлам знать, что этот запрос настоящий? Что реальный владелец счета отправил это сообщение?

По правилам биткоина, чтобы передать средства, нужен своеобразный пароль. Он называется *цифровой подписью*. Как и обычная подпись, цифровая заверяет сообщение, но делает это с помощью математического алгоритма, который предотвращает

копирование и подделку в цифровом пространстве. В противоположность постоянному паролю для каждой транзакции необходима новая уникальная цифровая подпись. Раскрывать пароль, который может скопировать и использовать кто-то чужой, крайне нежелательно. Цифровая подпись работает на основе 2 разных, но связанных ключей: приватного, который создает подпись, и публичного, с помощью которого её проверяют другие люди. Можно провести аналогию, что приватный ключ – это истинный пароль, а подпись – это посредник, который заверяет, что вы знаете пароль без необходимости его раскрывать. Публичный ключ служит адресом для получения биткоинов, т.е. отправляя человеку деньги пользователь отправляет их на его публичный ключ. Чтобы потратить деньги, необходимо доказать, что он реальный владелец публичного ключа, на который деньги отправлялись. Для этого генерируется цифровая подпись сообщения о транзакции и его приватного ключа.. С помощью привязанных к цифровой подписи вычислений, люди могут убедиться, что отправитель владеет приватным ключом, не видя сам приватный ключ. Важным аспектом является тот факт, что подпись зависит от сообщения, т.е. она новая для каждой транзакции и поэтому вор не сможет использовать её для другой транзакции. Эта зависимость от сообщения также означает, что никто не сможет изменить сообщение, передавая его по сети, поскольку изменения в сообщении сделают недействительной подпись.

Самые необыкновенные и интересные факты о биткоинах [2]:

С. Накамото — псевдоним создателя биткоинов, весь мир теряется в догадках о его настоящей личности. Сатоши создал биткоин в 2008 году. Предпринималось большое количество попыток раскрыть личность Сатоши, но пока без результатов. За 5 лет цена биткоинов выросла с нуля до 1000 долларов, а каждый день возникает примерно 3600 новых биткоинов. Монеты появляются в результате процесса, называемого майнингом.

В 2140 году будет добыт последний биткоин. 21 млн – именно таково максимальное количество биткоинов, которые будут когда-либо добыты. На сегодняшний день добыто около 12 млн. Алгоритм добычи уменьшает количество найденных монет в 2 раза каждые несколько лет, поэтому процесс неравномерен.

Житель Великобритании по имени Джеймс Хоулс по неосторожности выбросил жёсткий диск с ключом от кошелька, на котором находилось и до сих пор находится 7500 биткоинов, это примерно 5 млн долларов по текущему курсу

Исследователь из Норвегии по имени Кристофер Кох 2009 году купил биткоинов на сумму 27 долларов и забыл про них, а когда вспомнил, его инвестиция подорожала до 887 тысяч долларов, а покупал он те биткоины для своей дипломной работы по криптографии.

Таким образом, биткоин — это электронная валюта нового поколения, созданная по подобию золота и обеспечивающая достаточный уровень надежности и безопасности.

ЛИТЕРАТУРА

1. Bitcoin – что это такое? [Электронный ресурс] / Great-World.ru. – 2017. / Режим доступа: <http://great-world.ru/bitcoin-что-это-zarabotat>. – Дата доступа 01.04.2017.

2. Удивительные факты о "криптовалюте" Bitcoin [Электронный ресурс] / HiTech.ru. – 2017. / Режим доступа: <http://hitech.vesti.ru/news/view/id/3771/> Дата доступа: 01.04.2017.

Студ. Н. Н. Чобот

Науч. рук. проф., д. т. н. П. П. Урбанович
(кафедра информационных система и технологий, БГТУ)

АЛАН ТЬЮРИНГ И ЕГО СИСТЕМА ВЗЛОМА МАШИНЫ «ЭНИГМА»

Алан Тьюринг — математик, логик, криптограф. Знаменит исследованиями в области вычислимости, занимался расшифровкой машины «Энигмы». Тьюринг является одним из основоположников информатики [1].

Во время Второй мировой войны Тьюринг работал в Блетчли-парке — британском криптографическом центре, где возглавлял одну из 5 групп, Hut 8, занимавшихся в рамках проекта «Ультра» расшифровкой закодированных немецкой шифровальной машиной «Энигма» сообщений. Историк и ветеран Блетчли-парка однажды сказал: «Блетчли-парку был нужен исключительный талант, исключительная гениальность, и гениальность Тьюринга была именно такой».

Польские коллеги накануне Второй мировой войны пытались пробить брешь в кодировке и создать «свою криптологическую бомбу», используя ошибки немецких шифровальщиков, пробуя

полный перебор всех возможных комбинаций, что требовало просто нереальных затрат сил и времени. Но тогда немцы, узнав это, усовершенствовали «Энигму», сделав её ещё сложнее. В начале 1940 года Алан Тьюринг, основываясь на разработках поляков, предложил *более эффективный и наиболее общий способ*: перебор последовательностей символов на основе подобранного открытого текста. Достаточно было ежедневно узнавать или угадывать один небольшой отрывок из сообщения. Это было не сложно, так как немецкие военные общались между собой достаточно стереотипными фразами. Далее механическим перебором двадцати шести символов латинского алфавита было нетрудно определить точное место этого отрывка в полном зашифрованном тексте. Перебор ключей выполнялся за счёт вращения механических барабанов, сопровождавшегося звуком, похожим на тиканье часов, из-за чего «Бомба» и получила своё название. Для каждого возможного значения ключа, заданного положениями роторов (количество ключей равнялось примерно 10^{19} для сухопутной «Энигмы» и 10^{22} для шифровальных машин, используемых в подводных лодках), «Бомба» выполняла сверку с известным открытым текстом, выполнявшуюся автоматически.

Turing Bombe — электронно-механическая машина для расшифровки кода «Энигмы». Главной целью этого изобретения было нахождение ежедневных настроек машины «Энигма» на различных немецких военных соединениях: в частности, позиции роторов. Позиции роторов определяют ключ зашифрованного сообщения.

Первая Bombe была запущена 18 марта 1940 года. Дизайн «Бомб» Тьюринга также был основан на дизайне одноимённой машины Реевского. Машина Turing Bombe состояла из 108 вращающихся электромагнитных барабанов и ряда других вспомогательных блоков. Она была 3,0 м длиной, 2,1 м высотой, 0,61 м шириной и весила 2,5 тонны. Серийно выпускалась до сентября 1944 года, когда ход войны сделал ненужным увеличение их количества.

Всего в «Блетчли-Парке» было установлено 210 машин типа Bombe, что позволило ежедневно расшифровывать до 3 тысяч сообщений. Это внесло существенный вклад в военные усилия Британии, в особенности в борьбу с подводными лодками в Атлантике. В числе полученной информации были и сведения о подготовке вторжения в СССР.

Принцип работы «криптологической бомбы». Расшифровать сообщения немецкой машины «Энигма» возможно лишь в том случае, если известен ключ, то есть положение роторов. Для того, чтобы его узнать, «Бомба» повторяет действия нескольких соединённых вместе машин «Энигма». Стандартная «Энигма» имеет три ротора, каждый из которых может быть установлен в любую из 26 позиций. Машина «Бомба» есть эквивалент 26 машинам «Энигма», каждая из которых состоит из трёх барабанов. «Бомба» может одновременно работать над тремя ключами секретных сообщений. В отличие от роторов «Энигмы», машина «Бомба» имеет барабаны с входными и выходными контактами. Таким образом, они могут быть соединены последовательно. Каждый барабан имел 104 проволочные щётки, которые касались пластин, на которые они были загружены. Щётки и соответствующий набор контактов на пластине были организованы в четырёх концентрических кругах из 26. Внешняя пара кругов была эквивалентна току, проходящему через «Энигму» в одном направлении, в то время как внутренняя пара была эквивалентом тока, проходящего в противоположном направлении.

Пример разгадывания исходного сообщения. Прогноз погоды всегда начинался со слов: WETTERVORHERSAGEBISKAYA

Допустим, что шифротекст выглядит таким образом: ...QFZWRWIVTYRESXBFOGKUNQBAISEZ...

Для того, чтобы узнать соответствие букв, необходимо сопоставить эти тексты таким образом, чтобы буква не шифровалась сама в себя:

Q F Z W R W I V T Y R E S X B F O G K U N Q B A I S E Z
W E T T E R V O R H E R S A G E B I S K A Y A

Буква S шифруется сама в себя:

Q F Z W R W I V T Y R E S X B F O G K U H Q B A I S E Z
W E T T E R V O R H E R S A G E B I S K A Y A

Таким образом, если тексты правильно сопоставлены, то получается, что R расшифровывается как W на первой позиции и так далее:

R W I V T Y R E S X B F O G K U H Q B A I S E
W E T T E R V O R H E R S A G E B I S K A Y A

Вывод. Машина, созданная на основе этой спецификации, выполняла следующие последовательные действия:

1. Искала возможные настройки, использованные для шифрования сообщений (порядок роторов, положение ротора, соединения коммутационной панели), опираясь на известный открытый текст.

2. Для каждой возможной настройки ротора (у которого было 10^{19} состояний или 10^{22} в модификации, использовавшейся на подводных лодках) машина производила ряд логических предположений, основываясь на открытом тексте (его содержании и структуре).

3. Определяла противоречие, отбрасывала набор параметров и переходила к следующему. Таким образом, бóльшая часть возможных наборов отсеивалась и для тщательного анализа оставалось всего несколько вариантов.

А.Тьюринг занимался также разработкой шифров для переписки Черчилля и Рузвельта, проведя период с ноября 1942 года по март 1943 года в США.

ЛИТЕРАТУРА

1 Свободная энциклопедия «Википедия» [Электронный ресурс] / Тьюринг, Алан. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%A2%D1%8C%D1%8E%D1%80%D0%B8%D0%BD%D0%B3,%D0%90%D0%BB%D0%B0%D0%BD>. — Дата доступа: 01.04.2017

УДК 003.26+347.78

Студ. И.А. Лесняк
Науч. рук. проф. П.П.Урбанович
(кафедра информационных систем и технологий, БГТУ)

АНАЛИЗ ЦЕЛОСТНОСТИ ТЕКСТОВЫХ ФАЙЛОВ-СТЕГАНОКОНТЕЙНЕРОВ С ОСАЖДЕННОЙ ИНФОРМАЦИЕЙ НА ОСНОВЕ ПРИМЕНЕНИЯ КЕРНИНГА ПОСЛЕ СТРУКТУРНЫХ ИЗМЕНЕНИЙ СТЕГАНОКОНТЕЙНЕРА

Цифровая стеганография – направление классической стеганографии, основанное на сокрытии или внедрении дополнительной информации в цифровые объекты, вызывая при этом некоторые искажения этих объектов.

Целью данной работы является создание программного средства позволяющего производить стегопреобразование, модифицируя кернинг и проанализировать устойчивость стеганоконтейнера к конвертации. Кернинг – избирательное изменение интервала между буквами в зависимости от их формы [1] (рисунок 1).



Рисунок 1 – Демонстрация текста при использовании кернинга

Office Open XML — серия форматов файлов для хранения электронных документов пакетов офисных приложений — в частности, Microsoft Office. Формат представляет собой zip-архив, содержащий текст в виде XML, графику и другие данные, которые могут быть переведены в последовательность битов (сериализованы) с применением защищённых патентами двоичных форматов, спецификации которых были опубликованы Microsoft для пользователей OOXML на условиях Microsoft Open Specification Promise [2].

Для разработки была выбрана платформа .NET. Данный выбор обусловлен необходимостью тонкой работы с форматом OpenXML. Кроме того, данная платформа имеет хорошую производительность и документацию.

Как видно на рисунке 2, при открытии файла, который будет выступать в роли стеганоконтейнера, происходит поиск символов, пригодных для модификации, используя в качестве параметра кернинг.

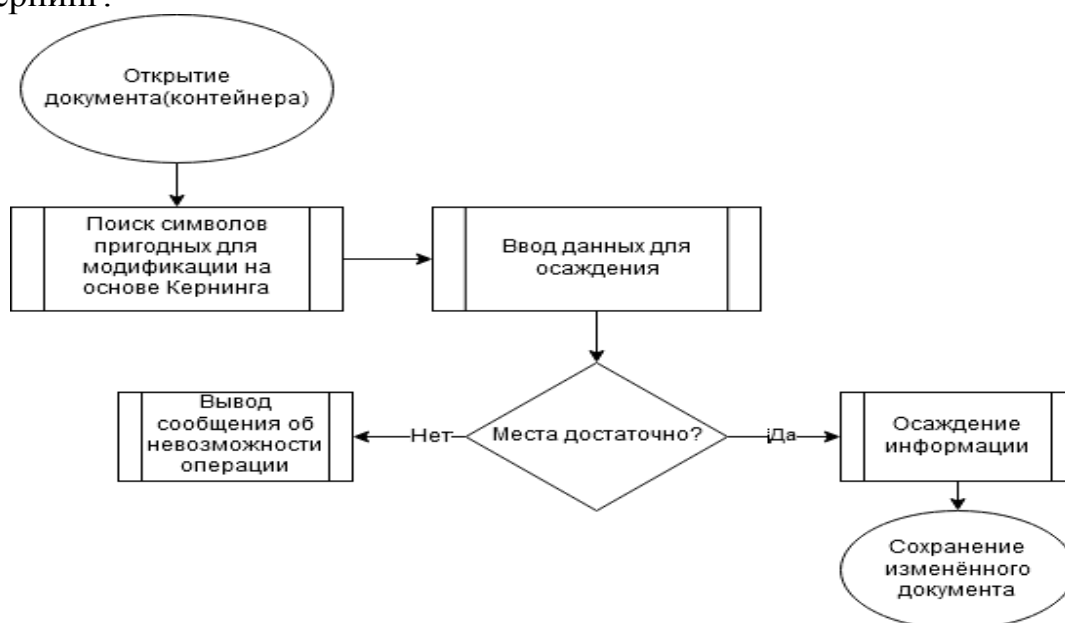


Рисунок 2 – Общая схема процесса осаждения информации

На деле это не совсем так. Дело в том, что каждый символ-буква, пригоден для осаждения в нём информации, используя кернинг. А будет ли изменяться внешний вид буквы, зависит от используемого шрифта и текстового процессора. Мы же, по сути, меняем размер шрифта, при котором будет применяться кернинг. Этот факт можно использовать для повышения секретности. Например, если задать параметр кернинга в значение 70 пт или выше, то без специального анализа обнаружить факт сокрытия информации не удастся, так как внешне текст изменяться не будет. Поэтому на данном этапе производится поиск всех символов-букв.

На рисунке 3 показано окно, в котором данные параметры, используя программу «Microsoft Word» можно задавать вручную.

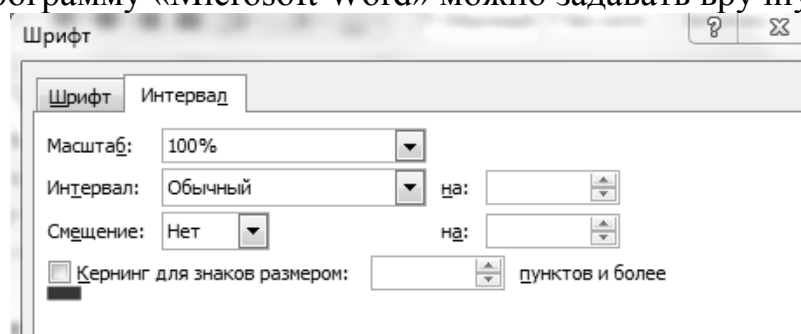


Рисунок 3 – Окно для задания значения кернинга в Microsoft Word

На рисунке 4 представлено главное окно программного средства. На данный момент оно позволяет вводить скрываемый текст вручную или загрузкой из файла. Использовать код Хэмминга перед осаждением информации, тут стоит отметить, что размер информации, которую мы можем скрыть, значительно уменьшается. Так же пользователь может сам задавать значение размера шрифта, при котором будет визуально заметно, использование кернинга. Важно отметить, что при обратном стегопреобразовании нужно будет указывать такой же параметр кернинга. И извлекать скрытую информацию.

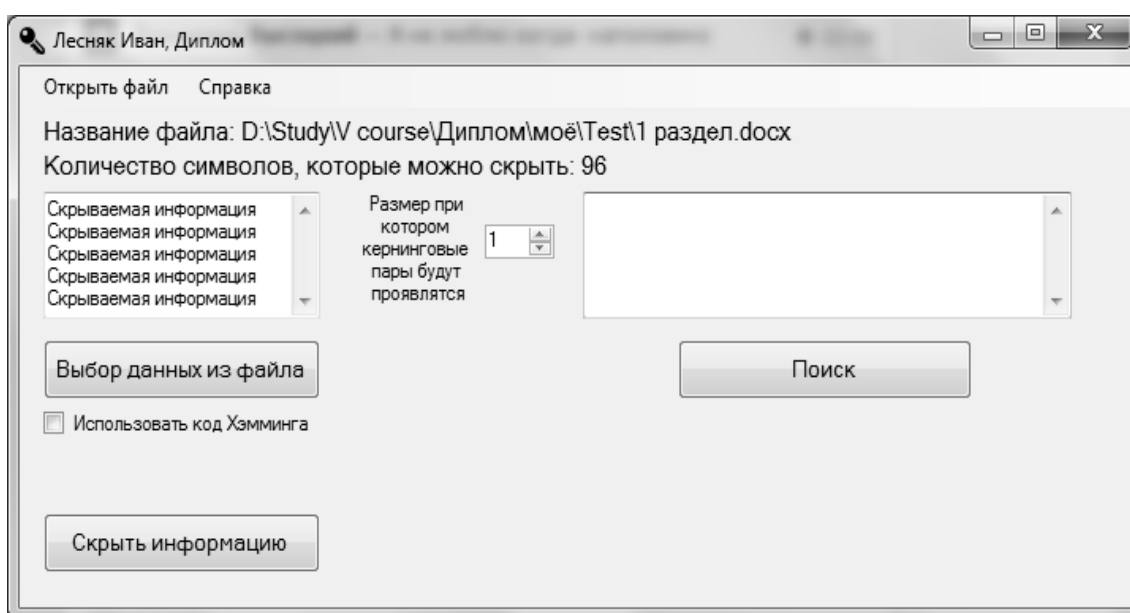


Рисунок 4 – Главное окно программного средства

Можно наверняка сказать, что при конвертации документа в формат, который поддерживает форматирование текста, потери информации не происходит. Но при конвертации в форматы, не поддерживающие форматирования, информация полностью теряется.

Таким образом, скрытие информации методами стеганографии на основе модификации кернинга является достаточно надёжным способом для защиты информации, который имеет свои плюсы и недостатки. К плюсам можно отнести гибкую настройку параметров осаждения, возможность абсолютной невидимости для глаз человека. Среди недостатков самым главным является полная потеря данных при конвертации в формат, не поддерживающий форматирования текста, хотя это уже скорее вопрос использования данного файла.

ЛИТЕРАТУРА

1. Wikipedia. Электронная энциклопедия [Электронный ресурс]: Кернинг. – Режим доступа: <https://ru.wikipedia.org>. – Дата доступа: 23.04.2017.
2. Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. — К.: МК-Пресс, 2006. — 288 с.

УДК 003.26+347.78

Студ. О.Л. Панченко

Науч. рук. проф. П.П. Урбанович

(кафедра информационных систем и технологий, БГТУ)

АНАЛИЗ УСТОЙЧИВОСТИ ПРОЦЕССА СИНХРОНИЗАЦИИ ДВУХ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧАХ ГЕНЕРАЦИИ КРИПТОГРАФИЧЕСКОГО КЛЮЧА

Для безопасной передачи данных применяются наиболее популярная технология – криптографическое преобразование информации. Криптографические системы подразделяются на асимметричные и симметричные. Их особенности заключаются в том, что в первом случае для зашифрования и расшифрования используются разные ключи, а во втором – один.

Симметричным системам свойственна проблема распределения ключей. Одни из способов решения данной проблемы является использование нейронных сетей. Данная идея была предложена И. Кантером и В. Кинцелем. В ее основе лежит использование архитектуры ТРМ (англ. Tree Parity Machine, древовидная машина четности). Возможно достижение состояния так называемой синхронизации нейронных сетей, под которой следует понимать равенство значений весовых коэффициентов нейронных сетей. Это и служит основой для протокола выработки общего ключа.

Нейронные сети представляют собой математическую модель, построенную на основе принципов работы биологических нейросетей. Нейрон представляет собой единицу обработки информации в нейронной сети. В этой модели можно выделить три основных элемента (рис. 1).

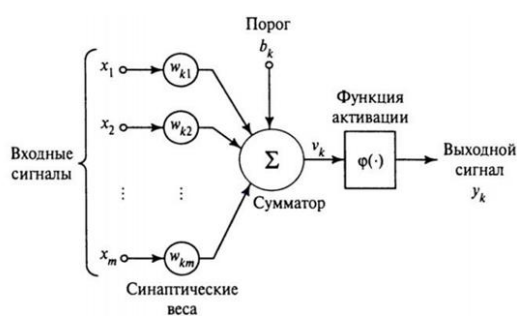


Рисунок 1 – Модель нейрона

Набор синапсов или связей, каждый из которых характеризуется своим весом. В частности, сигнал на входе синапса j , связанного с нейроном k , умножается на вес w_{kj} . Сумматор складывает входные сигналы, взвешенные относительно соответствующих синапсов нейрона. Эту операцию можно описать как линейную комбинацию.

Функция активации ограничивает амплитуду выходного сигнала нейрона. Эта функция также называется функцией сжатия. Обычно нормализованный диапазон амплитуд выхода нейрона лежит в интервале $[0, 1]$ или $[-1, 1]$.

Исходя из этого, нейронная сеть должна состоять из нейронов. Такую самую простую сеть с прямой связью составляет одиночный нейронный слой. Первым такую сеть предложил Фрэнк Розенблатт и назвал её одиночный перцептрон. В таком слое каждый нейрон получает одинаковый набор входных сигналов, и каждый из них имеет свой собственный вектор.

Весовые коэффициенты существенным образом влияют на результаты работы сети, поскольку они решают, правильно ли сеть действует. Представленная модель нейронной сети в принципе бесполезна, так как она не способна решать поставленные задачи до тех пор, пока не будет соответствующим способом «натренирована». Этот процесс называется обучением и основан на соответствующем подборе коэффициентов вектора весов в контексте решаемой задачи.

Обучение сети может быть реализовано двумя основными методами:

- обучение с учителем. Учитель дает сети правильный ответ – какое выходное значение она должна дать на представленный входной сигнал;
- обучение без учителя. Цель обучения не определена в виде конкретных образцов, так что, используя полученные данные, сеть учится сама [1].

Взаимное обучение сетей А и Б касается простейшей однонаправленной модели, в которой участвуют два персептрона. Роли таких нейронных сетей заранее не определены, каждая может выполнять функции как учителя, так и ученика. Это значит, что сети учатся друг у друга, используя для этого полученные результаты и, стремясь к общей цели, находят общие элементы в результате своих вычислений [2].

Эта модель обучения, благодаря своим свойствам, может быть использована в криптографии, а именно: в определении общего для А и Б криптографического ключа. Именно эта модель используется в приложении. Процесс синхронизации сетей происходит следующим образом:

1. Задаются случайные значения весовых коэффициентов;
2. Затем выполняются следующие шаги, пока не наступит синхронизация;
3. Генерируется случайный входной вектор X;
4. Вычисляются значения скрытых нейронов;
5. Вычисляются значение выходного нейрона;
6. Сравниваются выходы двух ТРМ:
7. Если Выходы разные: переход к п.3;
8. Если Выходы одинаковые: применяется выбранное правило к весовым коэффициентам.

После полной синхронизации, сети А и Б могут использовать веса в качестве ключа [3].

Нами разработать программное средство для проведения анализа устойчивости процесса синхронизации двух нейронных сетей в задачах генерации криптографического ключа.

В приложении пользователь вводит такие параметры как (рис. 2):

- количество входных нейронов у персептрона – N;
- количество персептронов – K;
- диапазон изменения L.

Количество входных нейронов у персептрона
N

Количество персептронов
K

Диапазон изменения
L

Весовые коэффициенты

Wа Wб

Рисунок 2 – Окно программного средства

В результате пользователю выводятся весовые коэффициенты каждой сети, что позволяет проверить результат. Проведя достаточное количество экспериментов, можно произвести анализ устойчивости двух нейронных сетей в процессе синхронизации.

ЛИТЕРАТУРА

1. Харин Ю.С. Математические основы криптологии / Ю.С. Харин, В.И. Берник, Г.В. Матвеев. — Минск: БГУ, 1999. — 319 с.
2. Плонковски, М. Д. Криптографическое преобразование информации на основе нейросетевых технологий / М. Д. Плонковски, П. П. Урбанович // Труды БГТУ. Сер. VI. Физико-математические науки и информатика. – Минск: БГТУ, 2005. – С. 161–164.
3. Плонковски, М. Д. Синхронизация криптографических ключей на основе нейросетевых технологий / М. Д. Плонковски, П. П. Урбанович // Материалы междунар. науч.-практ. конф., апрель 2006г. / Брест. гос. ун-т им. А. С. Пушкина. – Брест: Изд-во БрГУ. – С. 29.

УДК 004.056+347.78

Студ. А. А. Сущеня

Науч. рук. проф. П. П. Урбанович

(кафедра информационных систем и технологий, БГТУ)

СТЕГАНОГРАФИЧЕСКОЕ ПРЕОБРАЗОВАНИЕ ТЕКСТОВ-КОНТЕЙНЕРОВ НА ОСНОВЕ ЯЗЫКОВ РАЗМЕТКИ

Во все времена существовала необходимость передавать информацию, но в XXI веке этот вопрос стоит на первом месте. Зачастую передаваемая информация имеет строго конфиденциальный характер и нуждается в тайной пересылке. С задачей тайной передачи «справляется» криптография, делая невозможным чтение передаваемой информации без знания ключа. Но есть другой подход к защите пересылаемых данных. Можно не шифровать информацию, а делать ее «невидимой». Для этих целей используют стеганографию. Стеганография – это наука и искусство передавать сокрытые данные, внутри других, не сокрытых данных. Скрываемую информацию называют стегосообщением, а данные, внутри которых располагается стегосообщение, называют контейнером. Основным преимуществом стеганографии над криптографией является то, что само наличие скрытой информации определить затруднительно.

Цифровая стеганография — направление стеганографии, основанное на внедрении дополнительной информации в цифровые объекты. В качестве контейнеров цифровой стеганографии можно использовать изображения, видео, музыку, текст.

Метод основанный на стеганографии подразумевает, что сам контейнер, несущий в себе данные, подвергнется незначительным изменениям. Наиболее важным в данном процессе является то, что эти изменения не должны отразиться на визуальном отображении контейнера.

Роль стеганографического контейнера может выполнить html-документ. Основной акцент при встраивании информации в html-документ делается на то, что не все символы разработанного документа видны при отображении в браузере.

Известно, что при интерпретации html-документа браузер не придает значения тому, какой тип кавычек используется при его создании. Следовательно, если заменить какую-нибудь пару кавычек в валидном html-документе, например, с двойной на одинарную, то при визуальном анализе документа со стороны пользователя в браузере

никакой разницы видно не будет. Используя эту технику, в html-документ можно осадить бинарную последовательность.

Перед тем как начинать осажение информации нужно убедиться в том, что контейнер имеет достаточную информационную емкость. Емкость контейнера определяется как количество пар кавычек во всем документе.

При встраивании последовательности бит условимся, что единице будет соответствовать двойная кавычка, а нулю одинарная. Начиная с первой пары кавычек в документе будем ставить ей в соответствие бит встраиваемого сообщения и, при необходимости, изменять тип кавычки на противоположный (например, первая пара кавычек в документе двойная, а первый бит осаждаемой последовательности – нулевой, следовательно, необходимо тип кавычек заменить на одинарный).

При использовании данного метода также необходимо заранее определить количество бит, отводимое под один символ сообщения. Установка количества бит позволяет не использовать впустую место в контейнере. Ведь если необходимо передать текст, состоящий только из букв английского алфавита, то для представления одного символа в двоичном виде будет вполне достаточно семи бит, в отличие от русского алфавита где для представления одного символа необходимо уже как минимум одиннадцать бит.

В конец осажденного сообщения встраивается уникальная последовательность, указывающая на то, что сообщение закончилось.

Для демонстрации вышеописанного метода создано программное средство (HTMLStego), которое позволяет осажать в html-документ текстовую информацию.

При создании программного средства использовался язык программирования C#.

В процессе создания были разработаны следующие классы (рисунок 1): HTMLFile, FileManager, Embedder, Extracter.

HTMLFile – в своем составе имеет свойство File, представляющее собой загруженный файл html, а также метод GetContainerCapacity, позволяющий в зависимости от количества бит отводимых под один символ вычислять емкость контейнера.

FileManager – предназначенный для ввода и вывода html-документа.

Embedder – предназначенный непосредственно для осажения информации. Метод MakeBinaryString конвертирует сообщение в двоичный вид. Метод EmbedMessage, последовательно проходя по

сообщению и ставя в соответствие очередную пару кавычек, изменяет ее при необходимости.

Extractor – при помощи ExtractMessage извлекает бинарную последовательность из прочитанного документа пока сообщение не закончится (проверяет IsEndOfMessage). После извлечения используя метод RestoreMessage из бинарной последовательности получаем исходное сообщение.

Для осаждения сообщения следует указать количество бит, отводимое под один символ сообщения, затем загрузить контейнер, нажав на кнопку «Открыть HTML файл» и выбрать из файловой системы html-документ (рисунок 2).

После загрузки контейнера существует возможность узнать максимально возможное количество символов, которое можно осадить в контейнер. Осаждаемое сообщение вводится в поле «Встраиваемое сообщение». Для осаждения сообщения необходимо нажать кнопку «Встроить».

Для извлечения осажденного сообщения достаточно открыть стегоконтейнер, нажав на кнопку «Извлечь сообщение» и выбрать html-документ из файловой системы.

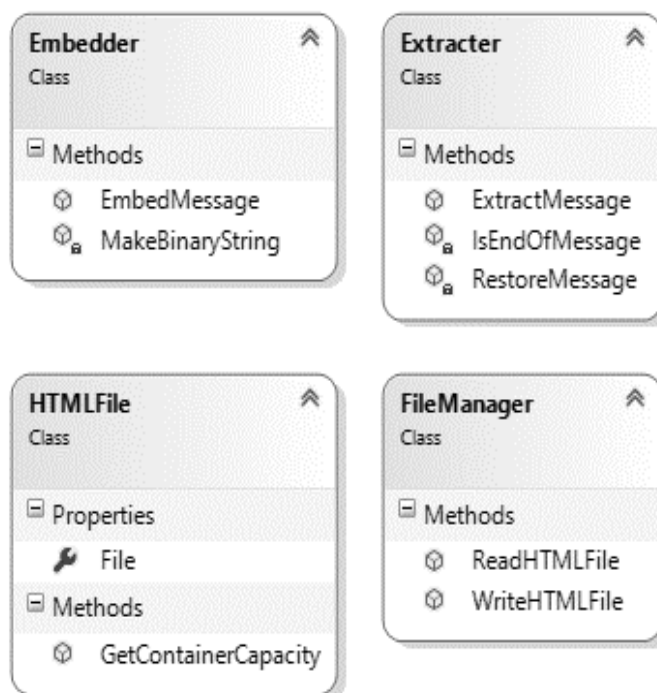


Рисунок 1 – Классы, представляющие функционал для осаждения информации

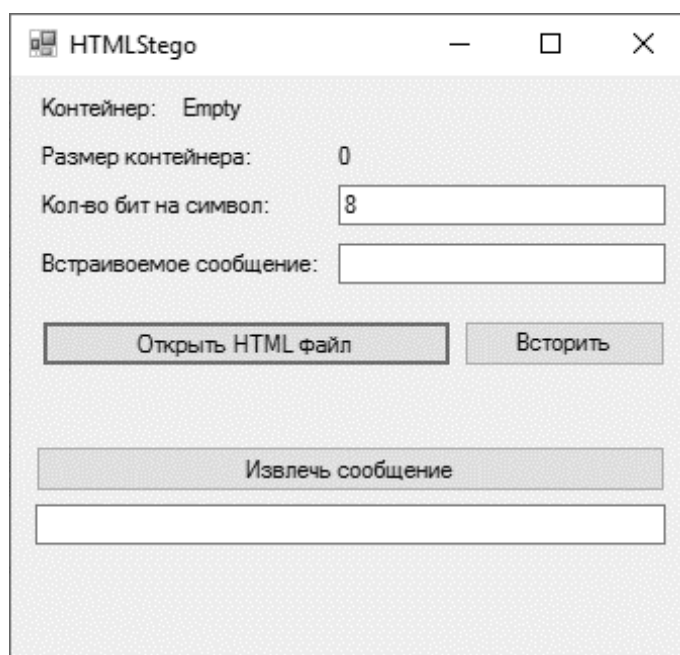


Рисунок 2 – Главное окно программного средства HTMLStego

Изложенный выше метод позволяет в документы типа html/xml встраивать текстовую информацию. Основная идея заключается в том, что при встраивании битовой последовательности каждому биту ставится в соответствие определенный тип кавычек, и при необходимости этот тип изменяется. При модификации контейнера его структура практически не изменяется.

В документе с n парами кавычек можно закодировать n бит сообщения. Выявить данный метод возможно только при просмотре кода html-документа. Подозрение может вызвать то, что зачастую при разработке сайтов используется какой-либо один тип кавычек.

ЛИТЕРАТУРА

1. Урбанович, П.П. Защита информации методами криптографии, стеганографии и обфускации/ П.П. Урбанович. – Минск : БГТУ, 2016, – 220 с.
2. Интернет-портал [Электронный ресурс]/ Стеганография & путешествия. – Режим доступа: <http://www.nestego.ru/2012/05/xml-html.html>. – Дата доступа: 20.03.1017.
3. Интернет-портал [Электронный ресурс]/ Стеганография & путешествия. – Режим доступа: http://www.nestego.ru/2012/05/blog-post_03.html. – Дата доступа: 20.03.1017.

4. Интернет-портал [Электронный ресурс]/ ДонНТУ Портал магистров. – Режим доступа: <http://masters.donntu.org/2015/fknt/sipakov/library/article6.htm>. – Дата доступа: 5.04.2017.

УДК 004.056

студ. А. А. Чопик

Науч. рук. проф. П. П. Урбанович
(кафедра информационных систем и технологий, БГТУ)

СТЕГАНОГРАФИЧЕСКАЯ СТОЙКОСТЬ ИЗОБРАЖЕНИЯ-КОНТЕЙНЕРА К ОБЪЕМУ ОСАЖДАЕМОГО ТЕКСТОВОГО СООБЩЕНИЯ

Одной из основных причин популярности исследований в области стеганографии в настоящее время является проблема защиты авторских прав и интеллектуальной собственности, представленной в цифровом виде [1-2]. При помощи методов стеганографии эта проблема решается следующим образом: есть тайное сообщение и контейнер – любая информация, используемая для сокрытия тайного сообщения. При помощи секретного ключа (стегоключа) отправитель встраивает сообщение в контейнер и получает стего. Получатель при помощи ключа может извлечь из стего тайное сообщение или, например, определить автора (рисунок 1).



Рисунок 1 - Стеганографическая система

Не менее важным является изучение вопросов заметности внедренных стеганограмм, потому что это непосредственно влияет на степень сохранности (стойкости) внедренной информации.

Целью исследования было изучить влияние объема осаждаемой в изображение-контейнер информации на заметность стеганограммы.

Для исследования разработано программное средство (рабочее название «STG 1.0»), реализующее один из стеганографических методов – метод LSB (Least Significant Bit, наименьший значащий

бит). Суть состоит в замене последних значащих бит в файле-контейнере (изображении) на биты скрываемого сообщения. На рисунке 2 приведен вид основного диалогового окна средства.

Данное программное средство позволяет осаждать информацию в определенное количество последних бит каждого байта изображения, (на основе модели RGB), а также извлекать эту информацию обратно в текстовый файл.



Рисунок 2 –Основное диалоговое окно программного средства «STG 1.0»

В качестве контейнера выбрано изображение в формате BMP, содержащее и мелкие детали, так и крупные части одного цвета. В качестве осаждаемой информации использовался отрывок произведения Historia Langobardorum, письменность – латиница.

Были проведены различные эксперименты с размером осаждаемой информации, количеством осаждаемых бит в изображении, использование различных комбинаций текстовой информации.

Результаты показывают, что при осаждении малого количества информации (по отношению к размеру осаждаемого изображения) или при осаждении небольшого количества бит (1-2) практически невозможно обнаружить видимых различий в изображении, если не использовать специализированных средств. Так как

стеганографическая стойкость определяется, в основном, эффективностью визуального анализа, то в данном случае можно говорить о высокой стеганографической стойкости.

При увеличении объема осаждаемой в изображение информации факт отличия от оригинала был очевиден. Так при осаждении 4-5 бит изображения можно видеть вполне различимое ухудшение качества (четкости) изображения. Изменяется насыщенность и резкость цвета. Наблюдалось уменьшение яркости изображения. Данное количество бит использовать для передачи какие-либо данные крайне опасно, так как такие изменения в изображении легко могут скомпрометировать информацию.

При еще большем увеличении объема осаждаемой в изображение информации, появлялись значительные изменения, детали заднего фона становились почти неразличимыми, изображение полностью теряло свои исходные черты.

В таблице 1 приведена статистика количества внедренной информации в зависимости от используемого количества бит каждого байта изображения.

Таблица 1 - Объем внедрённой информации при использовании различного количества бит

Количество используемых бит контейнера	Внедрено знаков (без пробелов)	Внедрено знаков (с пробелами)	Внедрено слов
1	18726	21674	2992
2	37413	43740	6026
3	56133	65743	9027
4	74635	87051	12055

На основе этого можно сделать вывод, что стеганографическая стойкость изображения-контейнера уменьшается при увеличении объема осаждаемой информации. Но даже если использовать для внедрения лишь небольшое количество последних бит изображения (1 или 2), то можно внедрить довольно большой объем информации и при этом подобное изображение будет обладать высокой стеганографической стойкостью.

ЛИТЕРАТУРА

1. Грибунин, В. Г. Цифровая стеганография / В. Г. Грибунин, И.Н. Оков, И. В. Туринцев. - М. : СОЛОН-Пресс, 2002. – 230 с.

2. Шутько, Н. П. Математическая модель системы текстовой стеганографии на основе модификации пространственных и цветовых параметров символов текста / Н. П. Шутько, Д. М. Романенко, П. П. Урбанович // Труды БГТУ. Сер. VI: Физ.-мат. науки и информатика. – 2015. – № 6. — С. 152–156.

УДК 003.26+347.78

Студ. А. Н. Щербакова
Науч. рук. проф. П. П. Урбанович
(кафедра информационных систем и технологий, БГТУ)

КЛОД ШЕННОН – ОСНОВОПОЛОЖНИК ЦИФРОВЫХ ТЕХНОЛОГИЙ И ТЕОРИИ БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

Клод Элвуд Шеннон (1916-2001 гг.) – американский инженер и математик. Человек, которого называют отцом современных теорий информации и связи.

Теория информации неразрывно связана с опубликованной в 1948 году статьей К. Шеннона «Математическая теория связи». Фактически данная работа предопределила путь, по которому с тех пор развивается теория информации. В своей книге Шеннон изложил способ, как количественно характеризовать сигнал. Для этого он использовал величину, которая называется количеством информации, или энтропией.

Энтропия означает мера неопределённости или непредсказуемости информации. Это количество информации, которое приходится на одно элементарное сообщение источника. Энтропия применяется во многих системах. Рассмотрим применение энтропии для анализа сложности пароля. Энтропия пароля (сложность пароля, измеряемая в битах), сгенерированного случайным образом, находится по следующей формуле:

$$H = L \frac{\ln N}{\ln 2}, \quad (1)$$

где L – набор символов в пароле, N – количество символов в используемом алфавите.

Вывод состоит в том, что чем меньше сложность пароля, измеренная в битах, тем легче его взломать методом полного перебора.

Множество работ Шеннона связаны с секретностью и стойкостью пароля. Рассмотрим понятие «расстояние единственности». Если криптографическая система не является совершенно секретной, то зашифрованное сообщение может дать криптоаналитику некоторую информацию об исходном сообщении. Существует некоторая длина перехваченного сообщения, после которой сообщение может быть дешифровано с вероятностью, близкой к единице. Шеннон ввел понятие расстояния единственности шифра (или расстояния уникальности) U , которое показывает, сколько букв зашифрованного сообщения необходимо перехватить для однозначного восстановления ключа. Для вычисления расстояния единственности необходимо знать энтропию ключа $H(K)$. Если известна энтропия ключа $H(K)$ для некоторого шифра, то расстояние единственности U для него вычисляется по формуле:

$$U = \frac{H(K)}{D}, \quad (2)$$

где D – избыточность шифруемого сообщения (бит).

К. Шеннон был одним из первых, кто развеял миф о «безопасности через сокрытие». До этого пытались сохранить в тайне именно алгоритмы шифрования в надежде на то, что не зная деталей реализации, враг не сможет перехватывать зашифрованные сообщения.

Шеннон сформулировал принцип «враг знает систему». Это значит, что нет смысла прятать алгоритм, а даже наоборот, следует открыть его для всех желающих. Но что-то скрываемое все равно должно быть, ведь шифр по определению – это сообщение, прочитать которое в идеале может лишь один человек – обладатель секрета шифра. При подходе «враг знает систему» подобным секретом стал так называемый ключ.

В своей исторической работе «Теория связи в секретных системах» К.Шеннон доказал, что одноразовый гамма-блокнот является «невскрываемой» шифрсистемой. Фактически Шеннон представил эту криптосистему как совершенную.

«Одноразовый блокнот» использует длинную шифрующую последовательность, которая состоит из случайно выбираемых бит или наборов бит (символов). Шифрующая последовательность побитно или посимвольно накладывается на открытый текст, имеет ту же самую длину, что и открытый текст, и может использоваться только один раз. Открытый текст сообщения m записывают как последовательность бит или символов:

$$m = m_1, m_2, m_3, \dots, m_{n-1}. \quad (3)$$

Шифрующая последовательность k той же самой длины:

$$k = k_1, k_2, k_3, \dots, k_{n-1}. \quad (4)$$

Шифртекст определяется следующим образом:

$$c_i = m_i \oplus k_i. \quad (5)$$

Чтобы получить шифртекст, необходимо выполнить операцию сложения по модулю 2 (XOR) над первым битом сообщения и первым битом в одноразовом блокноте, что дает первое значение шифртекста. Затем выполняется XOR над следующим битом сообщения и следующим битом в блокноте, что дает второе значение шифртекста. Этот процесс продолжается до тех пор, пока все сообщение не будет зашифровано.

Схема шифрования с использованием одноразового шифровального блокнота считается не взламываемой только в том случае, если в процессе ее реализации выполнены следующие условия:

1. Блокнот должен использоваться только один раз.
2. Блокнот должен существовать ровно столько же времени, что и само сообщение.
3. Блокнот должен распространяться безопасным образом и защищаться получателем.
4. Блокнот должен быть заполнен действительно случайными значениями.

К известным теоремам Шеннона можно отнести:

1. Прямая и обратная теоремы Шеннона для источника общего вида – о связи энтропии источника и средней длины сообщений.
2. Прямая и обратная теоремы Шеннона для источника без памяти.
3. Прямая и обратная теоремы Шеннона для канала с шумами – о связи пропускной способности канала и существования кода, который возможно использовать для передачи с ошибкой, стремящейся к нулю.
4. Теорема Найквиста-Шеннона об однозначном восстановлении сигнала по его дискретным отсчетам.
6. Теорема Шеннона об источнике шифрования (или теорема бесшумного шифрования).
7. Теорема Шеннона — Хартли.

Рассмотренные работы и теоремы К. Шеннона – лишь малая часть того, какой вклад он внес в развитие теории информации, криптографии и других сфер.

ЛИТЕРАТУРА

1. Совершенно секретные системы [Электронный ресурс]. Режим доступа: <http://www.intuit.ru/studies/courses/691/547/lecture/12395>. – Дата доступа: 11.04.2017
2. Информационная безопасность [Электронный ресурс]. Режим доступа: <http://dorlov.blogspot.com.by/2010/08/issp-06-1.html>. – Дата доступа: 10.04.2017

УДК 378.147

Студ. П. А. Короткая
Науч.рук.: доц., к.т.н. Н. Н. Пустовалова
доц., к.п.н. Н. П. Коровкина
(кафедра информационных систем и технологий,
кафедра автоматизации производственных процессов и электротехники БГТУ)

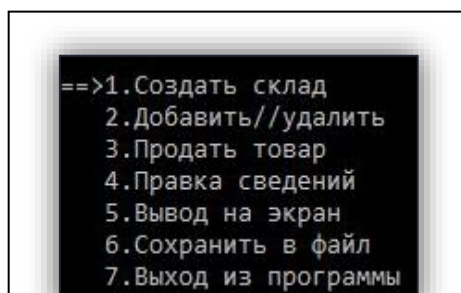
ПРОГРАММНОЕ СРЕДСТВО ДЛЯ УЧЕТА ТОВАРОВ НА ПРЕДПРИЯТИИ

В данной работе представлена программа, которая может найти практическое применение в области менеджмента и бухгалтерии, что поможет развить коммерческий потенциал организации, использующей программный продукт.

Программа написана на языке C++, что обуславливает потенциальную кроссплатформенность и расширяет круг устройств и операционных систем, поддерживающих данный программный продукт, а значит и целевую аудиторию, готовую приобрести его. Также с переносом программы на мобильные устройства увеличиваются возможности использования программы.

Рассмотрим работу программы на примере. Пусть необходимо создать склад для некоторого товара и управлять движением товара. Основное меню появляется сразу после открытия программы. Навигация по пунктам меню осуществляется при помощи клавиатурных стрелок, ввод – стандартный с клавиатуры. В первую очередь надо создать новый склад для товаров, нажав на выбранном пункте меню клавишу Enter.

После создания нового склада следует нажать клавишу «стрелка вниз» и каретка оказывается напротив пункта «Добавить//удалить».



Клавиша Enter служит для перехода в подменю добавления и удаления.

Пусть надо ввести информацию о первом товаре, нажав Enter. При этом пользователю будет предложено поэтапно ввести данные о товаре.

Данные вводятся с клавиатуры, окончание ввода подтверждается нажатием клавиши Enter, после чего всплывает следующее поле ввода. Нажатие Enter после ввода последнего поля добавляет товар в базу данных. Таким же образом добавляются данные других товаров.

Удаление товара также не составляет трудности. Для этого в подменю добавления и удаления необходимо выбрать пункт «Удалить товар» и ввести во всплывающее поле номер товара для удаления (нумерация начинается с нуля). По завершению ввода будет отображено сообщение об успешном удалении.

Следующим этапом будет добавление покупателя товаров. Поля для добавления заполняются в разделе «Добавить покупателя» подменю добавления и удаления. Поля заполняются аналогично прочим, всего необходимо заполнить 4 поля с номером продаваемого товара, названием фирмы, отпускной ценой и запрашиваемым каждый раз количеством товара. По завершению ввода будет отображено сообщение об успешном добавлении.

Аналогично добавляется информация о других покупателях.

Удаление покупателя также не требует значительных усилий и аналогично удалению товара с той лишь разницей, что нужно заполнять дополнительное поле с номером товара, который продается искомой фирме, и номер самой фирмы. По завершению ввода будет отображено сообщение об успешном удалении.

При появлении необходимости в продаже товара, нужно в главном меню выбрать пункт «Продать товар» и заполнить поле с номером продаваемого товара, номером фирмы-покупателя и количеством продаваемого товара. Процесс продажи сопровождается наглядными таблицами с имеющимися товарами и их потенциальными покупателями для упрощения ввода номера товара. По завершении будет отображена выручка в результате оферты.

В процессе добавления товаров со временем неизбежно возникают ошибки. Ликвидации ошибок способствует пункт «Правка сведений» главного меню программы (рис.1). Принцип использования аналогичен добавлению товара с той лишь разницей, что вначале необходимо ввести номер изменяемого товара, а лишь затем новые сведения о нем. По завершении редактирования будет отображено сообщение об успехе.

Для организации информации в программе используются стеки. Стек – это абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in – first out, «последним пришёл – первым вышел»).

Несмотря на изначально образовательные цели разработки данной программы, она имеет некоторые преимущества среди ей подобных, а именно:

- гибкость кода с возможностью добавления неограниченного числа необходимых товаров/услуг. Эта особенность сыграет на пользу небольшим, но динамически развивающимся компаниям, руководство которых заинтересовано в быстром росте уровня продаж без необходимости замены программного обеспечения в процессе;

- возможность вывода необходимой информации в файл и дальнейшей обработки файла специализированными программами бухгалтерского учета. При этом отпадает необходимость переучивания и переквалификации кадров для использования представленной программы;

- обратная совместимость с уже существующими ассортиментами товаров и, как следствие, возможность расширения ассортимента без необходимости очистки;

- относительная простота консольного интерфейса и, соответственно, легкость в освоении;

- небольшой размер и высокий уровень оптимизации под любую категорию устройств, что делает возможным перенос программы на мобильные платформы, характерно отличающиеся небольшими вычислительными ресурсами..

Основную целевую аудиторию программы составляют лица и/или организации, так или иначе участвующие в товарно-денежных отношениях в качестве поставщика товаров и/или услуг. Она будет иметь спрос среди юридических лиц, начиная от индивидуальных предпринимателей и заканчивая крупными торговыми организациями, в специализацию которых входит, в том числе, торговля.

Студ. Н.В. Рак

Науч. рук. доц. Н.А. Жилияк

(кафедра информационных систем и технологий, БГТУ)

ИНТЕРНЕТ - ОПИУМ XXI ВЕКА

Опиум — сильнодействующий наркотик, получаемый из высушенного на солнце млечного сока, добываемого из незрелых коробочек опийного мака.

Как он работает. После наслаждения от наркотика человек становится заторможенным. Больной совершенно не способен адекватно воспринимать мир. Постепенно умственные способности человека начинают ухудшаться. Вы замечаете, что он не может нормально поддерживать разговор.

Британская Ост-индская компания (БОИК) монополизировала промышленное производство опиума в Бенгалии, бывшей частью британской колонии Индии. Именно там производился самый высококачественный опиум. Членами и акционерами БОИК стали первые лица Британской империи — лорды-пэры. Именно они и начали формировать в Китае наркоцивилизацию.

Первоначально компания учредила «Китайскую внутреннюю миссию», задачей которой было пристрастить к опиуму китайских крестьян с помощью пропаганды опиокурения. Это создало рынок сбыта для опиума, который и был заполнен Британской Ост-индской компанией. Пропорционально ввозу опиума потребление наркотиков возрастало в Китае до огромных размеров.

К концу 1830-х годов употребление опиума превратилось в страшную разрушительную силу китайского общества. Императорский двор был напуган тем, что опиум курили даже солдаты, необходимые для подавления народа.

История Китая очень характерна как живой пример того, как уничтожают государства изнутри. Британским садистам не удалось превратить Китай в свою колонию как Индию, только потому, что в Китае была сильная центральная власть с дееспособным правительством, которое сумело обеспечить единство страны вплоть до середины XIX века. Неудача колонизации привела к испытанному веками опыту иудеев, в части разложения стран изнутри, и опиум был для этого главным инструментом.

Опиум не только приносил огромную прибыль, но и умирал, делал довольными и безопасными толпы расплодившихся нищих.

Таким образом на протяжении сотни лет Китайцы становились более зависимыми и пустоголовыми от опиума, что конечно можно

сравнить с ситуацией, которая сейчас происходит на нашей планете с интернетом. Как часто вы наблюдали за собой бессмысленное залипание в компьютере, затем переключение на телефон и социальные сети? С чего вы начинаете свой день. Для многих людей стало привычкой начать его с проверки социальных сетей и постов в инстаграме. Почти половину времени бодрствования, не учитывая детство, отнимают у нас телевизор, телефон и компьютер – 20 лет. Пугающие цифры. На что бы вы потратили это время, начав всё сначала. Сейчас у вас есть время задуматься над этим, в старости вы уже ничего не измените. Вычислить первые признаки интернет-зависимости не сложно:

- вы ощущаете огромную радость перед каждым новым сеансом, а без доступа в интернет испытываете грусть и уныние;

- вы постоянно обновляете страничку в соцсетях, часто проверяете e-mail почту и т.д;

- вы теряете чувство времени. Заглянув вроде бы по делу на несколько минут, вы зависаете в сети на несколько часов и не замечаете, как стремительно пролетает время;

- также важно, что именно человек делает в интернете. Одно дело, когда это связано с работой, производственной необходимостью или обучением, и другое — если человек бесцельно рассматривает сайты, играет в сетевые игры, просматривает бессмысленные видеоролики ;

- обычно на фоне погружения в сеть человек отказывается от нормального, реального, личного общения, от работы и учёбы, портятся отношения с окружающими людьми, нарушаются режим сна и режим питания;

- при попытках ограничить или сократить время пользования Интернетом возникают негативные реакции: депрессия, стресс, ярость, гнев;

 - расстройства режима сна и питания;

 - использование Интернета как способа ухода от проблем, стрессов, неприятностей.

Зависимость от сети вызывает эмоциональную подавленность, трудности с концентрацией и отстранённость. В результате человек предпочитает компьютер и виртуальное общение — общению с живыми людьми.

Доктор Иван Голдберг, уверял, что чрезмерное пользование интернетом вызывает депрессию, стресс, агрессию. Сегодня известно,

что у детей и подростков зависимость от виртуального пространства вызывает задержку социального и эмоционального развития — неспособность работать и достигать результата, иметь друзей, создать семью. Вернуть человека снова в реальную жизнь иногда бывает сложно. Первым делом нужно обратиться к врачу-психотерапевту, который проанализирует проблему, исключит другие заболевания и найдёт правильный выход из ситуации. Если вы сами за собой заметили, что проводите в сети слишком много времени, или в поведении близкого вам человека появились характерные для интернет-зависимости симптомы, то можно попробовать предпринять простые профилактические действия:

- установить точное количество времени, которое можно проводить в интернете;

- иногда заставлять себя несколько дней подряд не выходить в интернет;

- заблокировать доступ к конкретным ресурсам Сети (особенно для детей);

- установить правило - не общаться в интернете в субботу и в воскресенье;

- ввести чувствительные, но не вредящие здоровью санкции за несоблюдение правил и ограничений;

- насыщать реальную жизнь разнообразными событиями (хобби, общение, физическая активность, путешествия);

- самому или с помощью специалиста выявить первопричину тяги к интернету и избавиться от нее.

«Если вы не замечаете, как проходит время за компьютером, или наоборот, замечаете, что оно «очень быстро пролетает» - обратите всё-таки внимание, сколько вы времени на это тратите и чем конкретно занимаетесь. Проведите анализ - насколько это действительно важно для вас и необходимо. Избежать зависимости, если у вас есть склонность, можно только с помощью самоконтроля», - добавила психолог Анна Хныкина. Если вы задумаетесь о том, что интернет является не главной потребностью в вашей жизни и обратите внимание на действительно ценные вещи, значит не все ещё потеряно. Значит, этот мир ещё не трещит по швам.

В результате анализа данной проблемы в процессе разработки программное средство контроля трафика посещения сети Интернет.

УДК 324

Студ. Деликатная М.М.

Науч. рук. доц. Жилияк Н. А.

(кафедра информационных систем и технологий, БГТУ)

БАЗА ДАННЫХ «СТУДЕНТЫ»

При компьютерной обработке информации упорядоченные данные принято хранить в базах данных – файлах, использование которых вместе со специальными программными средствами позволяет пользователю как просматривать необходимую информацию, так и, по мере необходимости, манипулировать ею, например, добавлять, изменять, копировать, удалять, сортировать и т.д [1]. Цель данной работы – создание базы данных для более эффективной работы деканата. В этой базе будет храниться информация о студентах и посещенных ими мероприятиях за все время обучения. Проектирование базы данных происходило в четыре этапа [2]: анализ предметной области, определение целей, выбор системы управления базами данных (СУБД) и проектирование базы данных.

Анализ предметной области. В базе данных «Студенты» будет храниться информация мероприятиях, которые посетили студенты за время обучения в университете, а также информация о самих студентах. В данной базе будет предоставлена возможность изменения и обработки информации, а также возможность выполнения запросов и получение отчетов.

Особенности базы (рисунок 1):

- каждый факультет имеет несколько специальностей;
- на каждой специальности имеется несколько групп;
- каждая группа имеет список студентов;
- каждый студент участвовал в одном или нескольких мероприятиях.

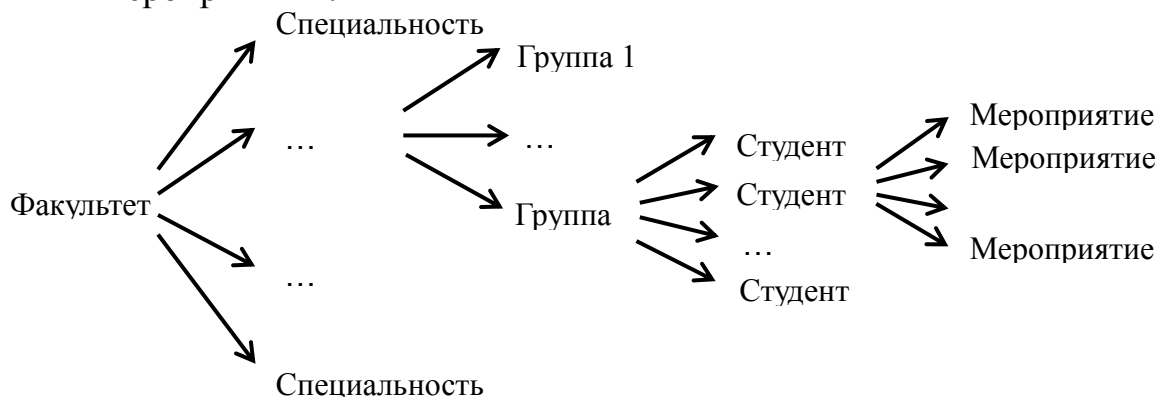


Рисунок 1 – Особенности базы

База данных является реляционной. В ней данные хранятся в таблицах, которые связаны отношениями. Каждая запись в таблице соответствует отдельной сущности, а количество строк записей в таблице неограниченно.

В базе данных «Студенты» выделено пять сущностей: *факультеты, специальности, группы, список студентов и список мероприятий*. Для каждой выбраны атрибуты и определены типы связей.

Сущность «*Список мероприятий*»:

- код мероприятия;
- название мероприятия;
- код студента.

Сущность «*Список студентов*»:

- код студента;
- фамилия;
- имя;
- отчество;
- курс;
- номер подгруппы;
- форма обучения;
- код группы.

Сущность «*Список студентов*» связана с сущностью «*Список мероприятий*» связью один ко многим.

Сущность «*Группы*»:

- аббревиатура;
- код специальности;
- код группы.

Сущность «*Группы*» связана с сущностью «*Список студентов*» связью один ко многим.

Сущность «*Специальности*»:

- аббревиатура;
- код факультета;
- код специальности.

Сущность «*Специальности*» связана с сущностью «*Группы*» связью один ко многим.

Сущность «*Факультеты*»:

- аббревиатура;
- код факультета.

Таблица «*Факультеты*» связана с таблицей «*Специальности*» связью один ко многим.

Определение целей. БД создаётся для декана факультета и преподавателей. Она должна отображать сведения о студентах и мероприятиях, которые они посетили. Также должен осуществляться поиск данных, а также реализация запросов и отчетов. Запросы должны быть направлены на получение информации о студенте и информации о посещенных им мероприятиях.

Выбор СУБД. Система управления базами данных – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных [3].

Данная база создавалась с помощью системы управления реляционными базами данных (СУРБД) *Microsoft SQL Server*. Основной используемый язык запросов, который компания *Microsoft* позиционирует как диалект языка *SQL* для своей фирменной СУРБД — *Transact-SQL* [4].

Проектирование БД. С учетом нормализации и составленных отношений, база данных будет выглядеть следующим образом, представленным на рисунке 2:

```

USE Студенты;

CREATE TABLE Факультеты
(
    "Код_факультета" INT IDENTITY(1,1) PRIMARY KEY,
    "Аббревиатура" NVARCHAR(40) NOT NULL,
);

CREATE TABLE Специальности
(
    "Код_специальности" INT IDENTITY(1, 1) PRIMARY KEY,
    "Аббревиатура" NVARCHAR(30) NOT NULL,
    "Код_факультета" INT FOREIGN KEY REFERENCES "Факультеты"(Код_факультета) NOT NULL
);

CREATE TABLE Группы
(
    "Код_группы" INT IDENTITY(1, 1) PRIMARY KEY,
    "Аббревиатура" NVARCHAR(30) NOT NULL,
    "Код_специальности" INT FOREIGN KEY REFERENCES "Специальности"(Код_специальности) NOT NULL
);

CREATE TABLE Список_студентов
(
    "Код_студента" INT IDENTITY(1, 1) PRIMARY KEY,
    "Фамилия" NVARCHAR(30) NOT NULL,
    "Имя" NVARCHAR(30) NOT NULL,
    "Отчество" NVARCHAR(30) NOT NULL,
    "Курс" NVARCHAR(30) NOT NULL,
    "Подгруппа" NVARCHAR(30) NOT NULL,
    "Форма_обучения" NVARCHAR(30) NOT NULL,
    "Код_группы" INT FOREIGN KEY REFERENCES "Группы"(Код_группы) NOT NULL
);

CREATE TABLE Список_мероприятий
(
    "Код_мероприятия" INT IDENTITY(1, 1) PRIMARY KEY,
    "Название_мероприятия" NVARCHAR(30) NOT NULL,
    "Код_студента" INT FOREIGN KEY REFERENCES "Список_студентов"(Код_студента) NOT NULL
);

```

Рисунок 2 – Проектирование БД

Таким образом, было создано пять таблиц. Все значения полей таблиц являются неделимыми, не имеется совпадающих записей. Все неключевые поля в таблицах зависят только от ключа таблицы и не зависят друг от друга. Это значит, что все таблицы БД приведены к 3 нормальной форме.

В каждой таблице имеется первичный ключ (*primary key*), который является идентификатором объекта. Каждый идентификатор – это счетчик (*identity(1,1)*), который автоматически увеличивается на единицу при добавлении в таблицу новой записи. Таблицы также включают поля с символьным типом данных (*nvarchar*). Для установления связей между данными в таблицах были созданы внешние ключи (*foreing key*). Все поля таблицы должны быть обязательно заполнены (значение *NOT NULL*).

ЛИТЕРАТУРА

1. Пинкус, М. Введение в базы данных [Электронный ресурс] / М. Пинкус. – Режим доступа: <http://onmcsso.narod.ru/db/>.
2. Карпова, И. П. Проектирование реляционных баз данных [Электронный ресурс] / И. П. Карпова. – Режим доступа: <http://rema44.ru/resurs/study/dbprj/dbprj.html>.
3. Эталонная модель управления данными: ГОСТ Р ИСО/МЭК ТО 10032-2007. Введен 27.12.07. – Москва: Стандартинформ, 2009. – с.3.
4. Ицик, Б. Г. Microsoft SQL Server 2012. Основы T-SQL / Б. Г. Ицик. – М. : Эксмо, 2015. – 400 с.

УДК 378.147

Студ. А. В. Бабако

Науч.рук.: доц., к.т.н. Н. Н. Пустовалова
(кафедра информационных систем и технологий, БГТУ),
доц., к.п.н. Н. П. Коровкина

(кафедра автоматизации производственных процессов и электротехники, БГТУ)

ОБУЧАЮЩАЯ ПРОГРАММА ДЛЯ ИЗУЧЕНИЯ ФАЙЛОВОГО ВВОДА/ВЫВОДА НА ЯЗЫКЕ C/C++

Приложение для изучения файлового ввода/вывода на языке C/C++ реализовано в виде двух этапов обучения. Первый этап представлен в виде теста, на втором этапе пользователь должен отвечать на различные вопросы по теме. Для перехода с первого этапа на следующий необходимо правильно выполнить больше 70%

заданий, в противном случае этап будет не доступен. Процент выполненных заданий отображается на главной форме (Меню). Если выполненных правильно заданий недостаточно, пользователь должен пройти тест заново.

Знание темы «Файлового ввода/вывода на языке C/C++» является фундаментом работы с информацией. Обмен информацией между программой и внешними устройствами осуществляется с помощью операций ввода/вывода. Под вводом/выводом в программировании понимается процесс обмена информацией между оперативной памятью и внешними устройствами: клавиатурой, дисплеем, магнитными накопителями и т. п. Ввод – это занесение информации с внешних устройств в оперативную память, а вывод – вынос информации из оперативной памяти на внешние устройства. Такие устройства, как дисплей и принтер, предназначены только для вывода данных; клавиатура – устройство ввода информации. Магнитные накопители (диски, ленты) используются как для ввода, так и для вывода.

Основным понятием, связанным с информацией на внешних устройствах компьютеров, является понятие файла. Всякая операция ввода-вывода трактуется как операция обмена с файлами: ввод – это чтение из файла в оперативную память; вывод – запись информации из оперативной памяти в файл. Поэтому вопрос об организации в языке программирования ввода-вывода сводится к вопросу об организации работы с файлами.

Для изучения темы в окно формы загружается html-документ с краткой информацией по работе с файлами, разбитый на логические блоки: создание потока, связь с файлом; открытие файла, режимы доступа; обмен информацией.

Подробно рассматривается организация записи информации в файл сначала на языке C++, а именно: вставка в поток (операция >>); посимвольная запись; метод `fl.write(const char* buf, type num)`, где `fl` – поток, связанный с файлом, первый аргумент – записываемая информация, второй – размер.

Аналогично с записью рассматривается считывание информации из файла: извлечение из потока (операция <<); посимвольное считывание; метод `read(char* buf, type num)` для работы со структурами, массивами.

При открытии файла, курсор устанавливается в начале либо в конце файла. Для работы с информацией необходимо иметь быстрый

доступ в любую позицию файла, поэтому рассматриваются способы использования функций позиционирования.

Закончив работу с файлом, первое, что необходимо сделать – отсоединить поток от файла, разорвать связь посредством функции `close()`.

После ознакомления с теоретическим материалом пользователю предоставляется возможность закрепить его и проверить, насколько хорошо усвоен материал, пройдя этап, включающий, как и запись ответа в маскированное поле, так и тестовые задания (рис. 1).

Чем характеризуется данный режим открытия файла? ios::ate

- Установить указатель позиционир. фай-ла на конец
- начать поиск с конца файла;
- Открыть файл для дозаписи в его конец

Параметр origin определён как константа в классе ios, поиск от текущей позиции в файле

- ios::beg
- ios::cur
- ios::end

Для чтения строки из файла используется функция (), которая является функцией класса ifstream.

Функция () считывает символ из файла. Функция () записывает символ в файл.

Continue Exit

Рисунок 1 – Окно для проверки знаний студентов

Поскольку в настоящее время для работы с файлами используется также и язык С, то с помощью данного обучающего программного средства изучаются способы обмена данными на этом языке.

При этом последовательность действий и их изучение при вводе/выводе сохраняется:

- создается специальная структура с именем FILE;
- открывается файл с помощью функции `fopen()`, где прописывается имя файла и метод открытия файла, который позволяет предусмотреть ситуацию, существует ли файл и установка курсора;
- работа с информацией осуществляется посимвольно, по строкам, по блокам;

– позиционирование, осуществляется посредством параметров: SEEK_SET, SEET_CUR, SEEK_END;

– в конце работы с файлом «отвязывается» структура FILE от файла.

Приложение реализовано на языке С# с привязкой html-документов и ориентировано на студентов, изучающих язык программирования С/С++. Оно помогает освоить теоретические основы, которые являются фундаментом практики.

УДК 625.539.3

Студ. И.Д. Сафончик

Науч. рук. доц. Н.В. Пацей

(кафедра информационных систем и технологий, БГТУ)

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ УПРАВЛЕНИЯ
ЕВРОПЕЙСКОЙ СИСТЕМОЙ ПЕРЕВОДА И НАКОПЛЕНИЯ
УЧЕБНЫХ КРЕДИТОВ (ECTS) ДЛЯ IT СПЕЦИАЛЬНОСТЕЙ**

Европейская система перевода и накопления кредитов была первоначально основана в 1989 году как пилотная схема в рамках программы Эразмус. В то время ее целью было обеспечить признание периодов обучения студентов, учащихся за границей путем перевода кредитов. В качестве трансферной системы ECTS применяется более, чем в 30 странах и используется более, чем в одной тысячи учреждений высшего образования.

40 государств, участников Болонского процесса определили ECTS как один из краеугольных камней высшего образования в Европе. Большое количество стран официально применяет ECTS как накопительную систему для своих собственных систем высшего образования, а другие страны готовятся к ее применению. В некоторых странах ECTS стала необходимой для получения аккредитации. Цюрихская конференция по переводу и накоплению кредитов, проведенная в октябре 2002 года Европейской ассоциацией вузов, подтвердила центральную роль ECTS в высшем образовании. ECTS будет иметь фундаментальное место в разработке национальной и Европейской квалификационной структуры. Схемы, основанные на ECTS начинают применяться и на других континентах.

Правильное использование ECTS зависит от усилий тысяч высоко мотивированных людей, занятых в высшем образовании в разных условиях и культурном окружении.

Модель для определения нагрузки студента по программам высшего образования была разработана в рамках университетского проекта «Настройка образовательных структур в Европе», поддерживаемой Европейской комиссией в рамках программы Сократ. Этот проект уделяет внимание полученным знаниям и общим (характерным) компетенциям, а также квалификации по конкретному предмету. Он также показывает, что на учебную нагрузку оказывают влияния подходы и оценка результатов и описывает четырехступенчатый подход в определении нагрузки студента.

Кредиты могут быть распределены по всем видам учебных программ, независимо от их продолжительности или содержания. Программы могут состоять из годовых курсов или более коротких модулей. Они могут охватывать всю практическую работу и исследования и быть первого, второго или третьего уровня. Кредиты могут быть использованы и для отдельных курсов, таких, как модули, предлагаемые для учащихся, занятых не по полной программе обучения.

Перевод кредитов, заработанных студентом между университетами достаточно сложная процедура. Для упрощения процесса накопления и перевода кредитов как для студентов, так и для университетов, а также для помощи студентам в выборе среди многообразия дисциплин (на основе рейтингов дисциплин, выставляется студентом, и рейтинга актуальности дисциплины, выставляет работодатель) при условии мобильности образования (разные вузы и страны), для фиксации выданных студентам кредитов, а также ряда других функций требуется средство автоматизации.

В соответствии с техническим решением функционал разрабатываемого приложения содержит:

- регистрация и авторизация пользователей;
- детализация информации о студенте;
- создание научных соглашений, дисциплин, групп;
- выбор дисциплин для изучения из списка возможных;
- оценивание результатов изучения дисциплины студентом;
- оценивание дисциплины студентами, прошедшими курс;
- поиск студента по необходимым критериям.

В соответствии с описанными функциями в системе существует три вида ролей:

- администратор;
- преподаватель (координатор);
- студент;

- работодатель.

Преимущества разрабатываемого приложения:

- одно приложение для всех типов пользователей (координатор, преподаватель, студент и работодатель);

- удобство в выборе университета и дисциплин для изучения для студентов;

- оперативность получения результатов изучения дисциплины группой студентов для преподавателей;

- гибкость в переводе кредитов между университетами для координаторов;

- возможность выбора специалистов по необходимым критериям для работодателей.

Принимая во внимание большую популярность операционной системы Android было принято решение сделать приложение именно под эту систему. В качестве базы данных используется набирающая популярность облачная NoSQL база данных для real-time приложений Firebase.

ЛИТЕРАТУРА

1. Кредитно-рейтинговая система и система оценок ECTS. Константин Венцлавович (РУДН). [Электронный ресурс]. - URL: <http://uu.kventz.ru/ects/>– Дата доступа: 4.05.2017

2. European Credit Transfer and Accumulation System (ECTS) (англ.). [Электронный ресурс]. - URL: European Commission. https://ec.europa.eu/commission/index_en/– Дата доступа: 4.05.2017

3. The European Credit system for Vocational Education and Training (ECVET) (англ.). European Commission. URL: European Commission. <http://www.russie.campusfrance.org/node/5823/>– Дата доступа: 4.05.2017

УДК 681.3.053

студ. А.А. Мартынюк

Науч. рук. доц. Н.В. Пацей

(кафедра информационных систем и технологий, БГТУ)

СВЕРТОЧНЫЕ КОДЫ ДЛЯ ИСПРАВЛЕНИЯ ПАКЕТОВ ОШИБОК ПРИ ПОМЕХОУСТОЙЧИВОЙ ПЕРЕДАЧЕ ИНФОРМАЦИИ

В реальных каналах часто вместо одиночных независимых ошибок, в результате коммутационных помех, фиксируют пакеты ошибок.

Пакетом ошибок длины b называется вектор ошибок E , все ненулевые компоненты которого расположены на отрезке из b подряд следующих позиций, причем в начале и конце отрезка расположены ненулевые компоненты. При $b = 1$ получим пакет длиной в один символ, т.е. одиночную ошибку. Это модель пакета в упрощенном виде. Различают однократные и многократные пакеты в зависимости от того, сколько пакетов искажают кодовое слово.

Для обнаружения и исправления пакетов ошибок существуют специализированные коды. Однако, для исправления пакетов ошибок могут использоваться и перемежители. Так, все символы одного кодового слова разносятся друг от друга на расстояние, превышающее длину пакета. В образовавшемся промежутке передаются символы других кодовых слов, так что друг за другом следуют символы не одного, а разных кодовых слов. В этом случае происходит переход от пакета к одиночным ошибкам и могут быть использованы коды для исправления одиночных независимых ошибок.

Строго говоря в классе сверточных кодов существуют отдельные частные решения для исправления пакетов ошибок. Однако не существует строгой обобщенной методики исправления пакетов.

Наиболее известен метод Ивадари-Месси, применимый для кодов со скоростями $R = (v - 1) / v$, где v - произвольное положительное число большее двух [1-2].

Данный метод позволяет для любого положительного целого числа β достичь исправляющей пакеты способности βv при, так называемом, защитном интервале $\beta v (2v - 1) + \frac{1}{2} v^2(v - 1) - 1$. Защитным интервалом будем считать последовательность свободных от ошибок символов, следующих за пачкой ошибок. Он необходим, чтобы пачки ошибок можно было исправлять. При больших β отношение длины защитного интервала к корректирующей пакеты способности стремится к $2v - 1$. Структурная схема кодера для метода декодирования Ивадари - Месси изображена на рисунке 1.

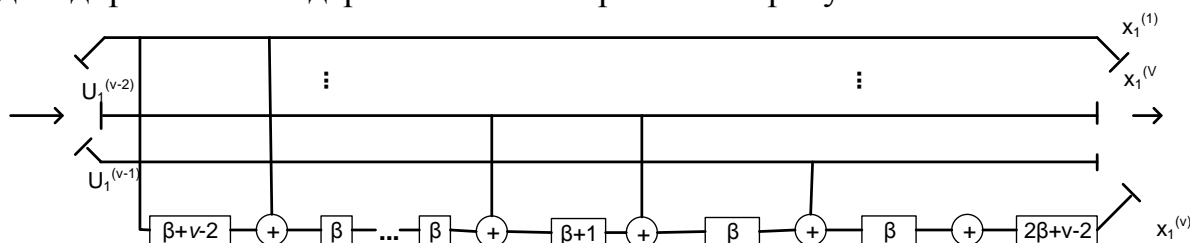


Рисунок 1 - Сверточный кодер для произвольного R

При декодировании используется синдром. В случае отсутствия ошибок синдром должен быть нулевым. Основная идея заключается в том, что любая ошибка должна иметь в синдроме так называемую собственную часть, не пересекающуюся с собственными частями других ошибок. Символы синдрома S_n определяются через принятые по каналу символы с помощью соотношения:

$$S_n = y_n^{(v)} \oplus y_{n-(v-2)\beta-(v-2)}^{(v-1)} \oplus y_{n-(v-1)\beta-(v-2)}^{(v-1)} \oplus \dots \\ \dots \oplus y_{n-(2v-2)\beta-(v-1)}^{(1)} \oplus y_{n-(2v-1)\beta-(v-2)}^{(1)}$$

Если шумовая последовательность содержит лишь пакеты не больше чем из символов βv при защитном интервале $\beta v (2v - 1) + \frac{1}{2} v 2(v - 1) - 1$, то каждый синдром S_n может содержать не более одного неисправленного символа.

Близкий класс кодов был исследован Вайнером и Эшем, Берлекэпом и Месси. Рассмотренный в этих работах метод приводит к несколько меньшему защитному интервалу при заданной корректирующей пакеты способности, чем метод Ивadari — Месси, однако увеличение сложности оборудования делает его менее полезным в практическом применении.

Таким образом, использование сверточных кодов и порогового декодирования позволяют в каскаде эффективно исправлять длинные пакеты при наличии промежуточных зон, содержащих мало ошибок. К практически эффективным кодам относятся коды: Вайнера и Эша, Препарта, Берлекэпа и Месси [1]. Кодирование подобных кодов основано на теории матриц, а декодирование на структуре систематических кодов.

ЛИТЕРАТУРА

1. Галпаген Р. Теория информации и надежная связь: Моветское радио - 1974.
2. Блейхут Р. Теория и практика кодов контролирующей ошибки. Перевод с англ. ИИ Группко ВМ Блиновский. Под редакцией: К.Ш. Зигангирова — М.: Мир, 1986. — 576 с

УДК 004.588

Студ. А.А. Боровик

Науч. рук. доц. Н.В. Пацей

(кафедра информационных систем и технологий, БГТУ)

ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В ОБРАЗОВАНИИ

В настоящее время наблюдается активное внедрение информационных технологий практически во все сферы деятельности

человека. Не исключением является и образовательный процесс. Появляется огромное количество различных технологических разработок и программных решений, предназначенных для использования в образовательных целях, и актуальной задачей, которая встаёт перед преподавателями, является внедрение данных технологий в учебный процесс для ежедневного использования как самими преподавателями, так и обучающимися.

Одной из передовых технологий является дополненная реальность, популярность которой в значительной степени возросла в последнее время, что стимулировало развитие различных разработок в данной сфере. Каждая технология имеет свои преимущества и недостатки. Если говорить о дополненной реальности, то она позволяет значительно расширить возможности образовательного процесса.

Дополненной реальностью (англ. Augmented Reality) называют тесное взаимодействие окружающего мира и виртуально созданной компьютерной реальности. Целью использования приложений данной направленности является расширение и пояснение того или иного реально существующего объекта при помощи различного рода устройств.

Дополненная реальность без сомнений является огромным прорывом и в способе подачи образовательного материала, и в усвоении информации школьниками и студентами. Эффективность ее применения доказывается различного рода тестами и экспериментами, которые демонстрируют достаточно яркие и показательные результаты. Например, была проведена серия экспериментов, при которых одной группе обучающихся во время уроков демонстрировали наглядный материал с дополненной реальностью, а второй группе — традиционные плакаты и схемы. В результате было выявлено, что для группы, где для подачи материала использовались технологии дополненной реальности, процент усвоения информации приблизился к 90%, уровень восприятия дисциплины возрос, и удалось удержать внимание порядка 95 % аудитории, в то время как во второй группе со стандартными пособиями все показатели были приблизительно на 50% меньше. Также было выявлено, что трехмерное изображение стимулирует мышление, развивает моторику, мимику, внимание и повышает степень усвоения, запоминания и, что самое главное, понимания подаваемой информации.

Вне зависимости от изучаемого материала дополненная реальность помогает повысить его привлекательность для учеников любого возраста и увеличивает мотивацию к получению знаний. Обучение с использованием дополненной реальности имеет также и материальные плюсы: отпадет необходимость в производстве и использовании громоздких плакатов, стендов, досок и прочих наглядных пособий, сократятся расходы на печать некоторых учебников. Размещенный перед камерой двумерный маркер, с которого считывается и анализируется вся информация, — вот все, что необходимо для получения эффекта дополненной реальности.

В начале проектирования любой системы необходимо иметь четкое представление о задачах, для решения которых она предназначена. Для определения и представления функций, которыми должна обладать система, необходимо разработать диаграмму вариантов использования. В соответствии с поставленными задачами, была разработана диаграмма вариантов использования системы, которая позволит внедрить технологии дополненной реальности в учебный процесс, представленная на рисунке 1.

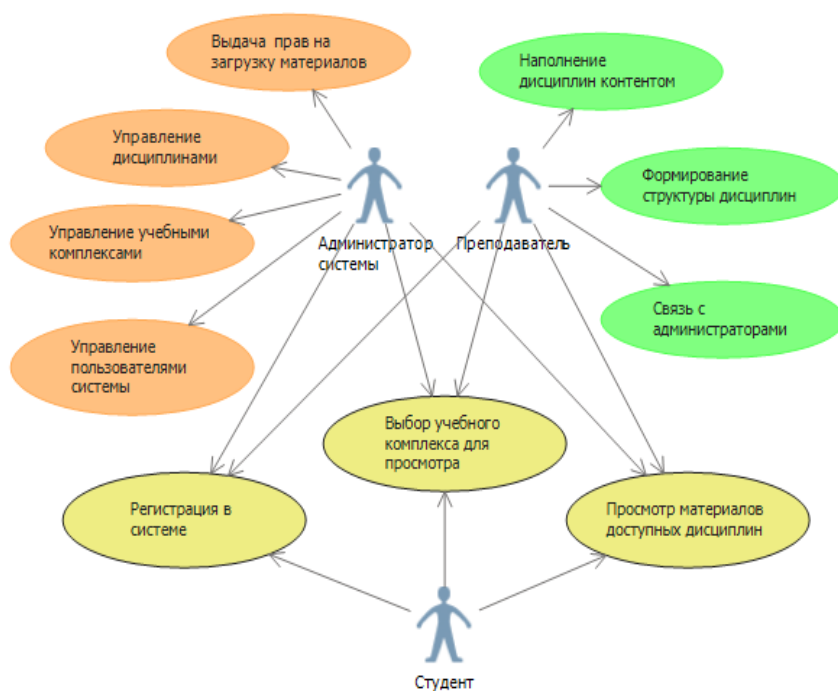


Рисунок 3 - Диаграмма вариантов использования

Предполагаемая структура системы состоит из модуля управления контентом, модуля хранения и обработки данных и модуля отображения.

Модуль управления контентом представляет собой веб-приложение, при помощи которого выполняется администрирование системы, а также производится наполнение и управление контентом.

Модуль отображения – приложение, предназначенное для работы на мобильном устройстве. При помощи данного приложения пользователи могут авторизоваться, выбрать интересующий материал и изучить его.

Модуль хранения и обработки данных является связующим звеном данной системы. Он отвечает за хранение информационного наполнения системы, а также за формирование информационной модели в соответствии с запросами пользователей.

Сейчас дополненная реальность присутствует практически на всех устройствах, от смартфонов до компьютеров со встроенными камерами. Поэтому с учетом доступности подобного рода гаджетов практически для всех слоев населения технический вопрос использования AR в образовании упирается только в выбор и внедрение конкретной, унифицированной платформы, на которой будет осуществляться весь процесс образования.

В настоящее время доступно множество библиотек, предназначенных для создания приложений дополненной реальности:

Таблица 2-- Библиотеки дополненной реальности

AR-фреймворк	Компания	Лицензия	Поддерживаемые платформы
Vuforia	Qualcomm	Платная и бесплатная	Android, iOS, Unity
ARToolkit	DAQRI	Бесплатная	Android, iOS, Windows, Linux, MacOS X, SGI
Kudan	Kudan Limited	Платная	Android, iOS, Unity

В целом, перечисленные библиотеки дополненной реальности предоставляют широкий спектр возможностей разработчику – от поддержки различных операционных систем до развернутого набора инструментов для распознавания и отслеживания объектов. Исходя из поставленных целей, становится очевидным, что наиболее предпочтительным становится использование библиотеки, которая будет требовать минимального вложения ресурсов, как с финансовой точки зрения, так и с точки зрения сложности разработки. Из вышеперечисленных библиотек наиболее соответствует поставленным требованиям Vuforia. Наличие бесплатной версии позволит сократить финансовые издержки, а поддержка Unity

позволит разрабатывать приложение, которое будет поддерживать всё множество платформ, поддерживаемых Unity, что упрощает процесс разработки и поддержки данной системы.

В заключение можно отметить, что сейчас, к сожалению, нет четкого движения в этом направлении и конкретных программ, позволяющих внедрять технологии дополненной реальности. Тем не менее, многие специалисты в области информационных технологий сходятся во мнении, что будущее дополненной реальности в различных областях нашей жизни имеет довольно радужные перспективы, а AR-технологии в образовании рано или поздно выведут систему образования на качественно новый уровень.

ЛИТЕРАТУРА

1. Зильберман М. А. Использование дополненной реальности в образовании: из опыта работы. [Электронный ресурс] - Режим доступа: <http://www.slideshare.net/School91perm/zilberman-42905347> (дата обращения 15.12.2016).
2. Петрова О. Г. Дополненная реальность в образовании. [Электронный ресурс] - Режим доступа: <https://sites.google.com/site/relarn2010/glavnaa-stranica/tezisy-relarn-2013/petrova-oksana-dopolennaarealnost-v-obrazovanii> (дата обращения 20.12.2016).
3. Обзор AR-библиотек для создания приложений с дополненной реальностью. [Электронный ресурс] - Режим доступа: <https://azoft.ru/blog/biblioteki-dopolnennoj-realnosti-dlya-ar-prilozhenij/> (дата обращения 20.03.2017).

УДК 004.42

Студ. А.В. Скородумова,
Науч. рук. ассист. Новицкая А.Д.
(кафедра информационных систем и технологий, БГТУ)

ВОЗМОЖНОСТИ ГЕОЛОКАЦИИ НА ПРИМЕРЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

В последние годы в нашу жизнь прочно вошли слова и понятия, о которых многие из нас раньше даже не подозревали. Одно из них – геолокация, которой раньше пользовались разве что представители некоторых профессий – военные, моряки, летчики и некоторые другие. Сегодня геолокация нашла широкое применение в повседневной жизни, развлечениях, интернет-общении миллионов людей, которые охотно пользуются новыми возможностями

электронных устройств.

Слово «геолокация» образовано из двух слов – греческого «гео», означающего поверхность земли, и английского «location» – расположение, позиция. Геолокация – это определение местоположения (геопозиции) устройства пользователя по исходящему от него радиосигналу.

Идея создания спутниковой навигации родилась ещё в 50-е годы. В тот момент, когда СССР был запущен первый искусственный спутник Земли, американские учёные во главе с Ричардом Кершнером наблюдали сигнал, исходящий от советского спутника и обнаружили, что благодаря эффекту Доплера частота принимаемого сигнала увеличивается при приближении спутника и уменьшается при его отдалении. Суть открытия заключалась в том, что если точно знать свои координаты на Земле, то становится возможным измерить положение и скорость спутника, и наоборот, точно зная положение спутника, можно определить собственную скорость и координаты [1].

Реализована эта идея была через 20 лет. В декабре 1973 году была инициирована программа DNSS, позже переименованная в Navstar-GPS, а затем в GPS.

GPS – это англ. GlobalPositioningSystem — система глобального позиционирования, читается Джи Пи Эс) — спутниковая система навигации, обеспечивающая измерение расстояния, времени и определяющая местоположение во всемирной системе координат WGS 84 [1].

В 1982 году появилась глобальная навигационная спутниковая система (ГЛОНАСС) – российская спутниковая система навигации, разработка которой началась в СССР по заказу Министерства обороны[2].

Кроме того, существуют еще европейская система Galileo и китайская система BeiDou/Compass.

На сегодняшний день функционируют 2 системы навигации: американская навигационная система GPS и российская ГЛОНАСС. В настоящее время точность GPS составляет 2-4 метра, а ГЛОНАСС – 3-6.

Первоначально GPS – глобальная система позиционирования, разрабатывалась как чисто военный проект. Но после того, как в 1983 году вторгшийся в воздушное пространство Советского Союза самолёт Корейских Авиалиний с 269 пассажирами на борту был сбит из-за дезориентации экипажа в пространстве, президент США Рональд Рейган с целью не допустить в будущем подобные трагедии

разрешил частичное использование системы навигации для гражданских целей. Во избежание применения системы для военных нужд точность была уменьшена специальным алгоритмом. Затем появилась информация о том, что некоторые компании расшифровали алгоритм уменьшения точности на частоте L1 и с успехом компенсируют эту составляющую ошибки. В 2000 г. это загроуление точности отменил своим указом президент США Билл Клинтон [1].

Хотя ранее GPS задумывалась как исключительно военная функция, то начиная с 2000-х годов навигация вошла в жизнь обычных людей.

Геолокация приносит огромную пользу обычным людям, а также тем, кому в силу профессии необходимо ориентироваться на местности: водителям, перевозящим пассажиров и грузы, рыбакам, землеустроителям и многим другим.

К наиболее популярным возможностям геолокации можно отнести:

1. Определение и мониторинг своего местонахождения.
2. Прокладывание маршрутов.
3. Ориентация на местности(поиск на местности объектов инфраструктуры).
4. Осуществление поиска электронных гаджетов(утраченных и украденных).
5. Слежение за перемещением своих посылок, перевозимых грузов.

Тема моего диплома «Геолокационное мобильное приложение для построения и мониторинга маршрута объекта». Для его реализации выбрана операционная система Android по ряду причин. Во-первых, Android поддерживает большое количество устройств разных производителей. Во-вторых, Android характеризуется высокой доступностью средств разработки. Средства разработки для платформы Android бесплатны, в то время как разработка, к примеру, под iOS (от компании Apple) требует немалых начальных финансовых вложений.

С экономической точки зрения геолокационные приложения – это серьезный фактор в мобильной телефонии. Значительная часть прибыли от продажи мобильных приложений относится к программам именно этого класса, кроме того, их доля на рынке продолжает быстро расти [3].

Когда-то люди ориентировались по звездам. Сейчас все гораздо проще, так как существует достаточное количество приложений,

которые сделают это за Вас.

Говоря именно о всем известных, популярных и используемых геолокационных приложениях, можно выделить такие как Google Maps, Яндекс Карты или OSMand. Для большей наглядности они представлены на рисунке 1.



Рисунок 1 – Наиболее популярные мобильные геолокационные приложения

Каждое из них работает со своими картами: google, yandex, openstreetmap. Последняя чаще всего используется в offline-приложениях геолокации.

Перечисленные выше приложения позволяют проложить маршрут, однако явный мониторинг этого маршрута в них не доступен. То есть он как бы есть, точка на карте движется вместе с мобильным устройством, на котором установлено приложение, но если появится желание «проверить себя» в тот или иной момент времени эти приложения не помогут. Таким образом, в полной мере оценить время потраченное на дорогу и выявить отклонения от маршрута невозможно. Данный проект пригодится тем, кто плохо ориентируется на местности и часто прибегает к применению геолокационных мобильных приложений, предпочитая осуществлять мониторинг своего пути следования.

Из ранее перечисленных возможностей геолокации, разрабатываемое, главным образом, позволяет строить маршрут и осуществлять его мониторинг.

В своем дипломном приложении я использовала карты google. Чтобы использовать Google Maps Android API, необходимо зарегистрировать свой проект приложения в Google API Console и получить ключ Google API, который далее нужно добавить в свое приложение.

Текущий интерфейс работы приложения представлен на рисунке 2.

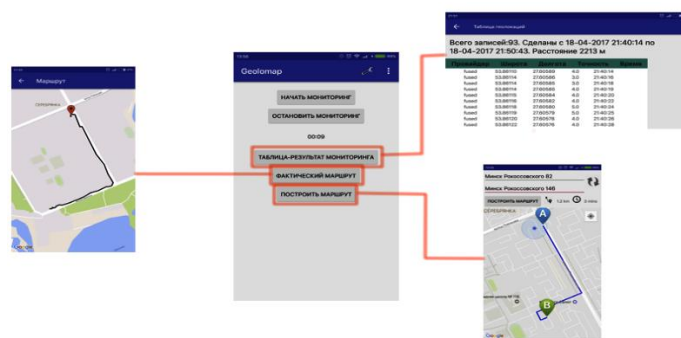


Рисунок 2 – Интерфейс работы приложения

Верхние 2 кнопки осуществляют контроль за мониторингом. Компонент Chronometer осуществляет счёт времени. В таблице указываются координаты, полученные посредством мониторинга, а фактически маршрут рисуется на основании заполненной таблицы.

Кнопка «Построить маршрут» позволяет построить маршрут.

ЛИТЕРАТУРА

1. GPS [Электронный ресурс]. – 2017. – Режим доступа: <https://ru.wikipedia.org/wiki/GPS>. Дата доступа: 18.04.2017.

2. ГЛОНАСС [Электронный ресурс]. – 2017. – Режим доступа: <https://ru.wikipedia.org/wiki/ГЛОНАСС>. Дата доступа: 18.04.2017.

3. Геолокация и картография в Android приложении [Электронный ресурс]. – 2017. – Режим доступа: <http://src-code.net/geolokaciya-i-kartografiya-v-android-prilozhenii/>. Дата доступа: 19.04.2017.

УДК 004.421.6

Студ. В.Е. Журавлёв

Науч. рук. ст.преп. Ю.О. Герман

(кафедра информационных систем и технологий, БГТУ)

ANDROID-ПРИЛОЖЕНИЕ ДЛЯ ПЕРЕВОДА ТЕКСТА НА ИНОСТРАННЫЙ ЯЗЫК

В настоящее время мобильные устройства находят свое применение практически во всех областях жизнедеятельности человека. Всё больше и больше сервисов и услуг переходят с настольных версий на веб или мобильные платформы.

На сегодняшний день людям приходится читать много литературы на иностранном языке. Чтобы скоротать время, например, в транспорте, большинство читает материал на своих мобильных телефонах. Однако незнание иностранного языка приводит к тому, что

необходимо постоянно прерываться, заходить в словарь, вводить нужное слово, что неудобно для пользователей. С целью решения подобных проблем разрабатывается мобильное приложение Copy Translate, показывающее перевод слова сразу, после того как оно скопировано в буфер. Суть использования Copy Translate заключается в следующем: Android-приложение работает постоянно в фоне как служба, следит за буфером обмена и, как только в буфер поступает новый текст, переводит его с выдачей результата.

Прототип приложения был создан в программе Adobe Experience Design [1]. Дизайн приложения сделан в стиле Google Material Design. Material Design — дизайн программного обеспечения и приложений операционной системы Android от компании Google. Идея дизайна заключается в приложениях, которые открываются и сворачиваются как карточки, используя эффекты теней. У приложений не должно быть острых углов, карточки должны переключаться между собой плавно и практически незаметно [2]. Основываясь на этом, были созданы основные блоки приложения.

Согласно Google Material, основные логические блоки следует вынести в «плитки», плитка – это прямоугольник с закругленными углами, который, используя тень, находится как бы над другими элементами управления.

Поиск, доступ в меню и включение-выключение службы вынесен в toolbar, который, опять же собран по советам Material Design Guidelines.

Программная реализация написана на языке Java на основе построенной UML-модели. Java-код создаёт сам Enterprise Architect.

В приложении имеется возможность просматривать историю переводов. Здесь же реализован поиск нужного перевода. Было решено сделать поиск нечётким. Нечеткий поиск является крайне полезной функцией любой поисковой системы. Вместе с тем, его эффективная реализация намного сложнее, чем реализация простого поиска по точному совпадению.

Нечеткий поиск - это поиск информации, при котором выполняется сопоставление искомого слова заданному образцу или близкому к этому образцу значению. Алгоритмы нечеткого поиска используются в большинстве современных поисковых систем, например, для проверки орфографии. Классическая задача нечеткого поиска может принимать следующий вид: имеется текстовая информация определенного размера. Пусть пользователь вводит слово или фразу для поиска, необходимо найти в тексте все совпадения с

заданным словом с учетом возможных допустимых различий. Например, при запросе пользователя «интернет» нужно найти «интерн», «интернат» и другие слова.

Для того, чтобы оценить сходство двух слов, в тексте используются специальные метрики нечеткого поиска. Метрикой нечеткого поиска называют функцию расстояния между двумя словами, позволяющую оценить степень их сходства в данном контексте [3].

В качестве метрик используют расстояния Хемминга, Левенштейна, Дамерау-Левенштейна. Расстояние Хемминга - это число позиций, в которых соответствующие символы двух слов одинаковой длины различны. У расстояния Хемминга есть один существенный недостаток - сравнивать можно только слова одинаковой длины. Из-за этого данную метрику практически не применяют на практике.

Расстояние Левенштейна или расстояние редактирования - это минимальное количество операций (вставки, удаления одного символа и замены одного символа на другой), необходимых для преобразования одной строки в другую. Данная метрика имеет следующие недостатки:

– при перестановке слов в предложении расстояние принимает большое значение;

– расстояние Левенштейна значительно зависит от длины слова.

Расстояние Дамерау-Левенштейна - это мера разницы двух строк символов, определяемая как минимальное количество операций вставки, удаления, замены и перестановки соседних символов, необходимых для перевода одной строки в другую. Данная метрика отличается от расстояния Левенштейна только добавлением новой операции (перестановки). Расстояние Дамерау — Левенштейна $d_{a,b}(|a|, |b|)$ между двумя строками a и b определяется функцией вида [3]:

$$d_{a,b} = \min(X_x + 1, X_y + 1, X_z + C_{\text{замены}}, X_m + C_{\text{транспозиции}})$$

$$C_{\text{замены}} = \begin{cases} 1, \text{если } S_1[i] \neq S_2[j] \\ 0, \text{иначе} \end{cases}, \quad (1)$$

$$C_{\text{транспозиции}} = \begin{cases} 1, \text{если } S_1[i] = S_2[j-1] \text{ и } S_1[i-1] = S_2[j] \\ \infty, \text{иначе} \end{cases}$$

где X_x , X_y , X_z , X_m – операции вставки, удаления, замены и перестановки символа соответственно.

В результате работы алгоритма Дамерау-Левенштейна составляется матрица преобразования символов, где ее значения - это расстояния для преобразования символов. Итоговым значением преобразования одной строки в другую будет элемент с максимальными индексами. Например, расстояние между словами «дагестан» и «арестант» будет равно 3. Данный алгоритм требует довольно значительных затрат памяти и времени выполнения. Однако он может быть упрощен путем сокращения количества вычислений. Сравнение строк может происходить с учетом морфем, так могут подвергаться трансформации только суффиксы слов или окончания. Другим вариантом снижения затрат может быть использование ограничений по количеству сравнений.

Ключевым недостатком данных методов является время выполнения поиска. При работе с мобильными устройствами время является важным критерием работы приложения. Однако данные методы можно оптимизировать за счет их комбинаций или внесения дополнительных условий, задаваемых пользователем или разработчиком.

ЛИТЕРАТУРА

1. Adobe Experience Design [Электронный ресурс]. – Режим доступа: <http://www.adobe.com/ru/products/experience-design.html>.
2. Google Material Design [Электронный ресурс]. – Режим доступа: <https://material.io/guidelines/#introduction-principles>.
3. Нечеткий поиск в тексте и словаре [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/114997>.

УДК 519.256

Студ. Д.А. Радиванович

Науч. рук. ст.преп. Ю.О. Герман

(кафедра информационных систем и технологий, БГТУ)

ПРИЛОЖЕНИЕ ДЛЯ ФИЛЬТРАЦИИ ЭЛЕКТРОННОЙ КОРРЕСПОНДЕНЦИИ

Спам — массовая рассылка сообщений различного содержания, пользователям, не выражающим желания их получать. Согласно определению «Лаборатории Касперского» спам — это анонимная массовая непрошенная рассылка. В этом определении важно каждое включенное в него слово. Анонимная: все страдают именно от автоматических рассылок со скрытым или фальсифицированным

обратным адресом. Массовая: эти рассылки именно массовые, и только они являются настоящим бизнесом для спамеров и настоящей проблемой для пользователей. Непрошенная: очевидно, подписные рассылки и конференции не должны подпадать под наше определение (хотя условие анонимности и так в значительной мере это гарантирует) Спам является довольно серьёзной проблемой нашего времени, создающей множество неудобств большинству пользователей. Мошенничество, вымогательство, захламление почтовых ящиков, распространение вирусов и даже доведение человека до нервного срыва – всё это и многое другое реализуется при помощи спама. На самом деле понятие спам, в котором понимает его большинство, не очень корректно. В зависимости от содержания таких сообщений, принято понимать два обобщённых типа: UCE - «коммерческий» спам — «unsolicited commercial e-mail» и (UBE) «некоммерческий» — «unsolicited bulk e-mail». По статистике на апрель 2015 года, собранной Лабораторией Касперского, спам составил 71,1% глобального почтового траффика. К тому же это на 7,6% больше чем в марте месяце. Причиной возросшего количества сообщений стали пасхальные праздники. Собственно, практически каждый значимый праздник, становится «праздником» для спамеров. Такое количество бесполезных сообщений создаёт проблему для компаний, предоставляющих услуги электронной почты, ведь это в разы увеличивает нагрузку на серверы и заставляет тратить большее количество денег на оборудование. В роли источника такой рассылки могут выступать как сами спамеры, так и ни о чём не подозревающий пользователь, чей IP адрес или адрес электронной почты, был захвачен злоумышленниками и используется для рассылки [1].

Методов борьбы со спамом существует огромное количество, что, во-первых, показывает, что это именно проблема для Интернета, а, во-вторых, – что ею активно занимаются. Но никто не стал бы заниматься этим просто так, что наталкивает на мысль – спам приносит хорошие доходы своим организаторам, что говорит о том, что несмотря ни на что, пользователь всё-таки пользуется тем, что предлагают спам-сообщения, и до тех пор, пока доходы превышают расходы, спам будет существовать. Тогда можно с уверенностью сказать, что главным методом борьбы со спамом является полный отказ от услуг, предлагаемых таким образом, и максимально возможное игнорирование подобных сообщений. В этом случае лица, организующие и использующие такие методы воздействия на обывателей, потеряют всякий доход с этого предприятия. Но,

несмотря на идейные, есть также и технические методы борьбы со спамом [2].

Превентивные методы – методы предварительной защиты, направленные на предотвращение попыток использования, к примеру, электронной почты. Здесь многое зависит от человеческого фактора. Если вы и в правду хотите сохранить свою электронную почту в безопасности, то можно предпринять целый ряд мер. Не публиковать свой адрес на общедоступных сайтах. Если вдруг это действительно необходимо, то постараться сделать так и написать его таким образом, чтобы он был не читаем для программ-харвестров, например, поставить нижние подчёркивания и пробелы в словах, заменить буквы на цифры, где возможно. Для того, кто должен прочитать это сообщение, оставить комментарии, хотя в некоторых случаях программы могут распознать закодированный текст.

Публичные сайты, должны быть организованы так, чтобы электронный адрес не был виден, а сообщения было бы возможно отправлять, но никнэйму. Адрес можно предоставить в виде картинки, или QR-кода, при этом не стоит доверять онлайн-службам, делающим это, так как известны случаи продажи таких баз данных с адресами спамерам. Лучше разобраться самому в графическом редакторе, или в крайнем случае написать на бумаге и сфотографировать. На web-страницах адреса можно кодировать с помощи JavaScript. Можно разделить всю свою почту на два ящика, один, используя для обмена личными и корпоративными сообщениями, а второй для регистрации и авторизации на различных сайтах и в разных службах, не внушающих доверия. Естественно, никогда не открывать и не читать сообщения подозрительного содержания, и тем более не отвечать, если вы на все сто не уверены в том, от кого это сообщение, не загружать никаких изображений, приложенных в письме.

При создании почтового ящика, постараться придумать сложный, трудноподбираемый и не имеющий никакого конкретного смысла адрес. Не использовать имена или простые распространённые слова, ведь спамеры люди подкованные и с лёгкостью, пользуясь «особенными словарями» и методом перебора, подберут ваш адрес. Также постараться сделать имя длинным, лучше всего, если это будет простой набор букв, не имеющий смысла. Использование, например, букв русского и латинского алфавита, а также цифр, изрядно усложнит задачу спамерам, и практически исключит случайную атаку. Однако все эти попытки избежать угадывания, несут один значительный минус. Делая email сложным для спамеров, мы также

усложняем жизнь себе. Особенно неудобно, если это адрес какой-нибудь фирмы, и он должен быть понятен и прост для чтения.

На сегодняшний день наиболее действенным методом является фильтрация. Автоматическая фильтрация – принцип, при котором специализированное программное обеспечение (ПО) сканирует сообщения и принимает решение спам/не спам. Предназначено, как для простых пользователей, так и для использования на серверах. Использует два основных подхода.

Первый – это анализ текста сообщения, которое опираясь на известные приёмы спамеров, методы байесовской фильтрации, а также поиск по ключевым словам принимает решение, является ли сообщение спамом или нет и помещает в соответствующую папку. Естественно, прежде чем запускать подобную программу, нужно провести её предварительное обучение. Однако при работе такого ПО не у клиента, а на сервере, возникает риск, что фильтр ошибочно пометит сообщение как спам, и оно не дойдет до пользователя. Здесь требуются большое количество статистических данных для увеличения вероятности правильного распознавания.

Второй подход заключается в том, чтобы, не изучая текст письма, опознать самого отправителя как спамера и заблокировать. Такое программное обеспечение может работать только на сервере. Но опять же, не исключены вероятности ошибки, и что простая массовая рассылка людям, которые друг с другом знакомы, будет предпринята как попытка распространения спама. Существуют также специализированные online-сервисы, например, «Лаборатория Касперского», Outcom «СПАМОРЕЗ», ИНКАП «Антиспам-Пост», ContrSpam, Антиспамус, предоставляющие платную защиту от спама. Изменение MX-записи в доменном имени предприятия особым образом позволяет перенаправить почту для защищаемого домена на специализированный почтовый сервер, где она очищается от спама и вирусов, а затем направляется на корпоративный почтовый сервер. Неавтоматическая фильтрация – метод позволяющий пользователю самостоятельно создавать фильтры, состоящие из слов или выражений. Однако это слишком трудоёмко и практически не используется. Сформировав свой собственный фильтр, пользователь может быть уверен, что нужные ему сообщения не будут отсеяны [3].

Данное программное приложение позволяет фильтровать письма и делать вывод о спаме, опираясь на известную теорему Байеса [4]:

$$P(c/d) = \frac{P(d/c)}{P(d)}, \quad (1)$$

где $P(c/d)$ — вероятность, что сообщение d принадлежит классу c , именно её нам надо рассчитать;

$P(d/c)$ — вероятность встретить сообщение d среди всех сообщений класса c ;

$P(c)$ — безусловная вероятность встретить сообщение класса c в базе сообщений;

$P(d)$ — безусловная вероятность сообщения d в базе сообщений.

Приложение реализовано на платформе .Net с помощью языка C#.

Для верной классификации писем необходимо собрать статистическую информацию по предыдущим данным, которые заносятся в подключаемую к модулю базу данных. Любое новое письмо будет проверено на наличие спама, используя указанную теорему.

ЛИТЕРАТУРА

1. Спам [Электронный ресурс] / Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B0%D0%>. – Дата доступа: 01.04.2017.

2. Борьба со спамом: история и методы [Электронный ресурс] – Режим доступа: https://mipt.ru/dmcp/student/diff_articles/no_spam.php. – Дата доступа: 03.04.2017.

3. Методы борьбы со спамом [Электронный ресурс] – Режим доступа: <http://www.securelist.com/ru/threats/spam?chapter=157> – Дата доступа: 03.04.2017.

4. Герман, О. В. Экспертные системы: лабораторный практикум / О. В. Герман, Н. В. Батин. – Минск : БГУИР, 2003. – 75 с.

УДК 004.4

Студ. А.С. Бубель

Науч. рук. доц. Н.Н. Буснюк

(кафедра информационных систем и технологий, БГТУ)

АЛГОРИТМ ДЛЯ НАХОЖДЕНИЯ ВСЕХ ВОЗМОЖНЫХ РЕШЕНИЙ В СЕТЕВОМ ГРАФИКЕ, ЗАДАННОМ В ДЕКАРТОВОЙ СИСТЕМЕ КООРДИНАТ

Цель и задачи. Разработать алгоритм, который будет находить максимальное и минимальное расстояния между начальным узлом

сетевого графика и конечным узлом n , если возможно пройти это расстояние различными маршрутами [2]. Количество узлов n вводится вручную. Каждый узел задаётся в декартовой системе координат. Дуги между узлами создаются вручную.

Практическое применение. Алгоритм можно использовать в навигации и прокладке маршрутов, т.к. каждый узел сетевого графика имеет свои координаты на осях x , y . [1]. Маршрут может разветвляться, а сам алгоритм выводит все возможные маршруты для заданного условия, максимальный и минимальный маршрут, а также их длины.

Полученные результаты и их описание. Вводится количество узлов сетевого графика, а также координаты x , y для каждого из них. Инициализируется двумерный динамический массив M , в котором будут храниться решения. Создаются дуги между узлами. Производится запись решения в массив M , преждевременно для каждого маршрута проверяется: есть ли соответствующий переход из заданного узла функцией `isNum`. Если для узла создаётся больше 1 дуги - это ведёт к разветвлению маршрута: функция `new_str` производит копирование уже построенного решения до разветвления в новую строку массива M . Решения разветвляются. После произведённых вычислений в каждой строке двумерного массива M хранится полное и уникальное решение для данного сетевого графика. Каждый элемент строки массива представляет собой дугу, где номер столбца есть узел, от которого строится дуга к узлу, номер которого записан в этой ячейке массива. Производятся вывод маршрутов из массива. Вычисляются наибольший и наименьший маршруты путём складывания длин дуг, хранящихся в строке массива. Вычисление длины дуги производится в функции `length_s`. Производится вывод результатов выполнения операции. Освобождается память, выделенная под массив M .

Пример 1. Имеется сетевой график (рис. 1), состоящий из четырёх узлов и шести дуг в декартовой системе координат. В данном сетевом графике прослеживается всего три возможных маршрута: $1 - 2 - 4$; $1 - 2 - 3 - 4$; $1 - 3 - 4$. Введём известные нам данные в программу и получим результат (рис. 2).

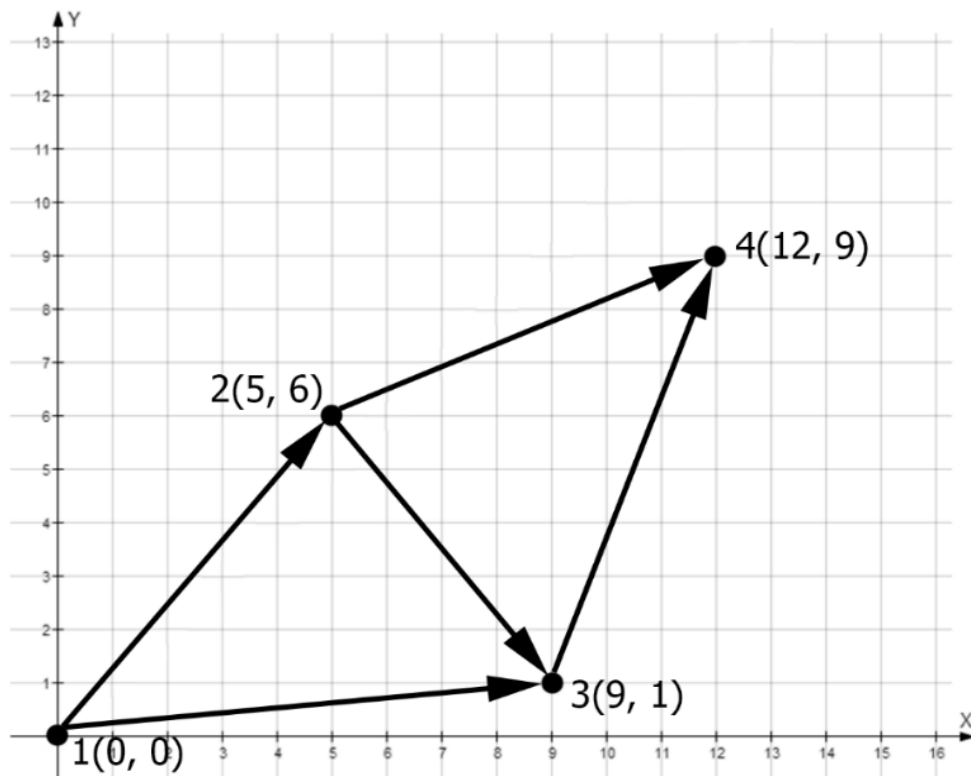


Рисунок 1 - Сетевой график, состоящий из четырех узлов и пяти дуг.

```

Все возможные пути:
1 > 2 > 3 > 4
1 > 3 > 4
1 > 2 > 4

Решение, имеющее максимально возможную длину пути: 1 > 2 > 3 > 4
Длина пути: 22.7574

Решение, имеющее минимально возможную длину пути: 1 > 2 > 4
Длина пути: 15.426
    
```

Рисунок 2 - Вывод результата в консольном приложении Win32.

Пример 2. Имеется сетевой график, состоящий из десяти узлов и шестнадцати дуг в декартовой системе координат (рис. 3). В данной сетевом графике существует 8 возможных маршрутов: 1 – 10; 1 – 2 – 3 – 5 – 8 – 9 – 10; 1 – 2 – 3 – 5 – 7 – 9 – 10; 1 – 2 – 3 – 8 – 9 – 10; 1 – 2 –

10; 1 – 2 – 4 – 7 – 9 – 10; 1 – 2 – 4 – 6 – 9 – 10 ; 1 – 2 – 4 – 6 - 10. Введём известные нам данные в программу и получим результат (рис. 4).

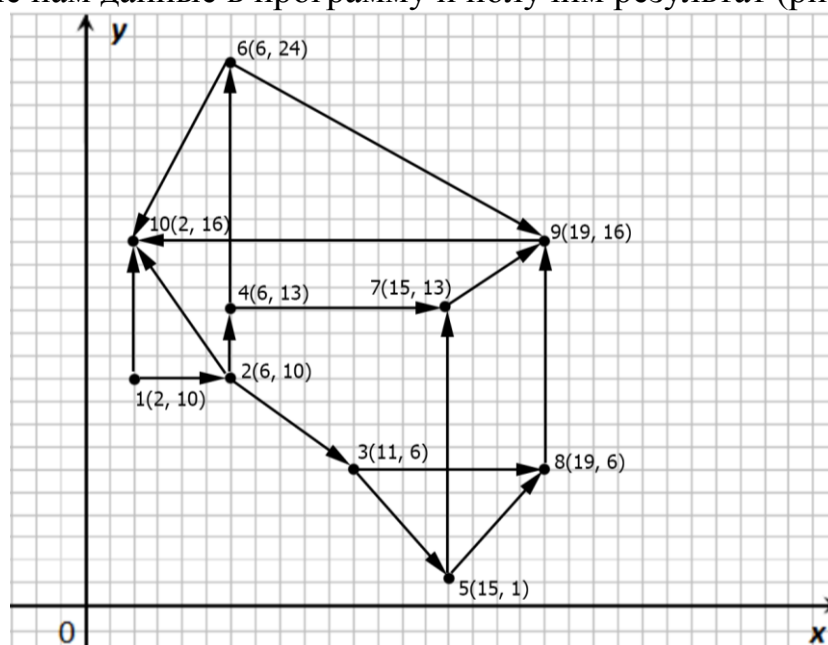


Рисунок 3 - Сетевой график, состоящий из десяти узлов и шестнадцати дуг

```

Все возможные пути:
1 > 2 > 3 > 5 > 7 > 9 > 10
1 > 10
1 > 2 > 4 > 6 > 9 > 10
1 > 2 > 10
1 > 2 > 3 > 8 > 9 > 10
1 > 2 > 4 > 7 > 9 > 10
1 > 2 > 3 > 5 > 8 > 9 > 10
1 > 2 > 4 > 6 > 10

Решение, имеющее максимально возможную длину пути: 1 > 2 > 3 > 5 > 7 > 9 > 10
Длина пути: 50.8062

Решение, имеющее минимально возможную длину пути: 1 > 10
Длина пути: 6
    
```

Рисунок 4 - Вывод результата в консольном приложении Win32

ЛИТЕРАТУРА

1. Пацей Н.В. Основы алгоритмизации и программирования, Минск, 2010г.
- Буснюк Н.Н. Метод оптимального решения задачи о назначениях в сетевом планировании, Минск, 2016г.

УДК 004.413.5

студ. А.А. Сосновский

Науч. рук. доц., к.ф.-м.н. Н.Н. Буснюк

(кафедра информационных систем и технологий, БГТУ)

ПРОГРАММНОЕ СРЕДСТВО LOGIC BLOCKS ДЛЯ ОПЕРАЦИОННОЙ СИСТЕМЫ ANDROID

Скорость разработки и выпуска программных продуктов в современном мире играет значимую роль, так как конкуренция на рынке мобильных приложений (и не только мобильном рынке) высока, и то что не реализовано сейчас, может быть выпущено в ближайшее время. Кроме этого, необходимо минимизировать затраты производства. Поэтому, необходимо как можно быстрее внедрять новые функциональные возможности, получать от пользователей обратную связь и улучшать приложение. Этот круг, «разработка – тестирование – улучшение», разработчик проходит большое количество раз. Чем быстрее придет обратная связь от пользователей, тем большей статистикой будет обладать разработчик, и на основании этих данных сможет вести дальнейшую разработку программного продукта в ту или иную сторону.

Прототипирование является неотъемлемой частью разработки ПО, так как позволяет опробовать новые идеи (функциональные возможности, правки интерфейса и прочее), избегая больших временных и материальных затрат, сфокусировав внимание на главных деталях.

Прототип – это результат процесса прототипирования. Прототипы бывают разные: начиная от пользовательского интерфейса и заканчивая деревянной моделью кораблика. При разработке ПО обычно прототируют ту часть ПО, которую необходимо “потрогать в живую”, то есть испытать на конечном пользователе. Это дает понятие разработчику о том, как пользователь воспринимает данную идею, как изменяется его поведение.[1]

В качестве примера прототипирования рассмотрим проект Logic Block.

Logic Blocks – это мобильная логическая игра, в которой пользователю предстоит проходить уровни, решая уровень из блоков, то есть перемещая блоки таким образом, чтобы игровое поле опустошалось. Игровое поле представляет собой прямоугольную матрицу размером $M \times N$. Блоки представляют собой квадраты разных цветов. Менять местами можно только соседние блоки и строго по

вертикали или горизонтали. Если блоки одного цвета образуют специальную комбинацию, то эти блоки исчезают. Кроме этого, на игровое поле действует «гравитация», то есть блоки, под которыми нет других блоков, опускаются вниз, максимум до первой строчки в матрице. Если столбец матрицы становится пустым, то он убирается, а столбы по обе стороны от него соединяются. Пример игрового уровня представлен на рисунке 1.

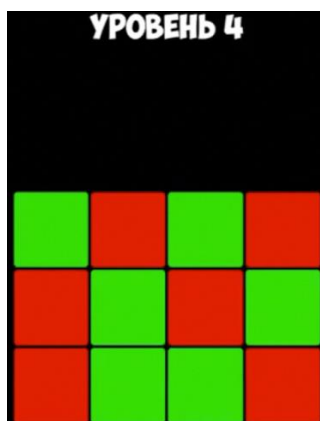


Рисунок 1 – Пример игрового уровня

Кроме основы игры, в современных играх реализовано огромное количество дополнительных функций, таких, например, как таблица рейтинга игроков (кто лучше, кто хуже), возможность поделиться результатом в социальных сетях, сохранение игрового прогресса и возможность продолжения игры с другого устройства, звуковое сопровождение, спецэффекты, достижения и масса других. Однако, не для всех игр это является обязательным, а точнее, коренным функционалом. Поэтому, в проекте Logic Blocks необходимо было реализовать только коренные функциональные возможности.

В Logic Blocks хотелось попробовать две идеи. Первая - игровые правила. Игровые правила – это сердце логических игр, так как от сложности и реиграбельности уровней (насколько интересно переигрывать один и тот же уровень), которые строятся на основе правил, зависит интерес к игре. Чтобы протестировать игровые правила, необходимо было разработать утилиту для генерации уровней (матриц), а также “решатель” уровней, чтобы убрать нерешаемые уровни из базы уровней.

Данный этап позволит создать скелет игрового процесса и проверить, насколько сложно или легко решать тот или иной уровень, а также нравится ли пользователю этот процесс или нет.

Вторая идея, которую хотелось попробовать, это добавить дополнительные параметры, которые могли бы мотивировать пользователя проходить последующие уровни. Например, это индикатор жизни, который уменьшается, если игрок не решил уровень.

При разработке идей необходимо пользоваться правилом «меньше делай, больше тестируй».

В ходе прототипирования иногда возникает необходимость улучшать некоторые параметры разрабатываемого функционала. В данном случае, необходимо было оптимизировать количество генерируемых уровней, так как при определенных параметрах их число было значительным (что требовало больших мощностных затрат и затрат на хранение), а количество необходимых из них уровней - малым. Однако не стоит доводить до идеала работу какой-либо функциональной возможности, так как это может потребовать огромное количество времени.

В примере выше с генерацией уровней изначально был применен алгоритм генерации последовательностей, который создавал цепочки определенной длины из комбинации нескольких цветов (размещения с повторениями). Данный алгоритм был оптимизирован следующими путями:

– если «шаблон» уровня повторялся, то его можно пропустить. К примеру, возьмем две строчки длиной 3 символа каждая, из двух символов - 1 и 0. Первая строка содержит 101, вторая 010. Эти строки принадлежат одному шаблону, так как если заменить символы второй строки на противоположные символы первой строки, то получится идентичная первой строка. Но если вторая строка содержит к примеру 100, то данная строка уже не подходит под шаблон первой строки. Таким образом, конечное число уровней можно уменьшить во столько раз, сколько цветов используется в уровне;

– если уровень содержит меньше символом одного типа, чем количество символов в минимальной комбинации, то данный уровень не решаемый и его следует пропустить;

– если уровень не решается «решателем» (программа, которая решает уровни автоматически), данный уровень пропускается.

На ряду с оптимизацией алгоритма генерации уровней был оптимизирован процесс записи конечных уровней в файл (к примеру, запись блока уровней в файл вместо записи каждого уровня отдельно), что дало прирост в производительности утилиты для генерации уровней.[2]

При разработке применялся принцип итерирования, который позволяет посмотреть на результат труда после выполнения небольшого объема задач. Это позволяет получать обратный отзыв от пользователей, что способствует улучшению качества приложения еще на этапе разработки, а не после выпуска.

Однако следует иметь в виду, что доступ к прототипу необходимо предоставлять ограниченному кругу лиц, потому что исход тестирования может повлиять на пользователя как в хорошую сторону, так и в плохую. Например, если пользователю не понравится какая-либо новая функциональность, он может перестать пользоваться приложением. Поэтому стоит вначале протестировать функциональность на небольшом (но достаточном) количестве пользователей, а дальше, по результатам решить, стоит ли давать доступ всем остальным пользователям.

Чтобы минимизировать временные расходы на разработку прототипа, целесообразно выбирать те инструменты, с которыми уже приходилось работать. Это поможет избежать лишних проблем при решении какой-либо задачи. Проблемы могут быть из-за:

- небольшого количества документации и примеров для выбранного инструмента;
- наличия ошибок внутри инструмента;
- сложности выбранного инструмента.

В качестве игрового движка был выбран Unity3D, для написания программного кода – язык программирования C#. Для разработки утилиты, позволяющей создавать игровые уровни, был использован язык программирования C, так как он обладает большей скоростью выполнения и дает полный контроль над памятью, нежели C#.

В заключение стоит отметить, что процесс прототипирования можно применять на практике и в учебных целях, к примеру, при написании курсовых и дипломных работ, чтобы провести первичный анализ выбранной темы, а также получить реальные данные с производства при эксплуатации прототипа. Это поможет выбрать наиболее оптимальный путь развития задуманной идеи.

ЛИТЕРАТУРА

1 Прототипирование // ВИКИПЕДИЯ информационный портал. – [Электронный ресурс]. –2017. – Режим доступа: <https://ru.wikipedia.org/wiki/Прототипирование>. – Дата доступа: 03.05.2017.

2 Основные формулы комбинаторики // МатБюро информационный портал. – [Электронный ресурс]. –2017. – Режим доступа: http://www.matburo.ru/tv_komb.php. – Дата доступа: 03.05.2017.

УДК 004.85

Студ. В.К. Сенюк

Науч. рук.: ст. преп. Е.А. Блинова

(кафедра информационных систем и технологий, БГТУ)

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ РЕДАКТИРОВАНИЯ И ПОИСКА КРАТЧАЙШИХ ПУТЕЙ ОБХОДА ГРАФА

При решении многих задач старая привычка толкает нас рисовать на бумаге точки, изображающие людей, население пункты, процессы и т.д., и соединять эти точки линиями или стрелками, означающими некоторые отношения. Такие схемы встречаются всюду под разными названиями: в экономике - диаграммы организации, в физике - электрические цепи, сети коммуникаций и т.д. Однако все эти понятия сводятся к одному – граф. Граф – это совокупность непустого множества вершин и наборов пар вершин (ребер). Теория графов является предметом анализа всего, что содержит в себе большое количество сложных связей.

Существует достаточно большое количество видов графов: мультиграфы (с кратными ребрами), псевдографы (с петлями), ориентированные (связи – направленные дуги) и неориентированные графы (вершины связаны ребрами) и т.д.

Одними из основных операций над графом являются – поиск кратчайших путей из одной вершины в другую, а также поиск максимального потока. Для реализации этих операций, в зависимости от вида графа и его свойств используются различные методы обхода. В результате поиска в ширину находится путь кратчайшей длины в невзвешенном графе, т.е. путь, содержащий наименьшее число рёбер. В результате поиска в глубину находится лексикографически первый путь в графе. Алгоритм Дейкстры предназначен для нахождения кратчайшего расстояния от одной из вершин графа до всех остальных, работает только для графов без рёбер отрицательного веса. Алгоритм Левита также предназначен для поиска кратчайших путей в графе, но отличается от алгоритма Дейкстры тем, что работает и для графов, которые содержат в себе ребра и положительного и отрицательного веса.

Методы и алгоритмы теории графов в настоящее время активно используются в различных областях науки и техники. Алгоритм Дейкстры нахождения кратчайшего пути между вершинами графов является основой навигации в различных информационных системах, в том числе телекоммуникационных и компьютерных сетях, включая глобальные, а также в системах мобильной связи и GPS-навигации. Широкое применение теория графов получила в таких областях как транспортные системы и маршрутизация данных в Internet. Новой областью приложений алгоритмической теории графов стала разработка и реализация параллельных алгоритмов и программ (грид-технологии, облачные вычисления и т.п.). Данная область связана с решением серьёзных задач на графах и сетях.

Актуальность данной темы подтолкнула к написанию программного продукта, основное окно которого представлено на рис. 1, с помощью которого граф можно задать визуально и затем преобразовать изображение в структуры данных (матрицы смежности, матрицы инцидентности и т.д.), с помощью которых можно реализовать алгоритмы обхода графа.

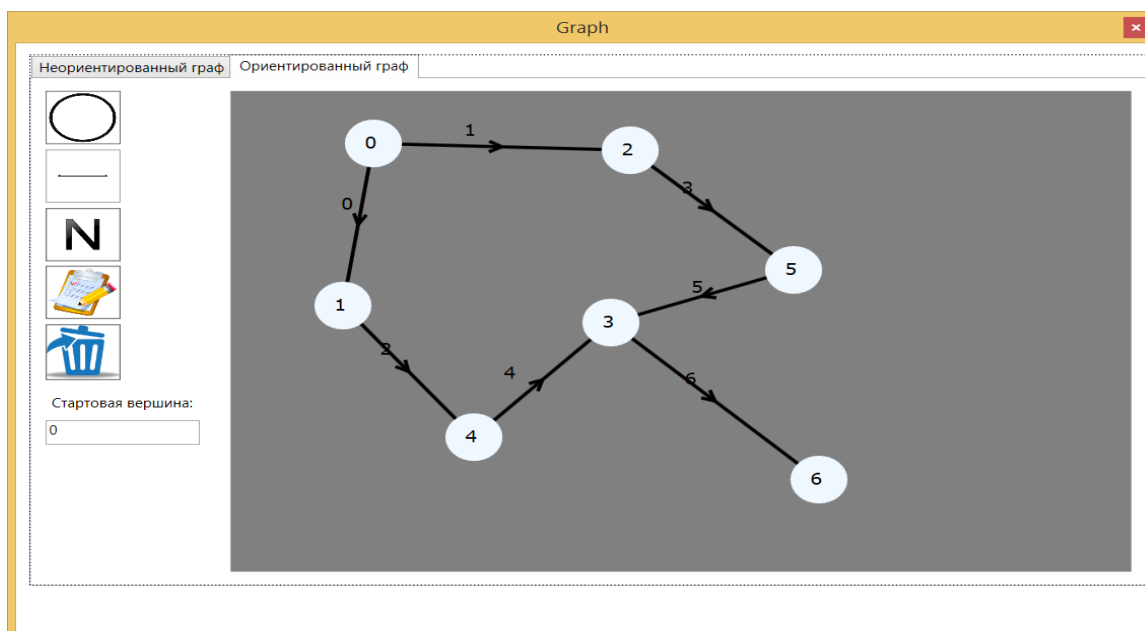


Рисунок 4 - Программный продукт

Программное средство включает в себя следующий ряд функций:

1. представление графа в виде матрицы смежности;
2. представление графа в виде матрицы инцидентности;
3. представление графа в виде списков смежности;

4. алгоритм DFS;
5. алгоритм BFS;
6. алгоритм Дейкстры;
7. определение степени вершин графа;
8. файловый вывод результатов методов обхода и сохранение изображения графа.

При кодировании граф был представлен в виде списка (List) экземпляров класса вершин и списка экземпляров класса ребер. Когда происходит выбор вершины мышью, поиск нужной вершины из списка осуществляется посредством перебора всех вершин и проверки условия принадлежности точки, в месте щелчка мыши, окружности вершины с помощью уравнения окружности. На основании этих методов и структур, путем перебора списков вершин и ребер заполнялись матрицы смежности и инцидентности, а также списки смежности. Далее, исходя из этих представлений, на графе реализовывались алгоритмы обхода.

Ознакомиться с исходным кодом проекта можно на моем github <https://github.com/Vlad-Senyuk>.

ЛИТЕРАТУРА

1. Домнин, Л.Н. Элементы теории графов. Пенза: Изд-во Пенз. гос. ун-та, 2007. 147 с.
2. Кристофидес, Н. Теория графов. Алгоритмический подход. Москва: Мир, 1978. 432 с.
3. Nathan, A. WPF 4. Unleashed. Indianapolis. Indiana 46240 US, 2010. 844 p.

УДК 004.77

Студ. Р.И. Белькевич

Науч. рук. ст. преподаватель Е.А. Блинова

(кафедра информационных систем и технологий, БГТУ)

ОБЗОР ГОТОВЫХ РЕШЕНИЙ ДЛЯ ЭМУЛЯЦИИ И ЗАМЕЩЕНИЯ БЭКЕНДА ВЕБ-ПРИЛОЖЕНИЙ

Front-end и back-end — термины в программной инженерии, которые различают согласно принципу разделения ответственности между представительским уровнем и уровнем доступа к данным соответственно. Front-end — интерфейс взаимодействия между пользователем и основной программно-аппаратной частью (back-end). Front-end и back-end могут быть распределены между одной или

несколькими системами. В работе представлены три актуальных отобранных готовых решения (Deployd, Feathers js, LoopBack от StrongLoop) для замещения бэкенда, предназначенные для тестирования взаимодействия веб-интерфейса и функционала сервера, сокращения затрат на написание бэкенда, сокращения времени разработки и упрощения жизни веб-разработчикам как в сфере front-end, так и в сфере back-end разработки.

Deployd — самый простой способ создать свой API. Deployd подразумевает использование готовой коллекции как ресурса, ресурсы добавляются через веб интерфейс панели управления. Можно развернуть приложение самостоятельно на любой сервис поддерживающий Node.js и MongoDB. С другой стороны, Deployd непригоден для публикации приложений.

Для создания приложения необходимо набрать следующее:

```
dpd create hello-world
cd hello-world
dpd -d
```

После этого в браузере откроется вкладка с панелью управления. Все доступные методы изображены на рисунке 1.

Task	Method	Route	Accepts	Returns
Listing data	GET	/test-stuff	Nothing	An array of objects
Creating an object	POST	/test-stuff	A single object	The saved object (or errors)
Getting an object	GET	/test-stuff/<id>	Nothing	A single object
Updating an object	PUT	/test-stuff/<id>	A single object	The saved object (or errors)
Deleting an object	DELETE	/test-stuff/<id>	A single object	Nothing

Рисунок 1 – доступные методы созданного API.

Пример использования:

```
function getAll (){
let xmlhttp=new XMLHttpRequest();
xmlhttp.open('GET','http://localhost:2403//test-stuff',false);
xmlhttp.send(null);
if(xmlhttp.status==200){
let result=JSON.parse(xmlhttp.responseText)
//ToDo }
}
```

Feathers.js — очень мощный и удобный фреймворк для создания серверных приложений на Node.js. Feathers.js предназначен для создания сервисов (REST, Socket.io и Primus), требует минимум усилий и доработки кода. Минусом является высокий порог вхождения.

Для создания приложения необходимо набрать следующее:

```
mkdir feathers-app
cd feathers-app/
feathers generate
```

После этого запустится генератор, с помощью которого будут установлены все необходимые настройки. Для использования необходимо сделать следующее:

Сервер:

```
const app = feathers();
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
app.configure(rest());
app.configure(socketio());
app.listen(3030);
```

Клиент (браузер):

```
var socket = io('http://localhost:3030');
var app = feathers();
app.configure(feathers.hooks());
app.configure(feathers.socketio(socket));
app.service('users').find({}).then(function(users){
    console.log('Users!', users);
});
```

LoopBack от StrongLoop — это полноценный Node.js-фреймворк для соединения приложений с данными через REST API. Позволяет разработчикам сосредоточиться на бизнес-логике, т.к. код для API генерируется автоматически. LoopBack последует идеологии open source и обладает высокой степенью расширяемости.

Для создания приложения необходимо набрать следующее:

```
npm install -g apiconnect
apic loopback
```

После завершения работы генератора по адресу localhost:8080/explorer получаем готовую документацию (рисунок 2):

/items		Show/Hide	List Operations	Expand Operations	Raw
POST	/items	Create a new instance of the model and persist it into the data source			
PUT	/items	Update an existing model instance or insert a new one into the data source			
GET	/items/{id}/exists	Check whether a model instance exists in the data source			
GET	/items/{id}	Find a model instance by id from the data source			
GET	/items	Find all instances of the model matched by filter from the data source			
GET	/items/findOne	Find first instance of the model matched by filter from the data source			
DELETE	/items/{id}	Delete a model instance by id from the data source			
GET	/items/count	Count instances of the model matched by where from the data source			
PUT	/items/{id}	Update attributes for a model instance and persist it into the data source			

Рисунок 2 – доступные методы.

Пример использования:

```
function getAll (){
let xmlhttp=new XMLHttpRequest();
xmlhttp.open('GET','http://localhost:8080/items',false);
xmlhttp.send(null);
if(xmlhttp.status==200){
let result=JSON.parse(xmlhttp.responseText)
  //ToDo  }
}
```

Если надо проверить, как будут выгружаться из хранилища данные и каким образом это будет отображаться на странице, достаточно использовать Deployd. Если планируется расширение проекта и требуется высокая надежность, лучше выбрать LoopBack. Если создаётся проект, не требующий серьёзной разработки, но все таки возможна какая-то кастомизация, то лучшим выбором является Feathers.js.

ЛИТЕРАТУРА

1. feathersjs.com [Электронный ресурс] Режим доступа: <https://docs.feathersjs.com/> — Дата доступа: 18.03.2017.

2. [deployd.com](http://docs.deployd.com/) [Электронный ресурс] Режим доступа: <http://docs.deployd.com/>— Дата доступа: 18.03.2017.

3. [loopback.io](http://loopback.io/doc/) [Электронный ресурс] Режим доступа: <http://loopback.io/doc/>— Дата доступа: 18.03.2017.

УДК 004.9

Студ. А.Н. Зайцев

Науч. рук.: ст. преп. Е.А. Блинова

(кафедра информационных систем и технологий, БГТУ)

ВЫБОР ОПТИМАЛЬНОГО АЛГОРИТМА ПОИСКА ПОДСТРОКИ В СТРОКЕ ДЛЯ РАЗЛИЧНЫХ ВИДОВ ДАННЫХ

В настоящее время довольно большой пласт информации представлен в текстовом виде: огромные библиотеки книг, архивы данных научных исследований. Одной из важнейших задач является поиск нужной информации за приемлемое время в этих огромных массивах данных. Целью данной научной работы является подбор оптимальных алгоритмов поиска текстовой информации в больших массивах данных для различных типов входных данных.

В ходе научной работы будут использоваться следующие обозначения: S — искомый текст (паттерн), T — текст, в котором ведётся поиск, N — длина S , M — длина T .

Самый простой существующий алгоритм работает по следующему принципу: идём слева направо по T и пытаемся найти такую позицию pos , где $T[pos... pos + N] = S$. Сравнение производится в цикле посимвольно. Асимптотика данного метода в худшем случае составляет $O(N * M)$. Надо отметить, что этот худший случай на реальных данных очень и очень редкий, так как для его выполнения требуется выполнение следующего условия: На каждом шаге алгоритма S должна полностью совпадать с подстрокой T , за исключением последнего символа. На самом деле алгоритм работает с гораздо лучшим ожиданием.

Вполне очевидным кажется развитие метода — сравнение строки не посимвольно, а с помощью хеш-функций, которые позволяют за $O(1)$ (в данной работе принимается, что асимптотика сравнения чисел процессором составляет $O(1)$) проверить на идентичность два числа. Данный алгоритм называется методом Рабина-Карпа [3]. Недостатком данного алгоритма является то, что подсчёт хеш-функции является довольно медленной операцией.

Следующий алгоритм поиска называется алгоритм Кнута-Морриса-Пратта (КМП) [2]. Данный алгоритм имеет асимптотику $O(N$

+ M) и имеет маленькую скрытую константу, что позволяет ему очень быстро работать. Алгоритм основан на работе префикс-функции. Префикс — функция от строки S (обозначается $p(S,i)$; где S — строка; i - длина префикса в S) — длина наибольшего префикса строки $S[1...i]$, который одновременно является её суффиксом.. Т.е. в строке S и её префиксе длиной i нужно найти такой префикс максимальной длины, который был бы суффиксом данной строки. Алгоритм Кнута-Морриса-Пратта очень хорошо себя зарекомендовал себя в поисках генетических подпоследовательностей, где размер алфавита крайне мал, а вот размер самих текстов велик.

Наиболее популярным на данный момент является алгоритм Бойера-Мура[4] и различные его модификации (алгоритм Бойера-Мура-Хорспула[7], турбо-алгоритм Бойера-Мура[4], оптимизированный алгоритм Бойера-Мура[8]). Алгоритм отличается от других тем, что, во-первых, сравнивает символы справа-налево, а не слева-направо. Также в алгоритме присутствует две стратегии перехода от одного потенциального вхождения подстроки к другой: правило сдвига хорошего суффикса и правило сдвига плохого символа. Вместе эти два правила дают очень хорошую скорость работы на реальных данных. Асимптотика алгоритма Бойера-Мура составляет в худшем случае $M * T$, но вероятность худшего случая крайне мала. Алгоритм является онлайн-овым, то есть не требует какой-либо предобработки.

Если же мы можем себе позволить преодобработать строки, то тут мы можем воспользоваться алгоритмом Ахо-Корасик[5], разработанным Альфредом Ахо и Маргарет Корасик в 1975 году. Алгоритм работает следующим образом: по искомым паттернам строится конечный автомат на структуре данных бор. Бор — это дерево с корнем в некоторой вершине R , причём каждое ребро дерева подписано некоторой буквой. К бору добавляются суффиксные ссылки, для эффективной работы конечного автомата. После построения мы можем запустить конечный автомат на тексте, в котором мы хотим осуществить поиск, и за $O(N)$ мы сможем найти все вхождения подстрок в данный текст. Алгоритм широко применяется, например, в антивирусах для поиска по сигнатурам. Недостатками данного алгоритма являются: высокое потребление памяти, что можно несколько исправить, используя для внутреннего представления вместо бора сжатый бор, который на каждом ребре дерева может хранить не просто символ, но и целую подстроку, оффлайновость алгоритма.

Также существуют алгоритмы, которые для поиска паттернов используют специально сформированные фильтры, позволяющие очень быстро перемещать окно поиска по тексту (например, алгоритм bitap algorithm[13]).

Отдельно стоит отметить группу алгоритмов (например, bit parallel length invariant matcher[12]) оптимизирующих операции сдвига окна при помощи битовой арифметики. Как известно, все числа внутри процессора представляют собой машинные слова. И если искомое число представляет собой одно машинное слово, то производительность повышается из-за более быстрых операций с этим словом, уменьшенным количеством обращений к оперативной памяти компьютера и кеш-промахам. Поскольку размер машинного слова на современных машинах ограничен 4-8 байтами, соответственно сдвиги будут сравнительно небольшими (здесь также играет роль, насколько плотно мы будем хранить информацию в машинных словах). Эти алгоритмы сильно специализированы для малых алфавитов, но показывают превосходную производительность.

Тестирование проводилось на следующем наборе: английские, итальянские и русские тексты произвольной, случайный набор символов с длиной строки $N = \{2, 4, 8, 16, 32, 64, 128, 250\}$ символов. Сами тесты были взяты из утилиты SMART[11]. Результаты тестирования показали следующее. Наивный алгоритм и наивный алгоритм Рабина-Карпа не предоставляют приемлемой производительности в задачах поиска. Алгоритм Рабина-Карпа может выигрывать в скорости только за счёт простой хеш-функции, но тогда сильно повышается вероятность возникновения коллизий, что приведёт к неверному результату поиска. Алгоритм Кнута-Морриса-Пратта показывает производительность близкую к алгоритму Бойера-Мура, но эвристики сдвига у префикс функции слабее, что негативно сказывается на общей скорости алгоритма. Алгоритмы группы Бойера-Мура отлично показывают себя в повседневных ситуациях: поиск в браузере, текстовом документе. Хотелось бы отметить не столько алгоритм Бойера-Мура, а дальнейшие его развития в сторону эвристик в различных направлениях: Чжу-Такаоки[4], Райта[6], Бойера-Мура-Хорспула, быстрый поиск[9]. Алгоритмы из группы конечных автоматов, к которым относится как ранее описанный алгоритм Ахо-Корасик, так и множество других алгоритмов, например, ЕВОМ[10], отлично показывают себя в ситуациях, когда мы заранее знаем набор паттернов, и меняется только текст, в котором следует произвести поиск.

ЛИТЕРАТУРА

1. String searching algorithm [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/String_searching_algorithm. — Дата доступа: 28.03.2017.
2. Knuth–Morris–Pratt algorithm [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Knuth%E2%80%93Morris%E2%80%93Pratt_algorithm — Дата доступа: 28.03.2017.
3. Rabin–Karp algorithm [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Rabin%E2%80%93Karp_algorithm. — Дата доступа: 28.03.2017.
4. Boyer–Moore string search algorithm [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Boyer%E2%80%93Moore_string_search_algorithm. — Дата доступа: 28.03.2017.
5. Aho–Corasick algorithm [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Aho%E2%80%93Corasick_algorithm. — Дата доступа: 28.03.2017.
6. Raita algorithm [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Raita_algorithm. — Дата доступа: 28.03.2017.
7. Boyer-Moore-Horspool algorithm [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Boyer%E2%80%93Moore-Horspool_algorithm. — Дата доступа: 28.03.2017.
8. Tuned Boyer-Moore algorithm [Электронный ресурс] / Institut delectroique et dinformatique Gaspard-Monge. — 2017. / Режим доступа: <http://www-igm.univ-mlv.fr/~lecroq/string/tunedbm.html>. — Дата доступа: 28.03.2017.
9. Quick search algorithm [Электронный ресурс] / Institut delectroique et dinformatique Gaspard-Monge. — 2017. / Режим доступа: <http://www-igm.univ-mlv.fr/~lecroq/string/node19.html>. — Дата доступа: 28.03.2017.
10. Extended backward oracle matching [Электронный ресурс] / Portale del Dipartimento di Matematica e Informatica. — 2017. / Режим доступа: <https://www.dmi.unict.it/~faro/smart/algorithms.php?algorithm=EBOM>. — Дата доступа: 28.03.2017.

11. SMART [Электронный ресурс] / Portale del Dipartimento di Matematica e Informatica. — 2017. / Режим доступа: <https://www.dmi.unict.it/~faro/smart/>. — Дата доступа: 28.03.2017.

12. Bit parallel length invariant matcher [Электронный ресурс] / Portale del Dipartimento di Matematica e Informatica. — 2017. / Режим доступа: <https://www.dmi.unict.it/~faro/smart/algorithms.php?algorithm=BLIM>. — Дата доступа: 28.03.2017.

13. Bitap algorithm [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Bitap_algorithm. — Дата доступа: 28.03.2017.

УДК 004.62

Студ. А.Н. Зайцев

Науч. рук.: ст. преп. Е.А. Блинова

(кафедра информационных систем и технологий, БГТУ)

АЛГОРИТМЫ ПРЕПРОЦЕССИНГА ИЗОБРАЖЕНИЙ ДЛЯ УЛУЧШЕНИЯ КАЧЕСТВА ОПТИЧЕСКОГО РАСПОЗНАВАНИЯ СИМВОЛОВ

Оптическое распознавание символов (англ. optical character recognition, OCR) — механический или электронный перевод изображений рукописного, машинописного или печатного текста в текстовые данные, использующихся для представления символов в компьютере (например, в текстовом редакторе)[1]. На данный момент лидером в области оптического распознавания является компания Abbyy. Они за долгие годы разработали крайне быстрый и эффективный механизм распознавания, недостатками которого являются только его цена и закрытость исходного кода.

В ходе работы были рассмотрены различные методы улучшения входного изображения для улучшения качества распознавания и их влияние на конечный результат. Определение качества распознавания — процент совпадения распознанного текста с текстом, который на самом деле находится на исходном изображении.

Рассмотрим методы улучшения качества распознавания.

Бинаризация. Для распознавания символов нам не нужна информация о цвете изображения, поэтому перед началом распознавания следует бинаризовать изображение, то есть привести его к чёрно-белому виду. Самым подходящим для большинства

изображений и эффективным по соотношению качество/скорость работы на данный момент является метод бинаризации Оцу[5].

Конвертация в серые тона. Для многих алгоритмов обработки изображений (например, детектор рёбер Кэнни[6]) информации из бинаризованного изображения мало, а информация о цвете лишняя. Тогда стоит применить конвертацию изображения в тона серого (обычно выбирают 8 битные изображения). Тут, также как и с алгоритмами бинаризации существует масса методов, но в основном применяется просто формула, где каждый канал входного изображения умножается на коэффициент [15].

Выравнивание. Крайне важно распознавать изображение, которое расположено ровно, иначе программа для распознавания не сможет корректно определить линии текста и результаты распознавания сильно ухудшаются. Для автоматического определения линий хорошо показывают себя алгоритм Хафа [8] и вероятностный алгоритм Хафа[9] для детектирования прямых. Также есть хорошо зарекомендовавший метод определения угла страницы по корешку книжного разворота: с помощью алгоритма Хафа можно определить местоположение корешка на фотографии, после этого определить угол поворота и повернуть страницу в противоположную сторону.

Устранение трапециевидных искажений. Если сделать фото страницы не под прямым углом к ней, то в результате у вас получится не прямоугольник страницы, а трапеция. В данном случае искажения будут такие, что, например, верхняя часть страницы будет отдалена и будет намного менее разборчива. Это также влияет на качество распознавания.

Устранение искажений «книжный разворот». Если фотографировать книжный разворот, то по центру книги, возле разворота, будут искажения. Также книжный разворот может давать неблагоприятную тень, которая крайне отрицательно сказывается на алгоритмах бинаризации (даже на алгоритме локальной бинаризации Оцу).

Устранение шумов и размытия. Чем больше на изображении шумов, нечётких краёв символов, тем меньше шансов на корректное распознавание символов. Следовательно, общее качество распознавания ухудшится. Для устранения шумов можно применять различные фильтры, такие как: билатеральный фильтр[12], фильтр Гаусса[11], медианный фильтр[10].

Гамма-коррекция, коррекция цвета, баланс белого. Изображение на входе может иметь цветовые искажения, что негативно скажется на

качестве бинаризации, а, следовательно, и на качестве распознавания. Существует алгоритмы автокоррекции, в том числе и реализованные в библиотеках с открытым исходным кодом (например, в библиотеке Ximgproc[13]).

Обрезание изображений. Также важной составляющей хорошего распознавания является то, какой процент входного изображения содержит текст. Например, если на вашей фотографии есть страница с текстом, но большая часть фотографии занята фоном, то это негативно скажется на качестве распознавания. Автоматизировать процесс обрезки очень хорошо получается с помощью алгоритма детектирования рёбер Canny. Также хорошо себя показывает алгоритм Хафа для детектирования линий.

Сегментация изображения. Входное изображение может быть сложным и содержать очень много лишней информации: изображения, текст со сложным форматированием (например, порядок колонок текста в журналах). Если мы сможем правильно сегментировать изображение, вырезать все области с текстом, сформировать их корректный порядок, то это значительно улучшит качество распознавания.

Удаление лишних деталей. Например, на вход у нас может подаваться таблица с данными. Линии таблицы могут пересекаться с буквами, что в свою очередь не позволит правильно распознать букву. В данном случае следует удалять такие линии с исходного изображения, но при этом сохранять информацию о форматировании для последующего форматирования выходного текста.

Обеспечение приемлемого DPI[7] (Dots Per Inch, точек на дюйм). При меньшем DPI качество распознавания падает. Чтобы обеспечить необходимый минимум DPI надо при необходимости уменьшать размер изображения. Например, Tesseract[4] рекомендует использовать 300 DPI.

Использование готовых наборов алгоритмов препроцессинга для различных видов изображений. Для каждого типа документов можно подобрать набор алгоритмов с нужными параметрами, которые значительно улучшают качество распознавания. Возможно, в данном случае здесь могут помочь нейронные сети, которые отлично справляются при должных мощностях с проблемами категоризации. Это очень действенный, но в то же самое время очень трудоёмкий метод. Получить такие наборы алгоритмов и константы крайне сложно. Данные, которые накапливают фирмы в ходе долгих лет исследований и тестирования, являются коммерческой тайной.

Влияние методов на качество распознавания. Так как программы для распознавания символов работают с бинаризованными изображениями, то очевидно, что применение бинаризации обязательно, при этом в разных случаях следует применять разные алгоритмы бинаризации, например, для старых текстов очень хорошо себя показывает Sauvola фильтр [14]. Следующим по необходимости идет выравнивание: если текст имеет некоторый наклон, то результаты распознавания будут очень плохие. Обрезание позволяет удалить значительную часть мусора, который может быть ложно распознан как текст. Удаление шумов, цветокоррекция, устранение различных искажений — в целом улучшают качество распознавания, но в меньшей мере, чем предыдущие методы. Наилучшего качества распознавания можно добиться, только применяя к изображению все вышеперечисленные методики.

ЛИТЕРАТУРА

1. Optical character recognition [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Optical_character_recognition. — Дата доступа: 25.03.2017.
2. Tesseract (software) [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: [https://en.wikipedia.org/wiki/Tesseract_\(software\)](https://en.wikipedia.org/wiki/Tesseract_(software)). — Дата доступа: 25.03.2017.
3. Image analysis [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Image_analysis. — Дата доступа: 25.03.2017.
4. Tesseract. Improve Quality [Электронный ресурс] / GitHub. — 2017. / Режим доступа: <https://github.com/tesseract-ocr/tesseract/wiki/ImproveQuality>. — Дата доступа: 25.03.2017.
5. Otsu binarization [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Otsu%27s_method. — Дата доступа: 25.03.2017.
6. Canny edge detector [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Canny_edge_detector. — Дата доступа: 25.03.2017.
7. Dots per inch [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Dots_per_inch. — Дата доступа: 25.03.2017.
8. Hough transform [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Hough_transform. — Дата доступа: 25.03.2017.

9. Probabilistic Hough transform [Электронный ресурс] / The university of Edinburgh. — 2017. / Режим доступа: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV1011/macdonald.pdf. — Дата доступа: 25.03.2017.
10. Median filter [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Median_filter. — Дата доступа: 25.03.2017.
11. Gaussian filter [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Gaussian_filter. — Дата доступа: 25.03.2017.
12. Билатеральный фильтр [Электронный ресурс] / Studopedia. — 2017. / Режим доступа: <http://studopedia.info/3-111442.html>. — Дата доступа: 25.03.2017.
13. Ximgproc [Электронный ресурс] / OpenCV. — 2017. / Режим доступа: <http://docs.opencv.org/3.0-beta/modules/ximgproc/doc/ximgproc.html>. — Дата доступа: 25.03.2017.
14. Sauvola binarization [Электронный ресурс] / Mathworks. — 2017. / Режим доступа: <https://www.mathworks.com/matlabcentral/fileexchange/40266-sauvola-local-image-thresholding>. — Дата доступа: 25.03.2017.
15. Grayscale [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: <https://en.wikipedia.org/wiki/Grayscale>. — Дата доступа: 25.03.2017.

УДК 004.451.9

студ. А. Ф. Булова, Е. И. Акшевская
Науч. рук. доц. Н. Н. Буснюк
(кафедра информационных систем и технологии, БГТУ)

BULL MILKMAN ИЛИ КАК СОЗДАТЬ ХОРОШИЙ ПРОТОТИП ИГРЫ ЗА 48 ЧАСОВ?

Главный вопрос: “Почему так быстро?!”. Весь прототип игры собирался в режиме хакатона, который проходил в Минске в марте 2017 г. Основной его задачей было собрать рабочую версию продукта за 48 часов. Так как большинство игр для мобильных платформ используют достаточно простую механику – за основу мы взяли Unity как игровой движок, и добавили качественной векторной графики.

Однако главной целью для нас было пройти полный цикл разработки: от идеи до прототипа.

Используемые технологии:

- 1) Основы программирования на C#.
- 2) Графический движок Unity.
- 3) Векторных редактор Illustrator.

Развитие прототипа. Согласно первоначальному замыслу, игроку предстоит примерить роль быка-молочника, отбиваясь от главных любителей молока – котов, норовящих украсть его груз. Чтобы это сделать необходимо пальцем «тапать», т.е. нажимать на экран, убирая препятствия и обезвреживая врагов. В конце уровня ждет цель путешествия в виде молочного завода, куда необходимо доставить ценный груз. Препятствия исчезают, а враги движутся в обратном направлении, не причиняя вред молоку. Так как геймплей призывает игрока постоянно быть сосредоточенным на устранении препятствий, анимацию (рис. 1) мы постарались сделать плавной, поддерживающей динамичный темп всей игры.

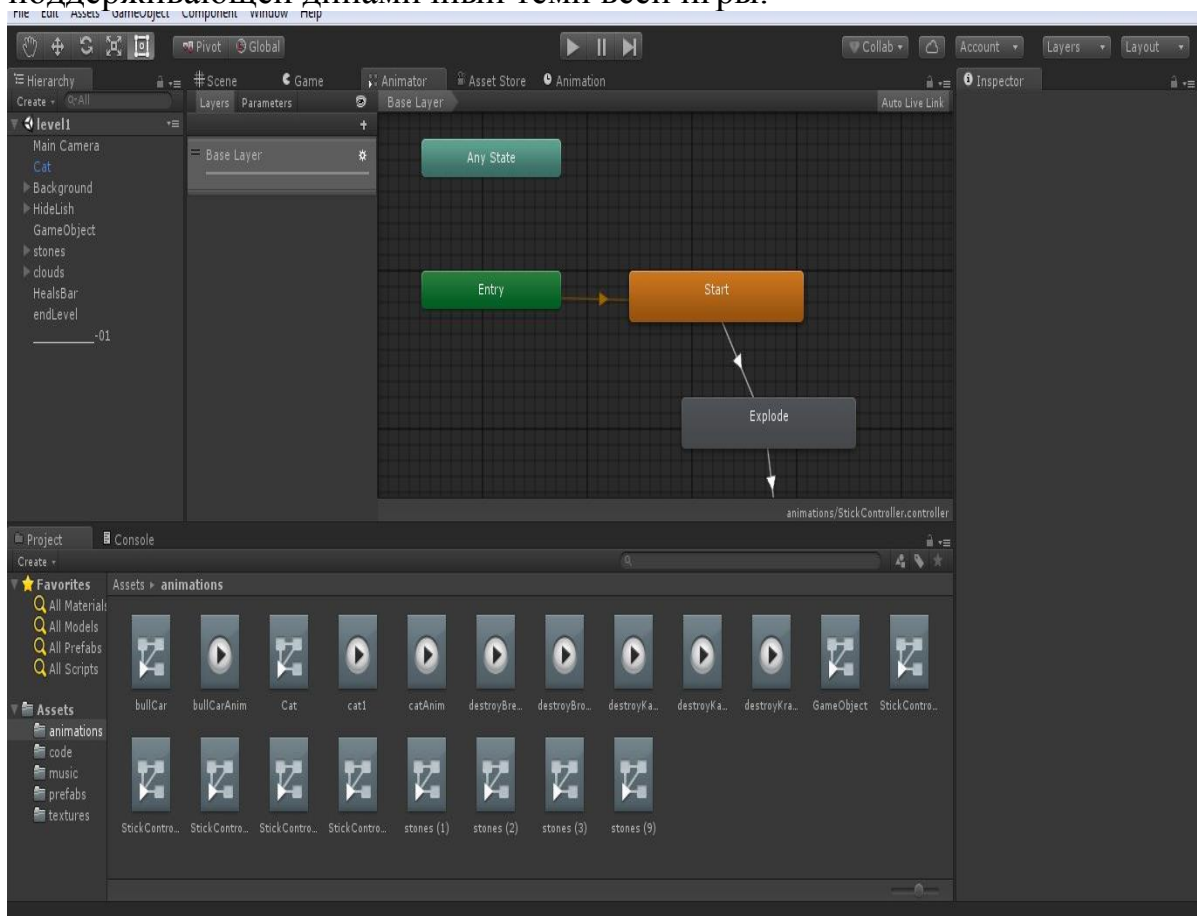


Рисунок 1 (Пример смены анимаций в компоненте «аниматор», Unity)

После того, как игровой процесс был определен, мы приступили к добавлению контента и оформления. Так за следующие пару часов появились: заставка в меню, несколько вариантов локаций, которые сменяли друг друга на разных уровнях, другие декорации, такие как кусты, деревья, горы, а также несколько видов врагов.

Графика и анимация. Все персонажи претерпевали изменения по сравнению с первоначальными образами, добавлялось все больше деталей. (рис. 2)



Рисунок 2 (Скриншот рабочей сцены из игры)

Оптимизация. Когда все было почти готово, несмотря на тесты на реальных девайсах, мы столкнулись с проблемой производительности, оптимизация заняла ощутимое количество времени. У нас были неполадки с пересечением коллайдеров, при которых персонаж менял траекторию пути, а также съедал две жизни, если враг успевал пересечь границу коллайдера главного персонажа, переходы анимаций по определенному событию (функции) в скрипте, а также мелкие баги, которые нам удалось оперативно исправить.

Для врагов мы также применили “запекание” физики в анимацию. На каждого врага было записано по 2 анимации. На деле выглядит почти так же, как и с применением симуляции физики.

Заключение. Таков был наш первый опыт разработки в подобном формате. (рис. 3, рис. 4) Это дало нам возможность погрузиться в рабочую атмосферу, где важно время и качество итогового продукта, а также дало неплохой опыт по поиску решений возникнувших задач и проблем, либо поиску обходных путей, что, по нашему мнению, важно в работе любого программиста. В свою очередь конференция в БГТУ позволила представить нашу работу на суд зрителей, в том числе преподавателей, и получить дополнительную оценку специалистов и опыт представления своих программ, а также толчок к развитию в этом направлении, улучшению навыков программистов и художников, и созданию в будущем новых игр.



Рисунок 3 (Скриншот готового прототипа игры. (начальный экран))

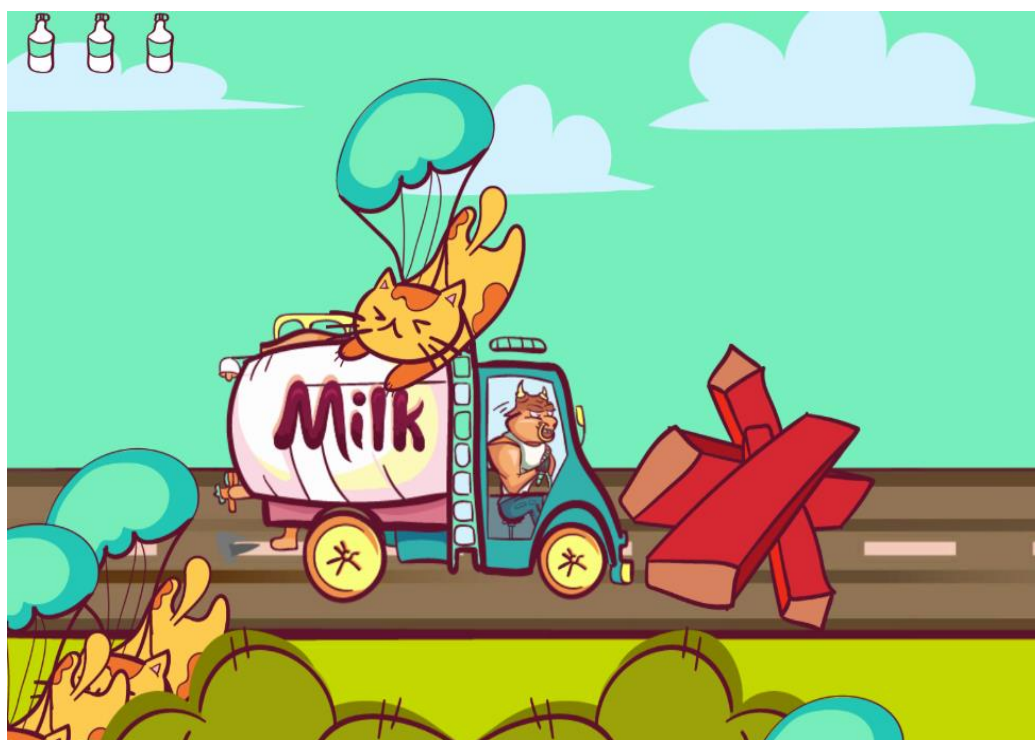


Рисунок 4 (Скриншот готового прототипа игры. (локация 1 уровня))

УДК 519.62 + 531.3 + 521.1

Студ. П. С. Шенец

Науч. рук. ст. преподаватель А. С. Наркевич
(кафедра информационных систем и технологий, БГТУ)

МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ ТЕЛ ПОД ДЕЙСТВИЕМ ФОНА ГРАВИТОНОВ

В 1930-х годах было обнаружено, что в скоплениях галактик отдельные галактики движутся заметно быстрее, чем это возможно, если их удерживает вместе притяжение светящейся массы скопления. Ф. Цвикки показал, что отношение необходимой массы к видимой составляет около 500 [1], т.е. большая часть массы не видна. Позже в спиральных галактиках наблюдения обнаружили слишком высокие скорости звезд на их периферии, причем эти скорости стремятся к постоянному пределу, зависящему от полной светимости галактики. Проблему недостающей массы (так ее назвали) можно решить, предположив, что существует некая темная материя, заполняющая галактики и их скопления [2]. Это принятая на сегодня гипотеза. Другой вариант – это возможная модификация уравнения движения

(II закона Ньютона) или закона тяготения. Наиболее известный пример – это модифицированная ньютоновская динамика (МОНД) М. Милгрёма [3].

В модели низкоэнергетической квантовой гравитации [4] взаимодействие тел с фоном гравитонов приводит к торможению движущихся тел в системе отсчета, в которой этот фон изотропен; при малых скоростях:

$$w = -H_0 c = 6,419 \cdot 10^{-10} \frac{M}{c^2}, \quad (1)$$

где H_0 – постоянная Хаббла, c – скорость света. Уравнение движения тела (материальной точки) в гравитационном поле большой массы M имеет вид [5]:

$$\ddot{\vec{r}} = -G \frac{M}{r^3} \vec{r} + w \left(\frac{\vec{u}}{u} - \frac{\vec{v} + \vec{u}}{|\vec{v} + \vec{u}|} \right), \quad (2)$$

где \vec{r} – радиус-вектор материальной точки, \vec{v} – ее скорость относительно центра притяжения, G – постоянная тяготения, \vec{u} – скорость центра притяжения системы относительно фона.

В этой работе, чтобы выяснить, может ли уравнение (2) быть связано с проблемой недостающей массы, разработана программа на C++ для численного решения этого уравнения, проведен анализ некоторых полученных решений. Алгоритм решения уравнения (2) основан на рекуррентных соотношениях:

$$\begin{cases} \vec{r}(t + \Delta t) = \vec{r}(t) + \vec{v}(t) \cdot \Delta t + \vec{a}(t) \frac{\Delta t^2}{2}; \\ \vec{v}(t + \Delta t) = \vec{v}(t) + \vec{a}(t) \cdot \Delta t; \\ \vec{a}(t + \Delta t) = -G \frac{M \vec{r}(t)}{r^3(t)} + w \left(\frac{\vec{u}}{u} - \frac{\vec{v} + \vec{u}}{|\vec{v} + \vec{u}|} \right). \end{cases} \quad (3)$$

В этой программе, данные были организованы в виде структуры. Структура содержит в себе 3 массива и 1 переменную. Так как мы рассматриваем частный случай движения в плоскости, то массивы содержат по две компоненты для численного описания таких величин, как радиус, скорость и ускорение по осям координат. Все вычисления производятся покомпонентно. В переменной записывается время. Все данные имеют тип double, так как мы работаем с большими числами, требующими большой точности при вычислениях.

Пользователь может выбрать желаемый режим расчёта и ввести исходные данные. Далее программа производит вычисления по выбранному варианту. Для дискретного моделирования

вычисляется период T , за который точка в классическом случае должна сделать полный круг. В переменную Δt записывается величина равная $T \cdot 10^{-6}$, чем задается количество точек за оборот, в которых производятся расчёты по формулам. В нашем случае берём 1.000.000 точек. Пользователь также может задавать количество периодов.

С помощью функции `clock()` библиотеки `<ctime>` определяем время работы программы. Полный цикл выполняется за 5-7 секунд на компьютере с процессором AMD A4-5100. Такое оптимально короткое время расчётов было достигнуто благодаря тому, что были использованы всего лишь 2 «буферные» переменные структурного типа, а не массив со всеми данными. Промежуточные результаты для 100 точек записываются в файлы. Это не загромождает память ненужными данными, так как для дальнейшего графического анализа нам достаточно 100 точек. Но впоследствии, для полноты картины и более точного построения графиков, это количество можно заменить на большее.

При работе создаётся 4 файла. В первый файл записываются длины векторов $r(t)$, $v(t)$ и $a(t)$. Во второй файл записываются поправки для r и v . В третий файл – координаты точек для последующего построения траектории, например, в MathCad.

Так же программа имеет функцию «дозаписи» - продолжения расчётов. Четвёртый файл нужен для того, чтобы при окончании вычислений записывать в него исходные данные и последнюю структуру, чтобы пользователь мог в любое время продолжить предыдущие расчёты и дозаписать новые данные в файлы.

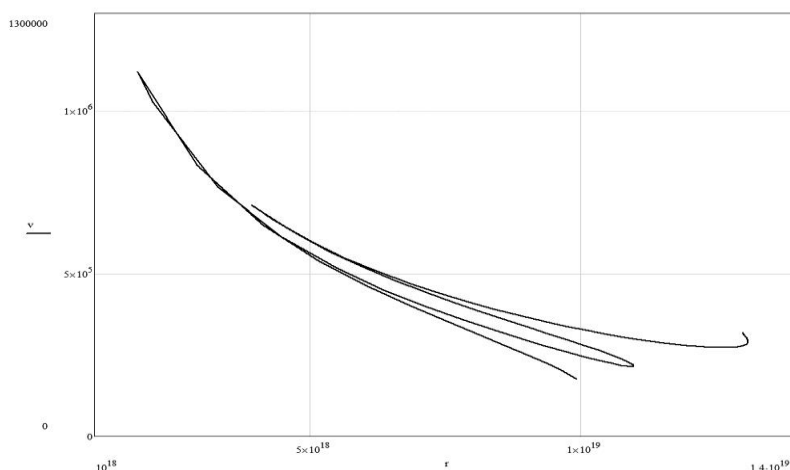


Рисунок 1 - Некеплеровская зависимость $v(r)$ для звезды в галактике при $w/a(0) = 0.082$

С помощью этой программы исследовано плоское движение материальной точки в поле притягивающего центра при малых (движение планеты в поле звезды) и больших (движение звезды на периферии галактики) значениях $w/a(0)$ и разных значениях u .

ЛИТЕРАТУРА

1. Zwicky, F. Die Rotverschiebung von extragalaktischen Nebeln. Helvetica Physica Acta 6, 110-127 (1933).
2. [https://ru.wikipedia.org/wiki/Тёмная материя](https://ru.wikipedia.org/wiki/Тёмная_материя).
3. Milgrom, M. ApJ, 1983, V.270, pp. 365–370 and 371–383.
4. Ivanov, M.A. Gravitons as super-strong interacting particles, and low-energy quantum gravity. In the book "Focus on Quantum Gravity Research", Ed. D.C. Moore, Nova Science, NY - 2006 - pp. 89-120; [hep-th/0506189].
5. Ivanov, M.A. Deceleration of Massive Bodies by the Isotropic Graviton Background as a Possible Alternative to Dark Matter [http://vixra.org/pdf/1108.0030v1.pdf].

УДК 004.451.9

Студ. А.А.Шавейко,

Научн. рук. ст. преподаватель А.С.Наркевич

(кафедра информационных систем и технологий ИСиТ, БГТУ)

ПРОГРАММНОЕ ПРИЛОЖЕНИЕ «РАЗВИТИЕ ПАМЯТИ»

Память (рисунок 1) – это навык, который является жизненно необходимым для каждого человека. Без нее мы бы никогда не были самими собой, не умели бы разговаривать и вообще не смогли бы мыслить. Но память является не только незаменимым навыком, но и важным элементом нашего образования и интеллекта. Развитие памяти, внимания и мышления часто представляет собой смежные задачи. От того, как развита наша память, напрямую зависят многие наши ментальные характеристики. Например, без



Рисунок 1 - Память

памяти не обойдется процесс овладения навыками скорочтения, ораторского мастерства и устного счета.

Для разработки данного программного продукта был использован объектно-ориентированный язык программирования C++ и визуальная среда разработки Microsoft Visual Studio 10. Выбор данной среды разработки позволяет быстро, качественно и эффективно разработать программный продукт любой сложности, при относительно минимальных затратах, если говорить о ресурсах персонального компьютера.

OpenGL (Open Graphics Library - открытая графическая библиотека) - спецификация, определяющая независимый от языка программирования платформонезависимый программный интерфейс для написания приложений использующих двумерную и трёхмерную графику. Это позволит в дальнейшем перенести приложение на другие платформы, (Linux, MacOS), т.к. стандарты библиотеки OpenGL везде одинаковые.

Этапы работы с программой:

- запуск программы;
- нажатие клавиши Enter;
- выполнение задания;
- просмотр результатов.
- выход из программы.

Интерфейс программы изображен на рисунке 2.

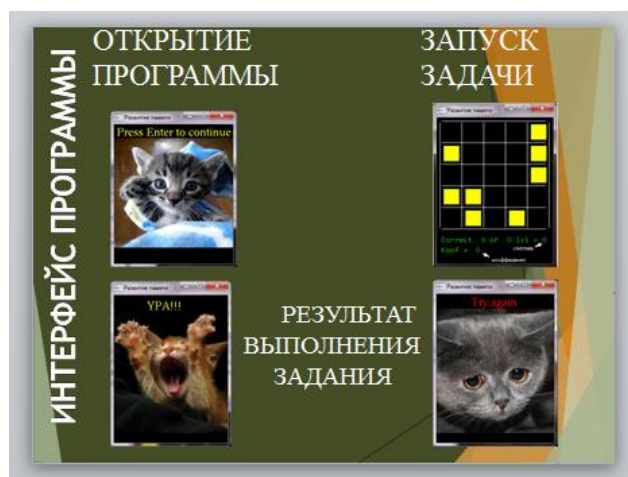


Рисунок 2 – Интерфейс программы

Задача заключается в повторении комбинации закрашенных областей после их потухания на экране.

При запуске приложения пользователь увидит заставку, ему предлагается нажать клавишу Enter для начала игры, в хаотичном порядке на игровом поле появляются желтые квадратики и через некоторое время исчезают, пользователь должен запомнить расположение желтых квадратиков и щелчком мыши указать на них, уровень считается пройденным, если правильно воспроизведена картинка.

При неправильном выделении квадрат будет выделяться красным цветом, при правильном выполнении он будет выделяться жёлтым цветом.

Выделять области можно до тех пор, пока количество выделений не будет равно количеству участков, которые были выделены при первичном появлении.

Если игрок неправильно указывает положение какого-либо квадратика, то место закрашивается красным, уровень считается не пройденным, и пользователь опускается на уровень ниже.

В процессе игры за каждый правильный ход игроку начисляются баллы, а в случае ошибки - баллы снимаются, при переходе на новый уровень дополнительно начисляется 100 баллов.

В ходе игры на экране также отображается счетчик: количество выполненных действий из максимально возможных на данном уровне.

В зависимости от того, с какими подробностями человек воспроизводит в памяти тот или иной визуальный образ, зрительная память бывает слабо развита, умеренно развита и развита в совершенстве.

Большая часть людей в этом отношении находится в пределах нормы, которой хватает для обычной жизни. Но есть сферы деятельности, где этого недостаточно. Специальным агентам, художникам, водителям, искусствоведам и т. п. нужна зрительная память выше средней.

Тренировать зрительную память, которая с годами только ослабевает, желательно всем. Если вы не хотите забывать лица знакомых, терять вещи, плохо ориентироваться на местности, регулярные тренировки зрительной памяти помогут вам сохранить ее.

УДК 519.686

Студ. Е.А. Богданович,
 Научн. рук. ст. преподаватель А.С.Наркевич
 (кафедра информационных систем и технологий, БГТУ)

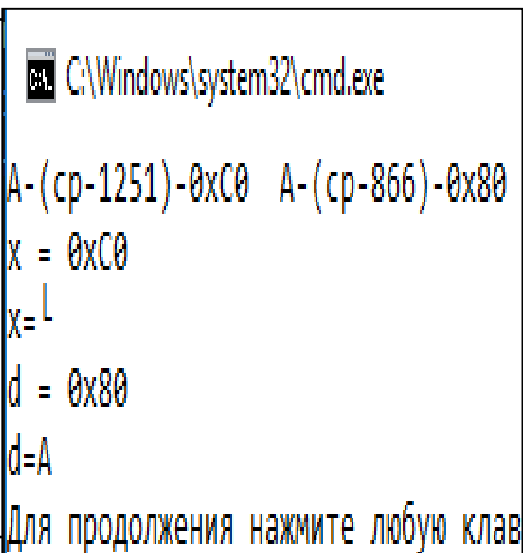
ЛОКАЛИЗАЦИЯ КОНСОЛЬНЫХ ПРИЛОЖЕНИЙ

При разработке консольных приложений возникает проблема их локализации. Проблема эта вызвана тем, что в средах разработки и выполнения программ используются разные кодовые таблицы.

Рассмотрим некоторые из них. Например, таблицу ASCII: American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). Однобайтовая схема кодирования позволяет работать только с 256 символами. Для представления символьных данных в кодировке ASCII используется символьный тип `char`, размер которого равен 1 байту. Данная кодовая таблица состоит из трёх частей. Первая часть: US-ASCII [1]. Символы с кодом `0x00-0x7F`. Данная таблица содержит различные знаки, буквы английского алфавита, а также цифры. Во всех кодовых таблицах коды данных символов одинаковы. Вторая часть: Windows1251 [2]. Символы с кодом `0x80-0xFF`. Данная кодовая таблица содержит различные знаки, а также буквы русского алфавита. Третья часть: CP866 [3] — символы с кодом `0x80-0xFF`. Данная кодовая таблица содержит множество различных знаков и русский алфавит.

Юникод (Unicode) [4] — двухбайтная схема кодирования позволяет работать с 65536 символами. Данная кодовая таблица содержит почти все символы всех письменных языков. Для представления символьных данных в кодировке Unicode используется символьный тип `wchar_t`, размер которого равен 2 байтам.

Рисунок 1- Отображение русских символов на экране

<pre>include<iostream> using namespace std; int main() { char x, d; cout << "A-(cp-1251)-0xC0 A-(cp-866)-0x80"<<endl; x = 0xC0; cout << "x = 0xC0" << endl; cout << "x="; cout << x; cout << endl; d = 0x80; cout << "d = 0x80" << endl; cout << "d="; cout << d; cout << endl; }</pre>	 <pre>C:\Windows\system32\cmd.exe A-(cp-1251)-0xC0 A-(cp-866)-0x80 x = 0xC0 x=L d = 0x80 d=A Для продолжения нажмите любую клавишу</pre>
---	--

В примере (рисунок 1) мы используем код большой русской буквы А из двух кодовых таблиц. Код символа А в кодировке Windows-1251 равен шестнадцатеричному числу 0xC0, а в кодировке CP-866 равен 0x80. Инициализируем созданные переменные кодами 0xC0 и 0x80. На экране отобразились два символа L и A. Просмотрев таблицы мы увидим, что код 0xC0 соответствует символу L в кодовой таблице CP-866. Такое отображение вызвано тем, что консоль имеет собственную настройку кодовой страницы CP-866. Visual Studio же, хранит содержимое сpp файлов и заголовочных файлов в кодировке Windows-1251.

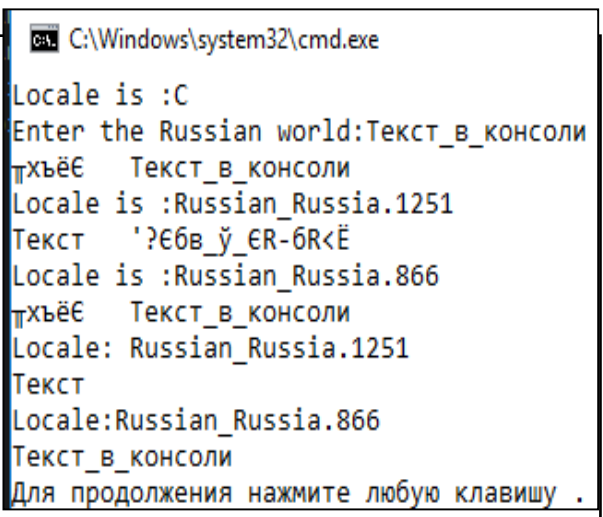
Средства локализации

Для учёта особенностей, связанных со страной и языком, используются специальные среды, так называемые локальным контекстом. Настройку программы на конкретный локальный контекст выполняет функция setlocale.

`char *setlocale(int category, const char *locale).`

Строки определяющие локальные контексты: C — используется по умолчанию английские буквы; de_DE — немецкий язык (Германия); de_AT — немецкий язык (Австрия); Rus, rus, Russian, Russian_Russia, Russian_Russia.1251, Russian_Russia.866 — русский язык (Россия).

Аргументы Category: LC_ALL — все категории, LC_COLLATE — сравнение и преобразование строк, LC_CTYPE — обработка символов, LC_MONETARY — формирование денежных сумм, LC_NUMERIC — формирование чисел, LC_TIME — формирование времени.

<pre>#include<iostream> #include<string> using namespace std; void main() { string text ="Текст"; string Text_entered_in_the_console; cout << "Locale is :"; << setlocale(LC_ALL, NULL)<<endl; cout << "Enter the Russian world:"; cin >> Text_entered_in_the_console; cout << text << "\t" << Text_entered_in_the_console << endl; cout << "Locale is :"; << setlocale(LC_ALL, "") << endl; cout << text << "\t" << Text_entered_in_the_console << endl; cout << "Locale is :"; << setlocale(LC_CTYPE, ".866") << endl; cout << text << "\t" << Text_entered_in_the_console << endl; cout << "Locale: " << setlocale(LC_CTYPE, ".1251") << endl; cout << text << endl; cout << "Locale:" << setlocale(LC_CTYPE, ".866") << endl; cout << Text_entered_in_the_console << endl;}</pre>	
<p>Рисунок 2</p>	

Первый вызов функции `setlocale(LC_ALL, NULL)` - настройка всех функций, `NULL` - использование стандартного локального контекста `C`.

Второй вызов функции `setlocale(LC_ALL, "")` - настройка всех функций, `""` - устанавливает локальный контекст с настройками операционной системы, где предусмотрена кодовая страница `Windows-1251`.

Третий вызов `setlocale(LC_CTYPE, ".1251")` - настройка обработки символов, `".1251"` - установка кодовой страницы `Windows-1251`.

Четвертый вызов `setlocale(LC_CTYPE, ".866")` - настройка обработки символов, `".866"` - установка кодовой страницы `CP-866`.

Локаль `".1251"` корректно отображает текст инициализированный в среде `vs`.

```
#include<iostream>
#include<string>
#include<locale>
using namespace std;
void main()
{
    locale locale_1;
    cout << "Locale is"
         << locale_1.name() << endl;
    string text = "Текст",
           Text_entered_in_the_console;
    cout << "Enter the Russian word:";
    cin >> Text_entered_in_the_console;
    cout << text << " "
         << Text_entered_in_the_console << endl;
    locale os("");
    locale::global(os);
    cout << "Locale is " << os.name() << endl;
    cout << text << " "
         << Text_entered_in_the_console << endl;
    locale locale_3("Russian_Russia.866");
    locale::global(locale_3);
    cout << "locale is " << locale_3.name() << endl;
    cout << text << " "
         << Text_entered_in_the_console << endl;
    locale locale_4("Russian_Russia.1251");
    locale::global(locale_4);
    cout << "Locale is " << locale_4.name() << endl;
    cout << text << endl;
    locale::global(locale_3);
    cout << "Locale is " << locale_3.name() << endl;
    cout << Text_entered_in_the_console << endl;}
```

```
cmd: C:\Windows\system32\cmd.exe
Locale is C
Enter the Russian word:ТехъёЕ текст_в_консоли
ТехъёЕ текст_в_консоли
Locale is
Текст в?ёбв_ў_ёR<ё
locale is Russian_Russia.866
ТехъёЕ текст_в_консоли
Locale is Russian_Russia.1251
Текст
Locale is Russian_Russia.866
текст_в_консоли
Для продолжения нажмите любую клавишу
```

Рисунок 3 - Отображение текста в консоли



В C++ локальный контекст реализуется с помощью библиотечного класса Locale (Рисунок 3). Первый шаг подключаем библиотеку locale. Далее с помощью конструкторов locale создаем объекты локального контекста. Locale locale_1- создаем стандартный локальный контекст C. С помощью метода name получаем название локального контекста. Locale os(“”) -создаем локальный контекст с настройками операционной системы, где предусмотрена кодовая таблица Windows-1251. С помощью функции global(os) устанавливаем локальный контекст глобальным. Locale locale_3(“Russian_Russia.866”) и Locale locale_4(“Russian_Russia.1251”). Устанавливаем кодовые страницы CP-866 и Windows-1251.

```
#include<iostream>
#include<Windows.h>
#include<string>
using namespace std;
void main()
{
    string text, text1="Текст";
    cout << "GetConsoleCP()-"
        << GetConsoleCP() << endl;
    cout << "GetConsoleOutputCP()-"
        << GetConsoleOutputCP() << endl;
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    cout << "Вывод инициализированного текста:"
    << endl;
    cout << text1<<endl;
    cout << "Введите текст:"<<endl;
    cin >> text;
    cout << "Вывод введенного текста:" << endl;
    cout << text;
    cout << endl;}
```

Рисунок 4

Рисунок 4 - Функции настройки консоли

Кроме локализации, возможность которой заложена в стандартах языков C и C++, существуют нестандартные средства. Для использования функций настройки консоли требуется подключить библиотеку Windows.h. В данной программе мы используем 4 функции. С помощью функций GetConsoleCP() и GetConsoleOutputCP()-получаем установленные кодовые страницы для ввода и вывода в консоли. А с помощью функций SetConsoleCP(), SetConsoleOutputCP() сами устанавливаем необходимую кодовую страницу.

В данном случае это Windows-1251. В результате программа отображает корректно все символы, введённые в консоли, либо инициализированные в программе.

В итоге мы рассмотрели 3 способа локализации консольного приложения. Самым простым и удобным, на мой взгляд, является последний, ведь там используется всего две функции, и текст, введённый в консоли, и текст из среды разработки будет отображаться корректно.

ЛИТЕРАТУРА

1. <http://vcampus.co/blogs/5190/ascii-table-with-html-entity> .
2. https://otvet.imgsmail.ru/download/b02dc38922ae1ad730945228f8b8349b_i-104.jpg
3. <https://allyslide.com/thumbs/d34770da0f90a71b4fa24a4bc57c3fbd/img>
4. <https://unicode-table.com/ru/#arabic-supplement> .
5. <http://www.c-cpp.ru/content/setlocale>

Студ. А. А. Худницкая

Науч. рук. ассист. Л. С. Мороз

(кафедра информационных технологий, БГТУ)

ИНТЕРНЕТ-ПРИЛОЖЕНИЕ ДЛЯ УЧЕТА ЛИЧНЫХ ФИНАНСОВЫХ РАСХОДОВ

Личное финансовое планирование позволяет любому человеку правильно оценить свое финансовое состояние и распределить ресурсы для получения большего дохода. Личное финансовое планирование – это полноценный бизнес-процесс, который включает несколько этапов:

- анализ текущей ситуации, составление личного финансового плана на базе первичной информации;
- составление личного финансового плана на базе уточненных данных;
- мониторинг выполнения личного финансового плана.

Как и любой бизнес-процесс, личное финансовое планирование имеет входы и выходы (результат). В качестве входа выступают имеющиеся активы и пассивы человека, текущие доходы и расходы и предполагаемый темп их изменения в будущем, финансовые цели, их предполагаемая стоимость и срок реализации.

Выходом, или результатом процесса «Личное финансовое планирование» является личный финансовый план. [1]

Для планирования личных финансов используют принципы: экономии, достаточности, работоспособности, исследования, приоритетности, отслеживания, скромной жизни, избегания долгов, упрощения, инвестирования, осторожности, сотрудничества [2].

Существуют основные способы учета:

Учет личных финансов на бумаге подразумевает записывание всех финансовых операций в блокноте или тетради. Он является довольно простым. Позволяет вести учет в любом доступном месте и не требует каких-либо технических средств и материальных затрат. Недостатком является необходимость самостоятельных подсчетов и отсутствие автоматизированной статистики и анализа.

Учет личных финансов в Excel и Word позволяют немного автоматизировать бумажный учет, но для этого уже необходимо иметь компьютер либо мобильное устройство и навыки для работы с ними. Также можно получить минимальную статистику, графики, диаграммы и их анализировать самостоятельно. Недостатком является вероятность сбоя и потери информации.

Учет личных финансов с помощью программ домашней бухгалтерии представляют собой более серьезное программное средство, которое содержит много возможностей для анализа. При всем широком функционале, такие программы являются не всегда бесплатными и не всегда содержат понятный интерфейс.

Учет личных финансов онлайн позволяют вести учет из любого места, где есть Интернет и хранят информацию на серверах или же облачных сервисах. Но при их использовании есть зависимость от сети и работы сервиса, а также риск несанкционированного доступа к информации. [3], [4]

Преимущественным является компьютерный учет личных финансов, который предоставляет следующие возможности.

Формирование отчетов. Компьютер позволяет быстро сформировать нужные вам отчеты для дальнейшего анализа. Анализ вашего бюджета позволяет разрабатывать стратегию и вырабатывать план дальнейших действий, либо вносить корректировки в уже существующий план.

Ведение нескольких типов счетов. Вы можете вести несколько счетов. Вся информация отображается на экране ПК, что очень удобно.

Планирование ваших финансов. Отличная возможность вносить будущие платежи и предполагаемые расходы. Планирование позволит вам рассчитывать свои действия на несколько шагов вперед.

Расчет кредитов и вкладов. Многие программы имеют встроенные калькуляторы для расчета кредита/вклада.

Контроль за долгами. Вводите своих должников, для точного отражения ваших финансовых дел. Эта функция также помогает не забыть кому и сколько вы заняли.

Защита данных. Все свои данные вы сможете скрыть от лишних глаз надежным паролем.

Напоминание о необходимых платежах. Заплатить за квартиру, погасить кредит, вернуть долг, купить подарок – чтобы ничего не забыть, программа по ведению семейного бюджета напомнит вам об этом. [6]

Для реализации основных возможностей было решено разрабатывать интернет-приложение с помощью на платформе ASP.NET MVC предоставляющий собой фреймворк для создания сайтов и веб-приложений с помощью реализации паттерна MVC используя HTML, CSS и JavaScript.

Концепция паттерна (шаблона) MVC (model - view - controller) предполагает разделение приложения на три компонента:

Контроллер (controller) представляет класс, обеспечивающий связь между пользователем и системой, представлением и хранилищем данных. Он получает вводимые пользователем данные и обрабатывает их. И в зависимости от результатов обработки отправляет пользователю определенный вывод, например, в виде представления.

Представление (view) - это собственно визуальная часть или пользовательский интерфейс приложения. Как правило, html-страница, которую пользователь видит, зайдя на сайт.

Модель (model) представляет класс, описывающий логику используемых данных.

Авторизация и аутентификация осуществляется с помощью системы ASP.NET Identity.

Для работы с данными в ASP.NET MVC использовался фреймворк Entity Framework, представляющий собой более высокий уровень абстракции, который позволяет абстрагироваться от самой базы данных и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работаем с объектами.

Адаптивность веб-приложения была создавалась с помощью фреймворка Bootstrap, который по умолчанию содержится в ASP.NET MVC. [6]

Интернет-приложение предоставляет собой веб-приложение «Учёт личных финансов», состоящее из основных страниц: счета, операции, категории, статистика, а также панель доступа к параметрам учетной записи.

Название	Тип операции	Категория	Дата	Сумма	Валюта	
Кошелек	Доход	Зарплата	14.04.2017 0:00:00	20,00	BYN	Подробности Редактировать
Зарплата	Доход	Зарплата	14.04.2017 0:00:00	20,00	BYN	Подробности Редактировать
Зарплата	Расход	Платежи	15.04.2017 0:00:00	5,00	BYN	Подробности Редактировать
Кошелек	Расход	Покупки	16.04.2017 0:00:00	10,00	BYN	Подробности Редактировать
Зарплата	Доход	Зарплата	17.04.2017 0:00:00	40,00	BYN	Подробности Редактировать
Карта	Расход	Валюта	18.04.2017 0:00:00	30,00	USD	Подробности Редактировать
Карта	Расход	Платежи	18.04.2017 0:00:00	1,00	USD	Подробности Редактировать

Рисунок 5 – Просмотр операций

Приложение позволяет вести учет финансов в основных валютах (BYN, RUB, USD, EUR), используя несколько счетов различных типов (наличные, банковская карта, банковский счет), и классификация операций с помощью категорий.

Статистика предоставляется в виде диаграмм, с выбором временного периода, счета и категории.

Для переноса данных был реализован экспорт операций в файл формата CSV.

ЛИТЕРАТУРА

1. Управление личными финансами / Под ред. А. В. Кочеткова. — М.: ПЕР СЭ, 2008. — 624 с
2. Планирование и учет финансов [Электронный ресурс]: URL: <https://4brain.ru/finance/plan.php>.
3. Зачем нужно вести учет личных финансов? | myfinsyst.ru [Электронный ресурс]. URL: <http://myfinsyst.ru/personal-finance/zachem-vesti-uchet-lichnyh-finansov>
4. Как вести учет личных финансов? | Финансовый гений [Электронный ресурс]: URL: <http://fingeniy.com/kak-vesti-uchet-lichnyh-finansov>

5. ВЕДЕНИЕ СЕМЕЙНОГО БЮДЖЕТА. Пример того, как и где вести семейный бюджет [Электронный ресурс]: URL: <http://www.myrouble.ru/kak-i-gde-vesti-semejnyj-byudzhet/>

6. Полное руководство по ASP.NET MVC 5, создание веб-приложений и веб-сайтов на языке программирования C# [Электронный ресурс]: URL: <https://metanit.com/sharp/mvc5/>

УДК 004.415

Студ. П.Е. Жаворонок

Науч. рук. ст. преп. И.Г.Сухорукова

(кафедра информационных систем и технологий, БГТУ)

СЕРВИС БРОНИРОВАНИЯ БИЛЕТОВ МАРШРУТНЫХ ТРАНСПОРТНЫХ СРЕДСТВ «MARSHRUTKA.BY»

На сегодняшний день в мире насчитывается несколько сотен авиакомпаний с тысячами воздушных судов, по маршрутам курсируют множества поездов, кораблей и, наконец, автобусов дальнего следования. В середине XX века перед транспортниками встала серьезная проблема распределения ресурса мест на транспортных средствах с минимальными затратами времени, то есть разгрузить потоки очередей в билетных кассах.

Для решения данной проблемы были придуманы специальные системы бронирования билетов. Помимо возможности выбора направления, даты, стоимости маршрута, они также дают шанс поучаствовать в специальных акциях компаний, отдать приоритет самым дешёвым альтернативам. Именно с помощью интернета покупка недорогих билетов становится простой и быстрой.

Бронирование билета – это процесс закрепления за конкретным пассажиром места на нужный рейс. Бронирование делается бесплатно. Результатом процедуры бронирования является возможность закрепить место на одном или двух рейсах разных компаний и затем в течение нескольких дней определиться в выборе.

В настоящее время существует огромное количество сайтов и сервисов по бронированию билетов на маршрутные такси. Все они обладают схожими недостатками, основной проблемой существующих сервисов по бронированию билетов является:

- отсутствие онлайн бронирования;
- перевозка только по одному направлению;

- бронирование билетов только у одного перевозчика;
- отсутствие адаптивного дизайна либо мобильной версии;
- сложный процесс бронирования билетов.

«Marshrutka.by» – современная и удобная система бронирования билетов на междугородние маршрутные такси. С ее помощью можно быстро забронировать билет на маршрутное такси по любому маршруту. В системе предоставлено огромное количество маршрутов как по территории Республики Беларусь, так и за её пределами.

Онлайн система бронирования билетов на маршрутные такси «Marshrutka.by», дает возможность в режиме реального времени получить информацию о маршрутах и расписании движения маршрутных такси, наличии свободных мест, сравнить цены нескольких перевозчиков, не выходя из дома.

Бронирование билетов происходит в несколько кликов что даёт возможность пользоваться системой без особого профессионализма. Базовых навыков обращения с компьютером и всемирной паутиной вполне достаточно чтобы забронировать билет. В ходе оформления брони у вас всегда есть возможность получить консультации диспетчеров. Так же у пользователя есть возможность в любое время просмотреть забронированные билеты в личном кабинете, посмотреть историю поездок, либо отменить бронь.

Целевой аудиторией данного сервиса является:

- пассажиры;
- перевозчики.

Достоинства бронирования билетов с помощью сервиса «Marshrutka.by»:

- любые направления. Наша система бронирования билетов работает без географических ограничений. С нами вы можете покупать билеты на сотни направлений и свободно путешествовать как по РБ, так и за её пределы;

- предложения от десятков перевозчиков. На нашем сайте можно забронировать места на рейсы отечественных и зарубежных компаний. Вы сами выбираете наиболее подходящее предложение по заданному маршруту из нескольких перевозчиков;

- профессиональное консультирование. Вы всегда можете обратиться к нашим диспетчерам с вопросами о брони билетов и не только. Они не только дадут консультации, но и помогут провести необходимые действия;

- привлекательные цены. Бронирование билетов через Интернет удобно и выгодно. У нас можно сравнить цены в разных

компаниях для выбора наиболее дешевого варианта. Кроме этого, наши клиенты регулярно участвуют в акциях и пользуются специальными предложениями;

- информативность. Всю информацию о компании, водителях, количестве свободных мест, можно узнать прямо на сайте;

- удобство. Билет всегда находится в одном месте. В любое время и в любом месте его можно посмотреть на сайте и для этого требуется всего лишь доступ к интернету и телефон. Даже если вы потеряли телефон у водителя всегда есть список пассажиров, зарегистрированных на данный рейс.

Достоинства использования сервиса «Marshrutka.by» для компаний:

- один диспетчер. Система бронирования билетов «Marshrutka.by» полностью автономна и не требует большого числа диспетчеров, всего один диспетчер может справиться с приёмом заказов как по телефону, так и с подтверждение бронирования билетов онлайн;

- удобство использования. Администратор компании в любое время может добавить новый маршрут, сменить водителя или посмотреть список забронированных мест.

- система отзывов. На каждого водителя можно оставить отзыв, что позволит информировать работодателя о компетентности водителя;

- реклама. Система «Marshrutka.by» является дополнительной рекламной площадкой, что позволит компаниям увеличить количество пассажиров и уменьшить расходы.

Достоинства использования сервиса «Marshrutka.by» для водителей:

- практичность. В любое время можно посмотреть список предстоящих маршрутов и список пассажиров, забронировавших билеты на эти маршруты;

- моментальное оповещение пассажиров. В любое время и всего за несколько кликов водитель может оповестить пассажиров о задержке рейса или непредвиденной ситуации.

Описание и основные преимущества данного сервиса представлены на рисунке 1.

Ключевые партнеры	Ключевые виды деятельности	Предоставленная ценность	Взаимоотношения с клиентами	Клиенты
<ul style="list-style-type: none"> частные перевозчики компании по перевозке пассажиров 	<ul style="list-style-type: none"> предоставление информации о поездках бесплатный сервис для онлайн-бронирования 	<p>Для пассажиров:</p> <ul style="list-style-type: none"> удобный принцип формирования маршрутов управление поездками на маршрутках интегрированная система скидок раннее бронирование высокий сервис <p>Для перевозчиков:</p> <ul style="list-style-type: none"> уменьшение риска реклама самосовершенствование компании увеличение аудитории 	<ul style="list-style-type: none"> автоматизированное обслуживание система отзывов 	<ul style="list-style-type: none"> перевозчики пассажиры

Рисунок 1 – Основные достоинства сервиса «Marshrutka.by»

УДК 004.65

Студ. А. Ю. Третьян, Н. С. Каргин,
 Науч. рук. ст. преп. Е.А. Блинова

(кафедра информационных систем и технологий, БГТУ)

ВЫБОР И ТЕСТИРОВАНИЕ NOSQL БАЗ ДАННЫХ

NoSQL (not only sql) — термин, обозначающий подходы, направленные на хранение и реализацию баз данных, отличных от реляционных СУБД. NoSQL решения используются в местах, где необходимо хранение больших объемов данных, линейная масштабируемость, кластеры, отказоустойчивость.

Под NoSQL понимают множество разнородных систем, однако есть набор характеристик, свойственных данному стеку технологий:

- не используется SQL;
- неструктурированные;
- представление данных в виде агрегатов;
- слабые требования к транзакционной системе.

В зависимости от модели данных и подходов выделяют четыре типа хранилищ: key-value, document store, column database, graph database.

При выборе NoSQL не стоит ориентироваться на популярность базы данных и их рекламу. Основными параметрами баз данных такого типа являются latency, throughput, то есть ожидаемое время исполнения запроса и количество запросов в секунду, которое может обработать база данных, соответственно. Также важным параметром является memory footprint — разница между реальным объемом занесенных данных и объемом, которую база данных занимает.

Для тестирования баз данных хорошо подходит утилита YCSB (Yahoo Cloud Serving Benchmark), так как с помощью неё легко реализовать свои драйверы для других баз данных, исключая стандартные, профиль нагрузок, является отраслевым стандартом. У YCSB существуют стандартные профили нагрузок (Workload), учитывающие основные базовые ситуации:

- A: 50% запросов на чтение, 50% на запись;
- B: 95% запросов на чтение, 5% на запись;
- C: 100% запросов на чтение;
- D: 100% запросов на чтение, 5% на добавление;
- E: 95% запросов типа scan, 5% на запись;
- некоторые другие профили.

У каждой NoSQL базы данных существует свой собственный способ представления данных и допустимых операций над ними, поэтому в YCSB используется максимальное обобщение. Набор данных, которыми оперирует YCSB — ключ (64-битный хеш), значение (десяток полей случайных бинарных данных).

Тест YCSB представляет собой выполнение определенного набора операций в разных режимах в зависимости от профиля нагрузок. Исходным результатом являются журналы, содержащие следующую информацию:

- средняя, минимальная, максимальная задержка;
- задержка, в которую уложились 95% и 99% операций;
- количество операций определенного типа;
- общее время теста;
- общее количество всех операций и среднее количество операций в секунду.

Самыми важными параметрами являются средняя задержка и количество операций в секунду. Необходимо проанализировать скорость вставки на SSD, память, производительность при интенсивной записи на SSD и в память. Самыми быстрыми NoSQL базами данных оказались хранилища типа key-value, что подтверждается результатами тестирования пропускной способности и средней задержки при модификации данных, результаты которых представлены на рисунках 1-11.

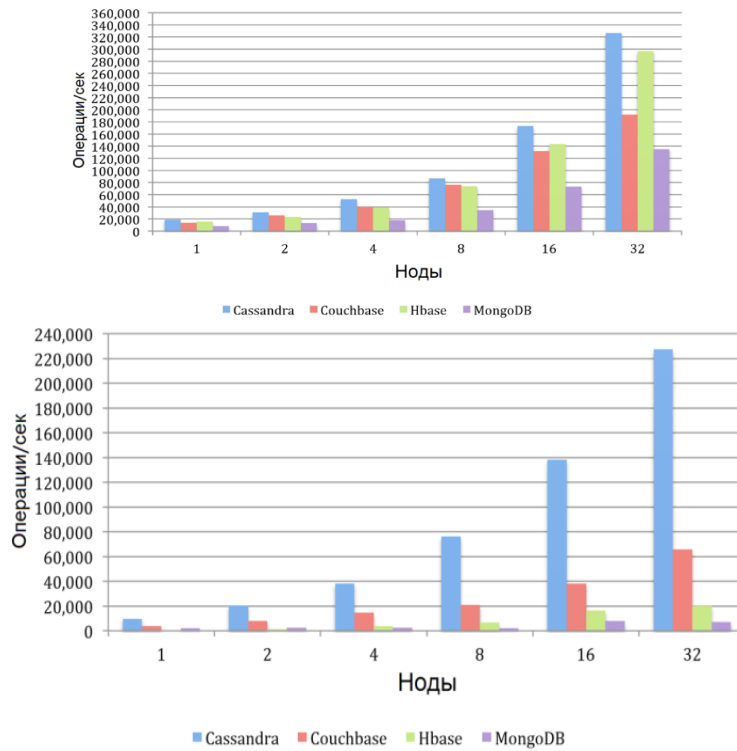


Рисунок 1 и 2 – Load process и Read-Mostly workload соответственно

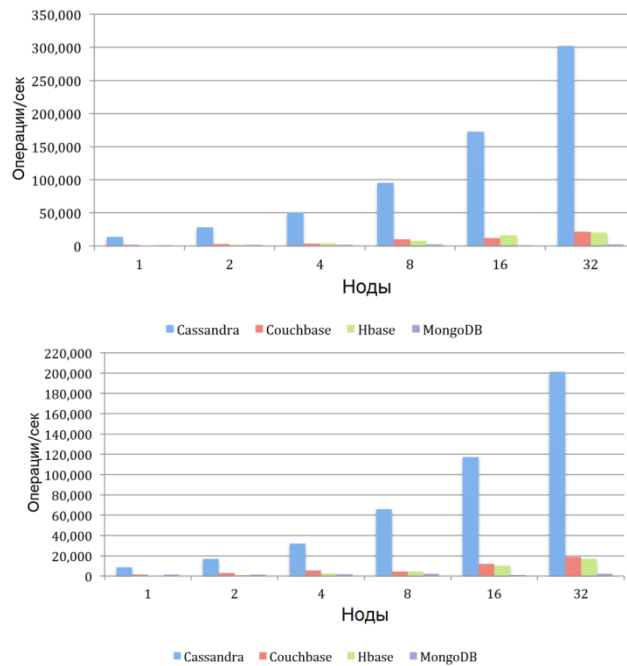


Рисунок 3 и 4 – Balanced Read/Write Mix и Read-Modify-Write Workload соответственно

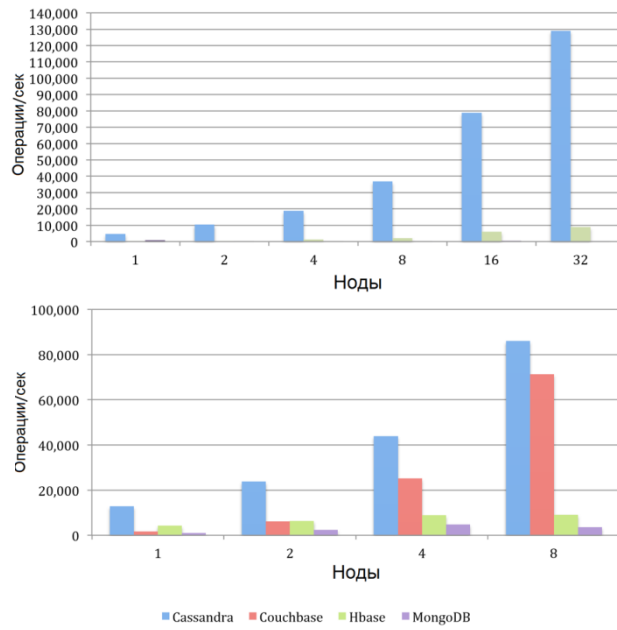


Рисунок 5 и 6 – Mixed operational and Analytical и Insert-mostly Workload соответственно.

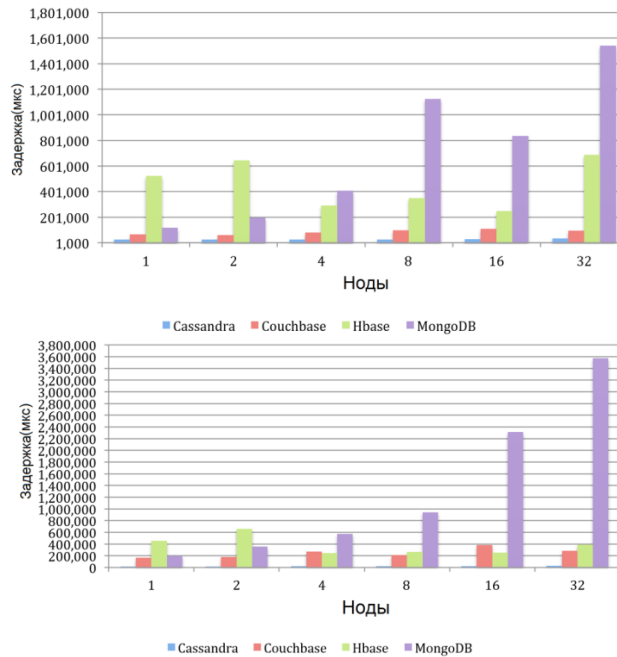


Рисунок 7 и 8 – Balanced Read/Write Mix и Read-Modify-Write Workload соответственно

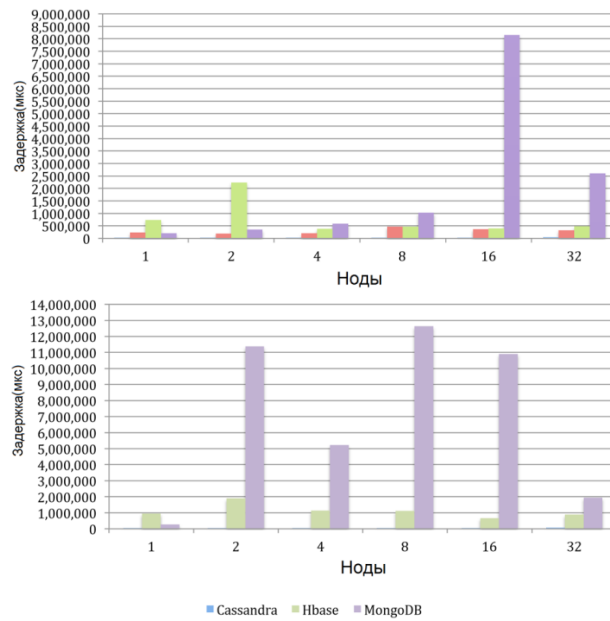


Рисунок 9 и 10 – Read-Modify-Write и Mixed Operational and Analytical Workload соответственно

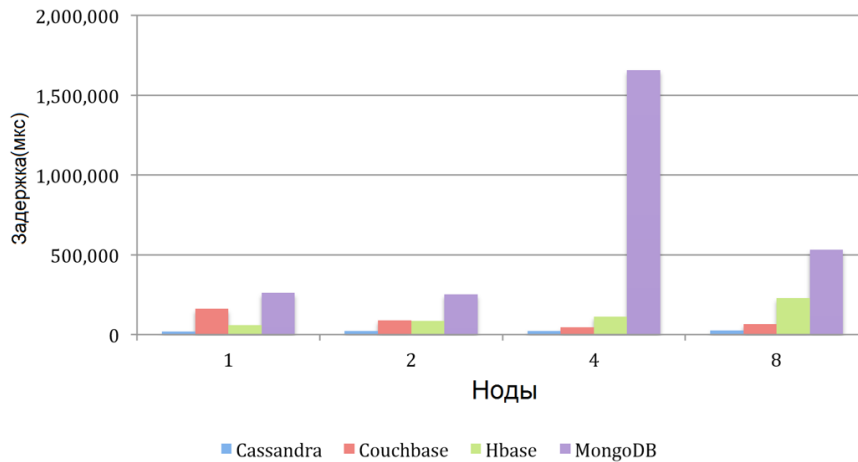


Рисунок 11 – Insert-Mostly Workload
ЛИТЕРАТУРА

1. NoSQL базы данных: понимаем суть [Электронный ресурс] / habrahabr.ru — 2017. / Режим доступа: <https://habrahabr.ru/post/152477/>. — Дата доступа: 05.05.2017.
2. A real comparison of NoSQL Databases HBase, Cassandra & MongoDB [Электронный ресурс] / <https://www.linkedin.com> — 2017. / Режим доступа: <https://www.linkedin.com/pulse/real-comparison-nosql-databases-hbase-cassandra-mongodb-sahu>. — Дата доступа: 05.05.2017.

УДК 004.415

студ. О.В. Луцевич

Науч. рук. доц. А.И. Бракович

(кафедра информационных систем и технологий, БГТУ)

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ РЕАЛИЗАЦИИ АЛГОРИТМОВ ПОИСКА НА ПРИМЕРЕ ШАХМАТНОЙ ИГРЫ

Машина играет настолько умно, насколько ее научит человек. Есть простые игры, типа "Крестиков-ноликов", с ограниченным количеством ходов. В таких играх машину ничему не нужно учить, она легко просчитывает ситуацию до конца и играет идеально. В шахматах количество ходов намного больше, и дерево перебора огромно. До конца просчитать ситуацию, кроме некоторых редких случаев, просто невысказимо. Как же машина играет и выбирает лучший ход?

1951 году, математик Клод Шеннон пишет свою первую статью о программировании шахмат. Он писал: «Хотя, возможно, это и не имеет никакого практического значения, сам вопрос является, теоретически, интересным, и будем надеяться, что решение этой проблемы послужит толчком для решения других проблем аналогичной природы». Шеннон отмечает теоретическое существование лучшего хода в шахматах и практическую невозможность найти его. Он описывает две стратегии поиска лучшего хода, обе основываются на эвристической функции оценки конечных точек:

- перебор всех возможных ходов на фиксированную глубину, с вызовом в конце оценочной функции (т. к. невозможно осуществить перебор до конца);
- выполняет только выборочное расширение определенных строк, используя накопленные шахматные знания, чтобы подрезать неинтересные ветви.

Теперь разберём сами алгоритмы. Для примера возьмём несколько алгоритмов: AlphaBeta, который основан на переборе всех возможных позиций, и Giraffe, обучением которого занималась нейросеть.

AlphaBeta. Как уже было упомянуто выше, AlphaBeta основан на полном переборе всех позиций, правда с механизмом осечения ветвей. Кажется на первый взгляд, что нельзя досрочно прекратить поиск в узле без ухудшения качества поиска. Но это не так. Дерево поиска,

формируемое рекурсивно, можно представить как своеобразную нейронную сеть. Лучшая оценка увеличивается, пока не достигнет некоторого предела. Какого? В этом весь вопрос. Представьте себе некоторый узел поиска в дереве перебора. Оценка в нем только увеличивается. Достигнутый максимум передается далее. Оппонент занижает оценку. Если он ее уже занизил до нашего максимума, то дальше искать нет смысла. Это позволяет существенно снизить поиск, отбрасывая ненужные ветви. Хоть алгоритм и кажется простым, тем не менее, он очень эффективен. До сих пор алгоритмы на его основе успешно учувствуют в международных соревнованиях.

Giraffe. Студент Имперского колледжа Лондона Мэтью Лай (Matthew Lai) разработал компьютерную программу Giraffe, которая трое суток играла в шахматы сама с собой — и смогла извлечь все предметно-ориентированные знания, необходимые для игры на уровне международного мастера ФИДЕ.

Для этого ей достаточно вычислительных ресурсов нормального компьютера. Обучение нейросети происходило 72 часа в 20 тредов на машине с двумя 10-ядерными процессорами Xeon E5-2660. В научной работе автор пишет, что после тренировки в течение 72 часов программа выбирала наилучший возможный ход в 46% случаев, а один из трёх наилучших ходов — в 70% случаев. Это очень неплохой результат даже для обычных шахматных программ. Но этот алгоритм достиг своего предела, и дальнейшее его обучение потеряло смысл (рисунок 1).

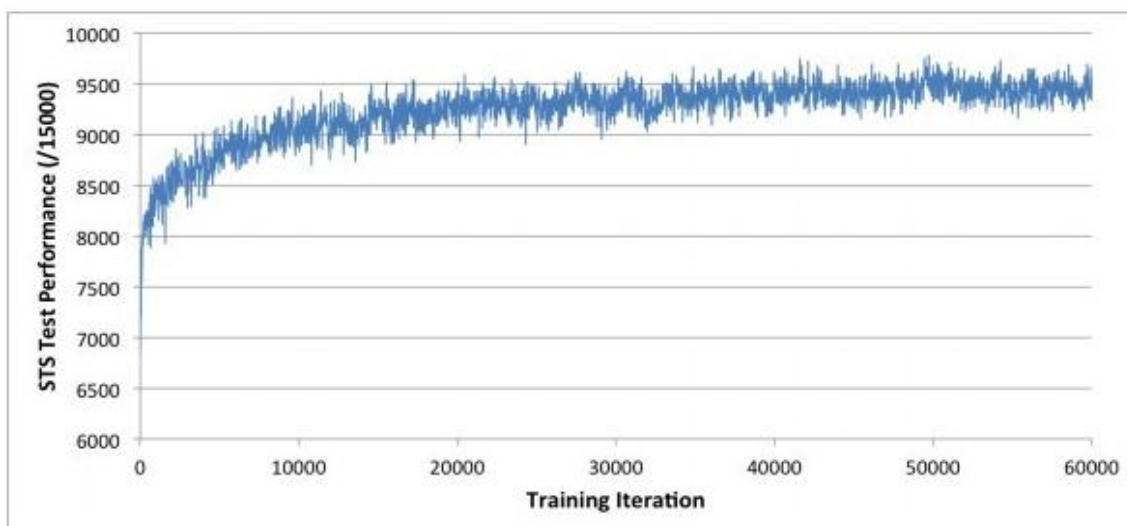


Рисунок 1 – Скорость обучения Giraffe

Муравьиный алгоритм. В основе алгоритма лежит поведение муравьиной колонии — маркировка более удачных путей большим количеством феромона.

Работа начинается с размещения муравьёв в на шахматной доске, затем начинается движение муравьёв — направление определяется вероятностным методом, на основании формулы (1):

$$P_i = \frac{l_i^q \cdot f_i^p}{\sum_{k=0}^N l_k^q \cdot f_k^p} \quad (1)$$

Решение не является точным и даже может быть одним из худших, однако, в силу вероятности решения, повторение алгоритма может выдавать (достаточно) точный результат. Эффективность муравьиных алгоритмов сравнима с эффективностью общих мета эвристических методов, а в ряде случаев — и с проблемно-ориентированными методами.

Наилучшие результаты муравьиные алгоритмы показывают для задач с большими размерностями областей поиска. Муравьиные алгоритмы хорошо подходят для применения вместе с процедурами локального поиска, позволяя быстро находить начальные точки для них.

Идея применения заключается в том, чтобы объединить AlphaBeta и Муравьиный алгоритм, тем самым добавив в него человечности, и возможность обучения.

Результатом выполнения данной работы является клиент-серверное приложение, с помощью которого можно наблюдать за эффективностью различных алгоритмов натравливая, их друг на друга, либо самим попробовать выиграть их в честном бою.

ЛИТЕРАТУРА

1. Евгений Корнилов «Программирование шахмат и других логических игр», 2005. – 272 с.

УДК 004.415

Студ. А.В. Бурмакова

Науч. рук. доц. А.И. Бракович

(кафедра информационных систем и технологий, БГТУ)

3D-МОДЕЛЬ ПРОГНОЗИРОВАНИЯ ПРОЦЕССА ЗАГРЯЗНЕНИЯ ГЕОЛОГИЧЕСКОЙ СРЕДЫ В РЕЗУЛЬТАТЕ ПРОЛИВА НЕФТЕПРОДУКТА

Ныне слово «экология» стало весьма популярным, этот термин нередко употребляют в сочетании с такими словами как общество, культура, семья, здоровье и так далее. Наиболее часто применяют это слово, указывая на неблагоприятное состояние окружающей нас природы.

Такая проблема как разливы нефтепродуктов носит трудно учитываемый характер, поскольку нефтяное загрязнение нарушает многие естественные процессы и взаимосвязи, существенно изменяет условия обитания всех видов живых организмов и накапливается в биомассе.

Нефтепродукты является продуктом длительного распада и очень быстро покрывает поверхность вод плотным слоем нефтяной пленки, которая препятствует доступу воздуха и света.

Благодаря современным информационным технологиям, таким как симуляцию объектов и их поведение основанных на реальных математических формулах, можно предугадать поведение нефтепродуктов и создать полное воплощение действий, их последствий.

Благодаря этому проекту наглядно будут видны многие проблемы современной экологии, не прибегая к бесконечным расчетам, а лишь установив некоторые параметры. Все расчеты делает программа и сиюсекундно проектирует все на модель.

Основная цель данного проекта заключается в создании реалистичной модели, которая будет отражать все последствия разлива нефтепродукта, где сам пользователь сможет управлять ключевыми параметрами.

Для достижения цели сформулированы следующие задачи:

1. Воссоздать реалистичную 3D модель лесной территории, почвы, грунтовых вод, объекты нефтепродукта.
2. Основываясь на математических формулах воссоздать поведение нефтепродукта, загрязнение почвы.
3. Симуляцию ситуации должен создавать пользователь,

поэтому модель должна быть гибкая, работать точно основываясь на параметрах, которые задал пользователь.

4. Модель должна быть реалистична, эстетически красива, с приятным интерфейсом.

5. Программа должна быть независимая, не требующая установки дополнительного ПО.

6. Модель должна быть проста и понятна для любого «рядового» пользователя, который не специализирует свои знания ни на программном обеспечении, ни в вопросах экологии.

Наиболее удобным инструментом для воссоздания данной задачи наиболее удобным решением является Unity3D версия 5.5.2f1.

Unity — это мощный инструмент для разработки двух- и трёхмерных приложений и игр, работающий под операционными системами Windows, Linux и MAC OS X.

У Unity есть два основных преимущества перед другими передовыми инструментами разработки игровых моделей: чрезвычайно производительный визуальный рабочий процесс и мощная межплатформенная поддержка.

Именно данные преимущества и сыграли решающую роль в выборе инструментария.

Помимо визуальной части большую роль играет написание корректного скрипта. Это необходимо для описания поведения модели. Осуществляется это привязкой в компоненту модели скрипта, в котором описывается логика поведения объекта в зависимости от заданных параметров и математических формул.

Скрипт для данной модели осуществлялся на языке программирования C#, который поддерживает платформа Unity3D.

На начальном этапе выбрали ключевые факторы, на основе которых и основывалась модель:

1. Радиус разлива нефтепродукта;
2. Толщина грунта;
3. Скорость передвижения нефтепродукта в грунтовых водах;
4. Интервал времени.

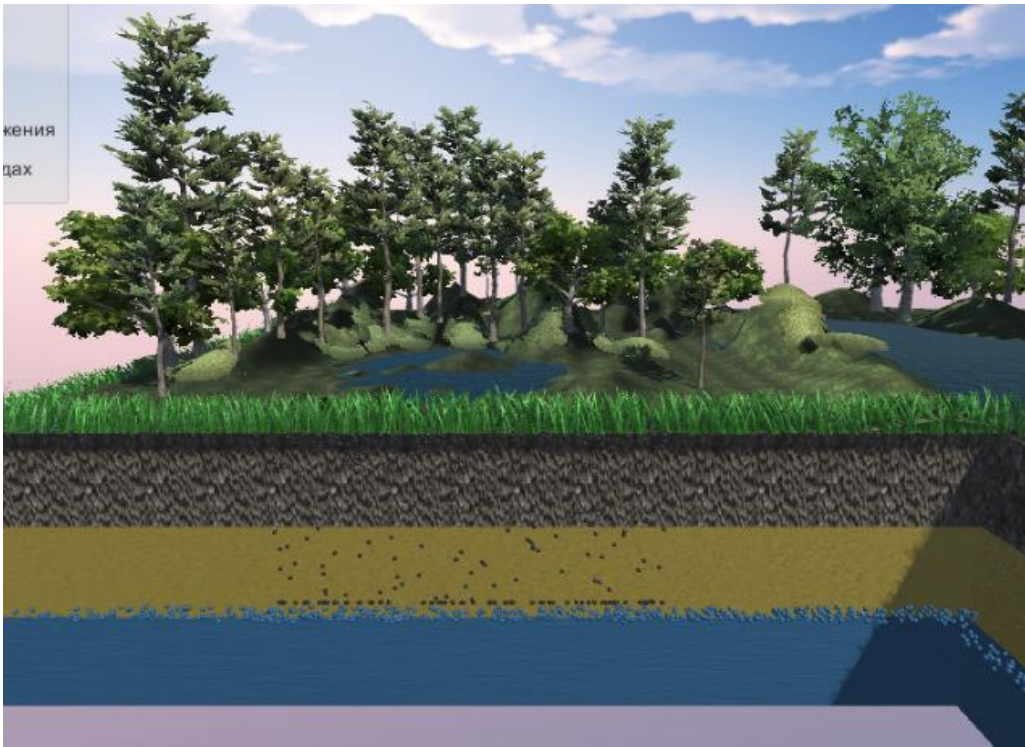


Рисунок 1 – Фронтальный вид модели

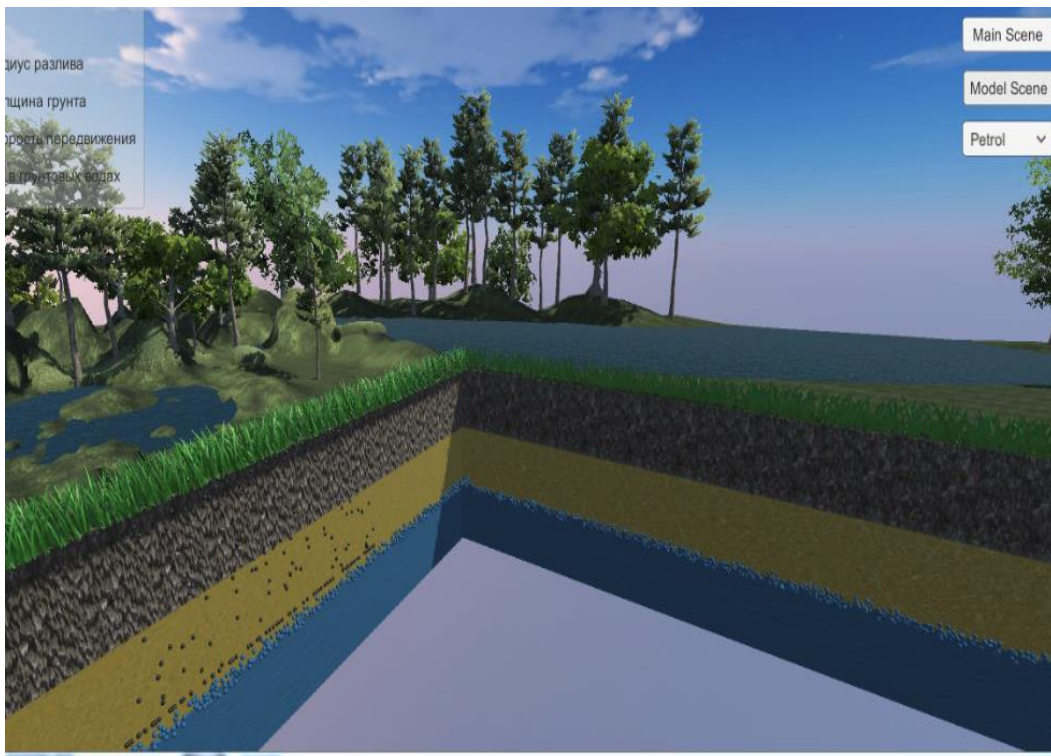


Рисунок 2 – Модель в разрезе, профильный вид

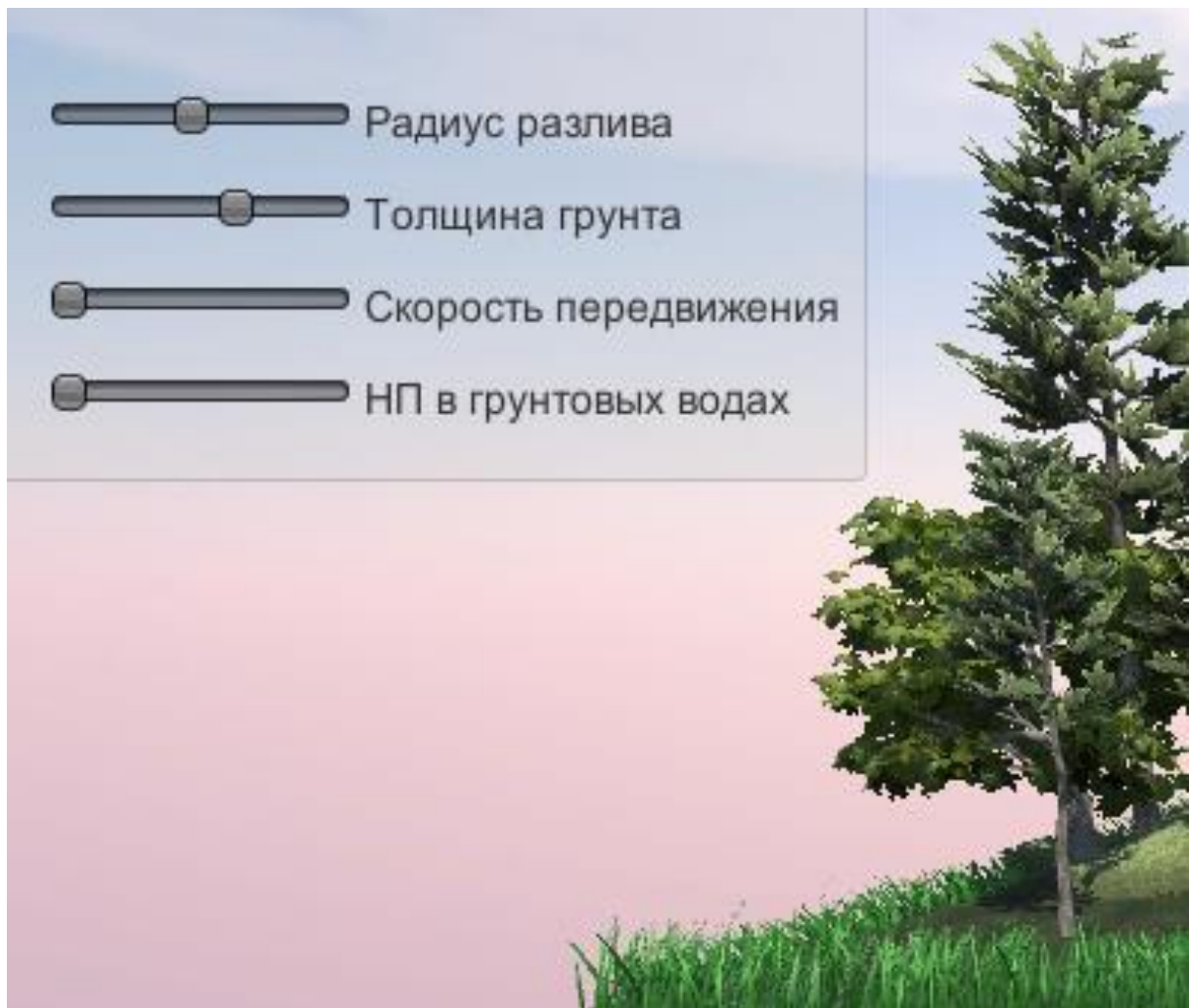


Рисунок 3 – Управление параметрами модели

Результатом выполнения данной работы является 3D модель, основанная на реальных математических формулах, помощью которой можно прогнозировать результат влияния пролива нефтепродукта на геологическую среду.

ЛИТЕРАТУРА

1. Википедия [Электронный ресурс]- https://ru.wikipedia.org/wiki/C_Sharp. Дата доступа 28.03.2017.
2. Википедия [Электронный ресурс] - <https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%82%D0%B5%D0%BD%D1%82>. Дата доступа 28.03.2017.

УДК 401.21

Студ. Врублевский Е.С.

Науч. рук. доцент Буснюк Н.Н.

(кафедра информационных систем и технологий)

РЕАЛИЗАЦИЯ ЗАДАЧИ ОБ ОПТИМАЛЬНОМ ПУТИ В СЕТЕВОМ ГРАФИКЕ В СЛУЧАЕ НЕФИКСИРОВАННЫХ ВЕСОВ ДУГ НА ЯЗЫКЕ C++.

Постановка задачи

В классической задаче сетевого планирования задана последовательность выполнения работ некоторого проекта и их продолжительность^[3].

Рассмотрим некоторый проект – совокупность N операций (работ), составляющий некоторый многошаговый процесс. На выполнение данного проекта выделено M работников. Каждый работник индивидуален и имеет свою производительность (время работы)^[1]. При этом производительность на каждой отдельной работе каждого работника может отличаться. Считаем известными все работы, которые предстоит совершить, и их последовательность, количество работников, а так же время выполнения каждой отдельной работы каждым работником. Проект может быть изображен в виде графа-сети. Зададимся целью определить кратчайший срок завершения проекта.

Решение

По введенным данным строится граф работы. Дугами графа будем изображать работы, а вершинами графа – некоторые события. Назовем элементарными событиями начало и конец каждой работы, а некоторую совокупность элементарных событий – событием.

Важно, единственным началом графа будет служить точка 0, которая является событием начала проекта. Учтем, что работа может иметь одно из следующих состояний:

1. Работа не доступна.
2. Работа доступна.
3. Работа ожидает (т.е. будет выполнена пока еще занятым работником).
4. Работа выполняется.
5. Работа выполнена.

Так же учтем то, что у каждой работы есть свои условия доступности. Условием доступности работы является завершение всех

работ, которые должны быть выполнены до данной работы и показывает то, что работа может быть выполнена в данный момент.

Работник может иметь одно из следующих состояний:

1. Работник свободен.
2. Работник занят.
3. Работник освобождается (т.е. работник на следующем шаге станет свободным).

Для просмотра точек используется метод «поиск в ширину» [2]. Вначале выполняется просчет точек 1 порядка. Все точки 1 порядка будут в начальный момент времени иметь состояние "Работа доступна", т.к. перед ними не нужно выполнять никакую другую работу.

После получение списка доступных работ, дальнейшие действия происходят по следующему алгоритму:

1. Если есть работа с состоянием "работа доступна" и есть работник с состоянием "работник свободен", выполняем следующее, иначе пункт. 5.

2. Методом перебора работников и их времени доступной работы находим минимальное из возможных времён.

3. Если этот работник на данном шаге свободен, то переводим работника в состояние "работник занят", а работу в состояние "работа выполняется". Иначе переводим работу в состояние "работа ожидает".

4. Переходим к пункту 1.

5. Если все работы выполнены, то переходим к пункту 10.

6. Меняем состояние всех работ с состоянием "работа ожидает" на "работа доступна".

7. Если какой-либо работник имеет состояние "работник освобождается", то состояние работы, которую он выполнял, меняется на "работа выполнена", а состояние работника на "работник свободен". Иначе пункт 9.

8. Проверяется условие доступности для всех недоступных работ, если для работы выполнены все условия, то она переводится в состояние "работа доступна".

9. Увеличиваем на 1 число выполненных шагов и переходим к пункту 1.

10. Выводим результат.

Пример

Рассмотрим решение задачи на простом примере. Проект состоит из 3 работ. Вся информация о последовательности работ представлена в таблице 1.

Таблица 1 - Информация о работе и производительности работы.

Работа	До неё	Шагов первому	Шагов второму
1	Нет	4	6
2	Нет	5	5
3	1 и 2	6	4

В проекте учувствуют 2 работника. Вся информация о производительности представлена в таблице 1.

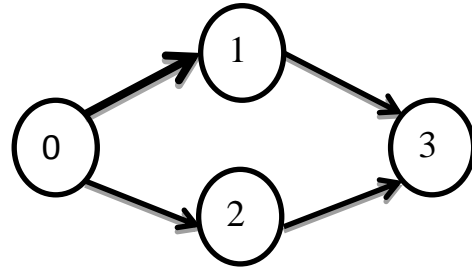


Рисунок 1 - Граф работы

В начале, на 0 шаге, имеется 2 доступные работы: 1 и 2, которые берутся на выполнение работникам 1 и 2 соответственно, после поиска рационального решение распределения времени. 1 работник освободиться через 4 шага, а 2 через 5. Работа 3 недоступна и имеет следующие условия доступности: выполнение 1 и 2 работы. Справедливо то, что пока все работники заняты, никаких новых событий достигнуто не будет.

На 4 шаге, когда освобождается 1 работник, снова проверяются условия доступности. 1 работа уже выполнена, 2 работа выполняется, а для 3 работы еще не выполнено условие доступности, нужно ожидать завершения 2 работы. Т.к. свободных работ нет, то 1 работник остаётся свободным и ожидает какого-то дальнейшего события.

На 5 шаге 2 работа завершается и второй работник освобождается. Т.к. работы 1 и 2 завершены, то 3 работа становится доступной для выполнения. Происходит поиск рационального решения. 2 работник берется за выполнение 3 работы, которая займет у него 4 шага.

На 9 шаге, после завершения 3 работы, все работы будут завершены, а минимальное, рациональное решение данного проекта будет найдено и выведено на экран.

```

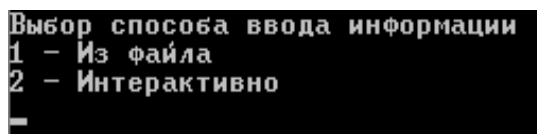
Шаг 0 => [Выполняется: работник 1 работа 1 за 4 шаг<ов>
Шаг 0 => [Выполняется: работник 2 работа 2 за 5 шаг<ов>
Шаг 5 => [Выполняется: работник 2 работа 3 за 4 шаг<ов>
Результат: 9 шагов
  
```

Рисунок 2 - Выполнение программы в среде Visual Studio

Реализация

Данный алгоритм был реализован в среде Visual Studio 2013 на языке программирование C++. Программа имеет следующий интерфейс при вызове. Существует два способа ввода начальной информации:

1. Ввод информации из файла
2. Интерактивный ввод



```
Выбор способа ввода информации
1 - Из файла
2 - Интерактивно
_
```

Рисунок 3. Интерфейс программы

После ввода данных, программа использует алгоритм, который описан выше и выводит результат.

ЛИТЕРАТУРА

1. Буснюк Н. Н., Новиков В.А. Метод оптимального решения задачи о назначениях в сетевом планировании // Труды БГТУ. 2016,
2. Буснюк Н. Н., Черняк А.А. Математическое моделирование. Минск: Беларусь, 2014. – 216 с.
3. Смелов В. В., Брусенцова Т.П. Основы сетевого планирования. Минск: БГТУ, 2010. – 212 с.

УДК 628.39

Студ. Ровба А.В.

Науч. рук. доц. Г. И. Касперов
(кафедра инженерной графики, БГТУ)

ПРЕДУПРЕЖДЕНИЕ ЧРЕЗВЫЧАЙНЫХ СИТУАЦИЙ НА ВОДОЕМАХ ТЕХНИЧЕСКОГО НАЗНАЧЕНИЯ

В Республике Беларусь эксплуатируется большое количество объектов энергетики, использующих в технологических целях (выработка пара, охлаждение и т.д.) воду из естественных и искусственных водных источников, к которым относятся водоемы-охладители. В ряде случаев водоемы-охладители создаются на базе существующих естественных озер (Лукомльская ГРЭС – Лукомльское озеро) или же создаются водохранилища - Чижовское водохранилище для ТЭЦ-3 и ТЭЦ-3 в г.Минске, водохранилище Дрозды – для ТЭЦ-4 и ряд других. Водоохранилище «Коммунарка» является источником подпитки ТЭЦ-2, Лукомльское озеро (Лукомльское водохранилище) – для Лукомльской ГРЭС. Аналогичные объекты расположены и в областных центрах страны. Объекты исследований (водохранилища для технологических нужд и пруды-охладители гидроэлектростанций и теплоэлектростанции)

расположены по всей территории Беларуси. Эксплуатация водоемов-охладителей происходит в особых термических, урвенных и ветроволновых режимах, которые сказываются на условиях функционирования и состоянии гидротехнических сооружений. Разрушение гидротехнических сооружений может создать чрезвычайные ситуации, развивающиеся по различным сценариям.

Практически все населенные пункты страны имеют очистные сооружения, в состав которых входят большие по площади пруды биологической очистки и пруды-отстойники. Данный тип водоемов имеет особый режим эксплуатации, характеризуемый работой в экстремальных условиях при повышенной загрязненности и фильтрации откосов. Линейные размеры прудов достаточно большие, что вызывает развитие значительного ветрового волнения. Все это вместе с колебанием уровней, атмосферными воздействиями ведет к разрушению незакрепленных откосов ограждающих дамб обвалования. Из-за недостаточного объема финансирования ряд объектов и сооружений эксплуатируются с нарушениями и требуют ремонтных и восстановительных работ. Неудовлетворительное состояние объектов очистки сточных вод приводит к авариям, разрушению дамб и загрязнению водоемов и территорий (например, авария на очистных сооружениях – прудах биологической очистки Круглянского жилищно-коммунального хозяйства, авария на минских очистных сооружениях и др.).

Таким образом, проблема эксплуатации и состояния водных объектов, используемых в энергетике, техническом водоснабжении, жилищно-коммунальном хозяйстве, является актуальной для страны и требует исследования для предупреждения и оценки последствий чрезвычайных ситуаций. Проведенные научные исследования позволили установить, что:

- наибольшее количество аварий происходит на водоемах технического назначения, срок эксплуатации которых превышает 35-40 лет и более;
- основными причинами аварий на очистных сооружениях являются: переполнение хранилища (31%), атмосферные осадки (27%), ошибки при проектировании (23%), другие факторы (19%);
- на вероятность возникновения чрезвычайных ситуаций на водоемах технического назначения, важную роль оказывает человеческий фактор – качество изысканий, проектирования, строительства и эксплуатации данного типа объекта повышенной опасности.

УДК 535.37+541.65+543.4

Студ. Е.С. Люлькович

Науч. рук. зав. кафедрой физики, д. физ.-мат. наук Н.Н. Крук

(кафедра физики, БГТУ)

СПЕКТРАЛЬНЫЕ ПРОЯВЛЕНИЯ АТРОПОИЗОМЕРИИ ПОРФИРИНОВ В РАСТВОРАХ

В работе исследованы спектральные характеристики водорастворимого 5,10,15,20-тетра-(3-*N*-метилпиридил)-порфирина [1]. Структурная формула данного соединения показана на рисунке 1. Из-за вращения пиридильных заместителей вокруг связи $C_{\text{мезо}}-C_1$ метилированный атом азота пиридила у *мета*-замещенного производного может располагаться либо с одной, либо с другой стороны плоскости тетрапиррольного макроцикла. Ранее нами было показано, что это приводит к формированию конформеров: в растворе могут одновременно находиться 4 дважды вырожденных атропоизомера, с соотношением статистических весов 1:4:2:1. [2,3]

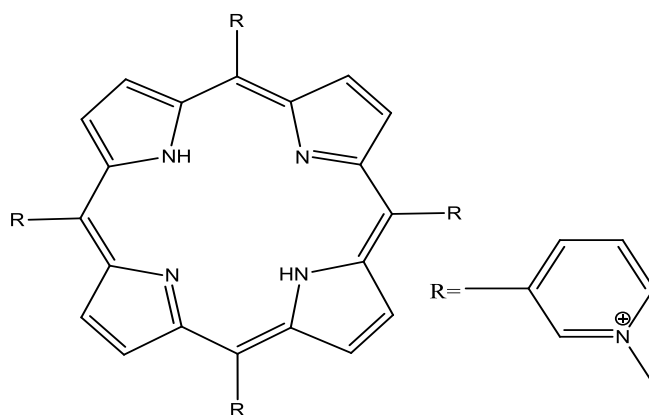


Рисунок 1 – Молекулярная структура 5,10,15,20-тетра-(3-*N*-метилпиридил)-порфирина

Анализ измеренных электронных спектров поглощения водных растворов исследуемого соединения указывает на наличие временной эволюции спектра поглощения (рисунок 2). Так, поглощение наиболее интенсивной полосы IV в видимой области спектра с максимумом при 514,5 нм убывает, в то время как поглощение соседней полосы III растет. Это свидетельствует о частичном превращении атропоизомера(ов), наблюдаемого(ых) в свежеприготовленном растворе, в другие атропоизомеры. Изменение оптической плотности на длине волны 514,5 нм в зависимости от времени хранения раствора, моделируется двухэкспоненциальной зависимостью с

характеристическими временами 0,63 часа и 79 часов (рисунок 3). Раствор приходит в равновесное состояние примерно через 350 часов и при дальнейшем хранении спектральные изменения не происходят.

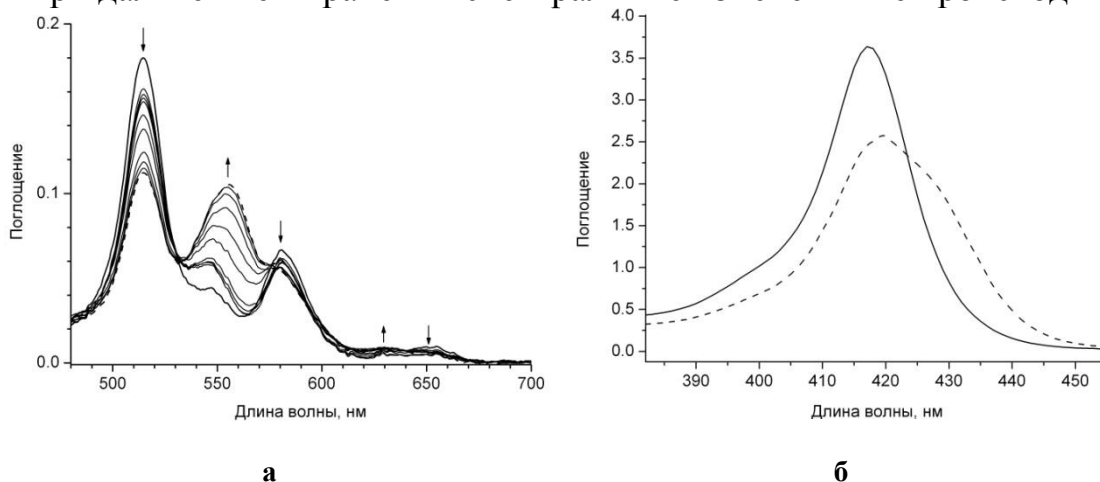


Рисунок 2 – Зависимость спектров поглощения 5,10,15,20-тетра-(3-N-метил-пиридил)-порфирина от времени хранения раствора: а) видимая область; б) полоса Сорэ. Равновесный спектр (через 1550 часов) показан пунктирной линией. Стрелки указывают направление спектральных изменений

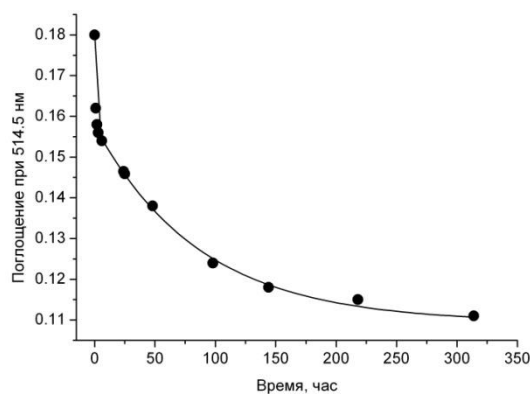


Рисунок 3 – Кинетика изменения поглощения 5,10,15,20-тетра-(3-N-метил-пиридил)-порфирина на длине волны 514,5 нм

Измерение спектров возбуждения флуоресценции 5,10,15,20-тетра-(3-N-метил-пиридил)-порфирина при различных длинах волн регистрации флуоресценции также обнаруживает гетерогенность раствора, обусловленную наличием нескольких излучающих центров. Нами были идентифицированы три атропоизомера. Спектр

возбуждения флуоресценции, измеренный при регистрации флуоресценции на длине волны 708 нм, совпадает со спектром поглощения исходного атропоизомера. При регистрации на длине волны 670 нм получены спектральные характеристики другого атропоизомера, характерной чертой которого является высокая относительная интенсивность длинноволнового электронного перехода. Третий атропоизомер, концентрация которого существенно меньше, идентифицирован при регистрации спектра возбуждения на длине волны 613 нм.

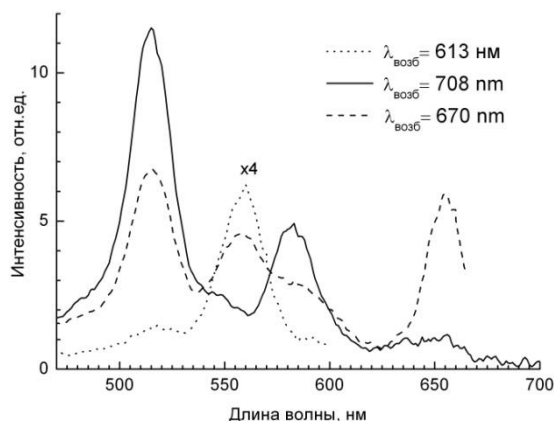


Рисунок 4 – Спектры возбуждения флуоресценции 5,10,15,20-тетра-(3-N-метил-пиридил)-порфирина при различных длинах волн регистрации флуоресценции

Таким образом, в результате наличия нескольких атропоизомеров 5,10,15,20-тетра-(3-N-метилпиридил)-порфирина спектры поглощения обнаруживают временную эволюцию, отражающую процесс установления равновесия между атропоизомерами в растворе. Спектральные характеристики трех атропоизомеров в равновесном растворе идентифицированы с помощью регистрации спектров возбуждения флуоресценции.

ЛИТЕРАТУРА

1. Hambright, P., Gore, T., Burton, M. Synthesis and characterization of new isomeric water-soluble porphyrins. Tetra(2-N-methylpyridyl)porphine and Tetra(3-N-methylpyridyl)porphine // *Inorganic Chemistry*. – 1976. – V. 15, № 9. - P. 2314-2315.
2. Люлькович, Е.С., Крук, Н.Н. Флуоресценция 5,10,15,20-тетраметилпиридил-порфиринов в растворах // 67-я научная

конференция учащихся, студентов и магистрантов, 18-23 апреля 2016 г., Минск. – Сб. научн. работ в 4-х частях. – Мн.: БГТУ, 2016. – Ч.4. – С.161-163.

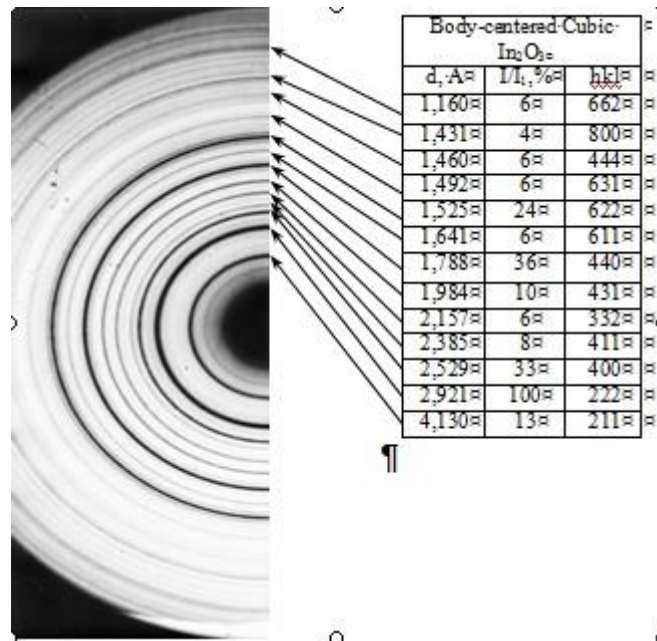
3. Крук, Н.Н., Люлькович, Е.С. Спектральные свойства конформеров тетра-(3-*N*-метилпиридил)-порфирина в растворах // Международная научная конференция «Молекулярные, мембранные и клеточные основы функционирования биосистем», XII-й съезд Белорусского общественного объединения фотобиологов и биофизиков, 28-30 июня 2016 г., Минск. – Сб. научн. работ в 2-х частях. – Мн.: Изд. центр БГУ, 2016. – Ч.2. – С. 54-57.

УДК 537.311: 621.3.084.2

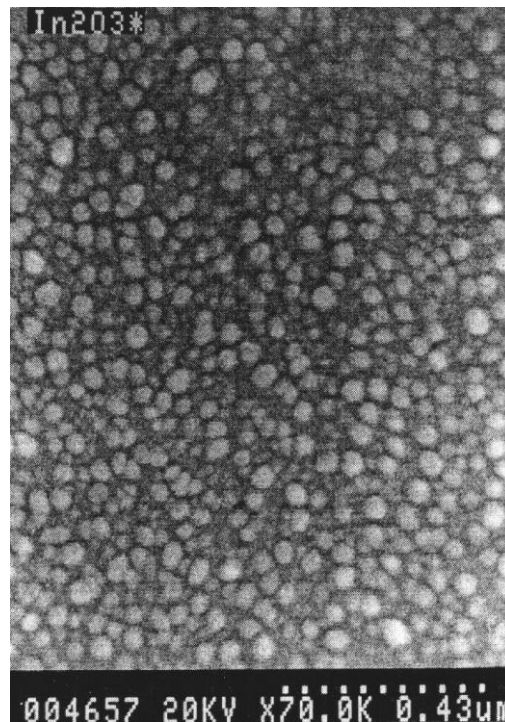
Студ. С.С. Шиканов
Науч. Рук. А. Е. Почтенный
(кафедра физики, БГТУ)

ПРОВОДИМОСТЬ ПЛЕНОК ОКСИДА ИНДИЯ В ПРИСУТСТВИИ АДСОРБИРОВАННОГО КИСЛОРОДА

Тонкие пленки оксида индия были получены с помощью термического окисления тонких пленок индия, осажденных на подложки из слюды (мусковит) методом магнетронного распыления на постоянном токе в атмосфере аргона. В качестве катода использовался металлический индий с чистотой ЧДА. После осаждения пленки индия окисляли в электрической муфельной печи в неизотермических режиме, нагревая до температуры 500–600°C в течение 40–60 минут и отжигали в изотермическом режиме при температуре 500°C и 600°C в течение 60 минут (температура и время окисления были подобраны экспериментально). Технология получения оксидных пленок методом термического окисления металлического слоя позволяет формировать оксидные слои толщиной от нескольких нанометров до нескольких сотен нанометров. Рентгеновская дифрактограмма показывает, что образуется оксид индия In_2O_3 , что представлено на рисунке 1.



**Рисунок 6- рентгеновская дифрактограмма
Электронно-микроскопическое изображение поверхности пленок
оксида индия на слюде и распределение наночастиц оксида индия в этой
пленке по размерам представлено на рисунках 2 и 3.**



**Рисунок 7 - Электронно-микроскопическое изображение наночастиц
In₂O₃**

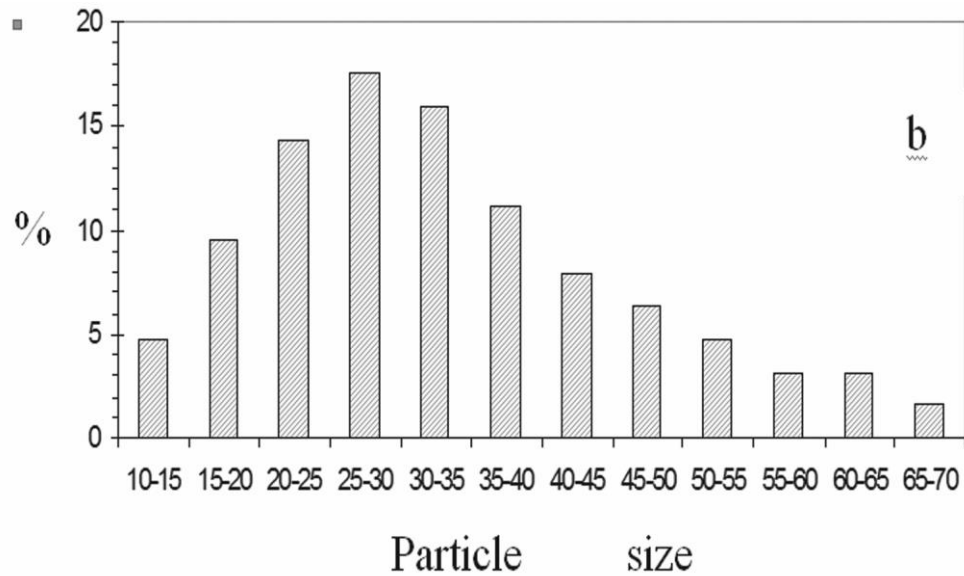


Рисунок 8 - распределение наночастиц оксида индия по размерам

В ходе эксперимента было обнаружено увеличение проводимости пленки при понижении концентрации адсорбированного кислорода в результате термодесорбции и откачки последнего форвакуумным насосом. Данные представлены на рисунке 4.

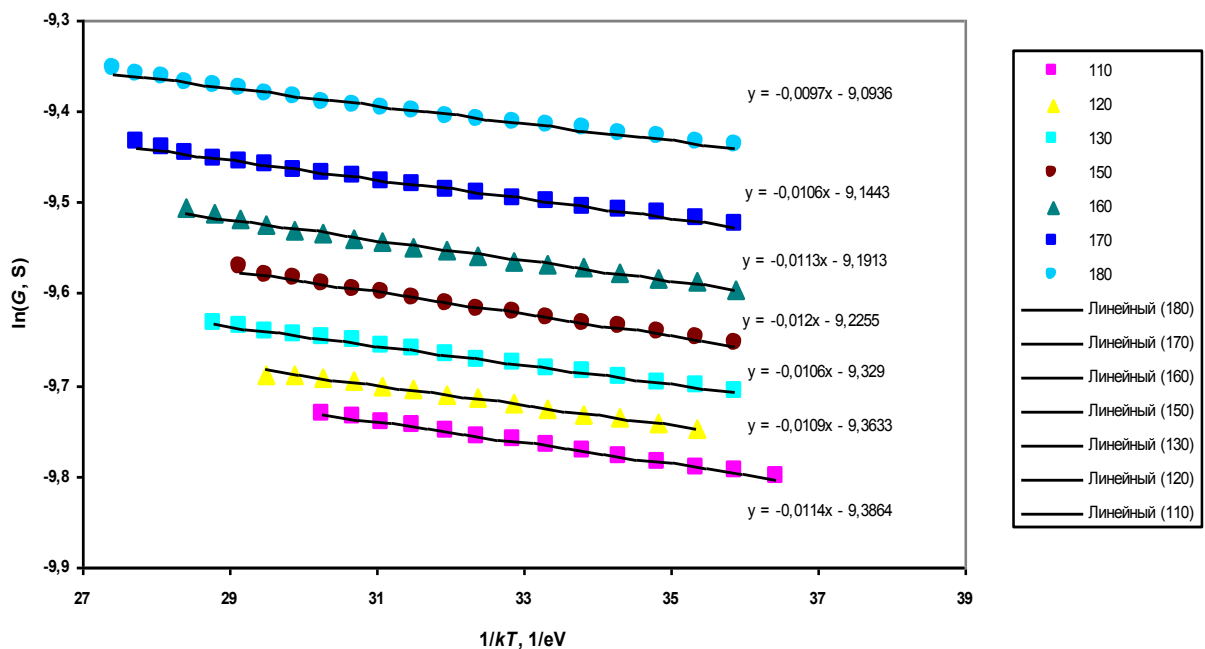


Рисунок 9 - температурная зависимость проводимости пленок оксида индия при разных температурах начала охлаждения

На рисунке 5 показан рентгеновский фотоэлектронный спектр кислорода в оксиде индия. Значение энергии 531,9 eV соответствует химически связанному кислороду, а 529,95 eV - адсорбированному.

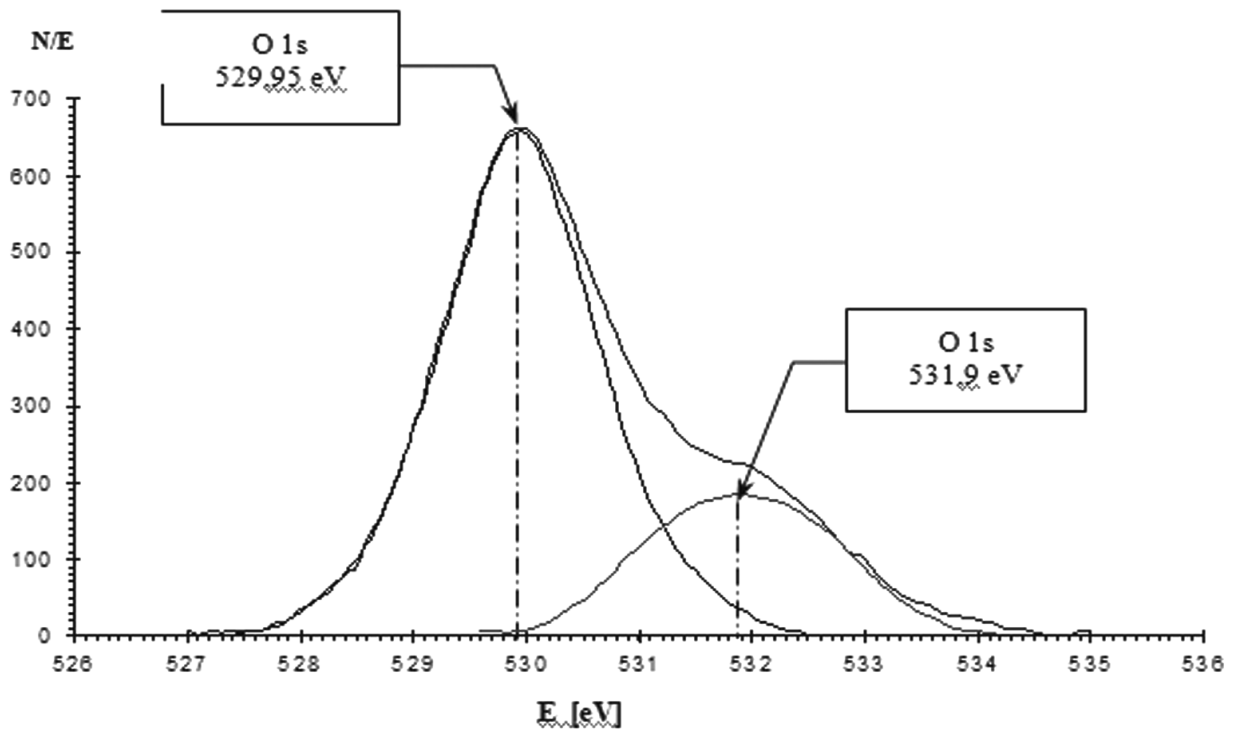


Рисунок 10 -- рентгеновский фотоэлектронный спектр кислород в оксиде индия

Малая энергия активации проводимости в сравнении со значениями при типичной примесной проводимости свидетельствует о том, что адсорбированный кислород является центром рассеяния носителей заряда и с уменьшением его концентрации проводимость пленки растет. Адсорбированный кислород в пленках In_2O_3 представляет собой центры рассеяния носителей заряда, уменьшение его концентрации приводит к увеличению подвижности носителей заряда и тем самым к увеличению проводимости пленок

Обнаруженный эффект может иметь практическое применение для создания адсорбционно-резистивных сенсоров кислорода.

ЛИТЕРАТУРА

1. Почтенный, А. Е. Прыжковая проводимость на постоянном токе в собственных и примесных органических полупроводниках: монография / А. Е. Почтенный. – Минск: БГТУ, 2016. – 171 с. – ISBN 978-985-530-509-6.

УДК 674.055:621.934(043.3)

Студ. В.А. Корнева

Науч. рук. доц. В.В. Чаевский

(кафедра физики, БГТУ)

АНАЛИЗ СЦЕПЛЕНИЯ ZrC-Ni-УДА ПОКРЫТИЙ С ТВЕРДОСПЛАВНОЙ ОСНОВОЙ

В настоящее время использование ультрадисперсных алмазов (УДА), получаемых детонацией взрывчатых веществ, в качестве композиционного материала в электрохимических покрытиях приводит к повышению их износостойкости, существенной адгезии, снижению коэффициента трения [1]. Положительными сторонами электрохимического процесса обработки инструмента являются простота технологических операций формирования гальванического слоя, отсутствие высоких температур при изготовлении инструмента, низкая энергоёмкость установок. Основной недостаток модифицированного инструмента гальваническим сплавом на основе никеля состоит в том, что Ni-матрица не обладает высокими прочностными характеристиками [2].

Поэтому целью данной работы было исследование поверхностных свойств (шероховатости, адгезии) ZrC-Ni-УДА-покрытий, сформированных на поверхности лезвий твердосплавных (из карбида вольфрама WC) ножей фрезерного инструмента комбинированным методом гальванической обработки и конденсацией вещества из плазменной фазы в вакууме с ионной бомбардировкой (КИБ), позволяющей существенно улучшить эксплуатационные свойства изделий, применяемых в различных отраслях промышленности, в т. ч. в станкостроении и деревообработке.

Ni-УДА композиционные электрохимические покрытия (КЭП) наносили на подготовленную поверхность лезвий твердосплавных ножей фирмы Leitz (Германия) на экспериментальной установке при плотностях тока 1–24 А/дм² в гальваностатическом и импульсном режимах электролиза из сульфаминовокислых электролитов никелирования. В качестве дисперсной фазы использовали УДА (ТУ РБ 28619110.001-95), являющиеся продуктом детонационного превращения взрывчатых веществ с размерами 3–5 нм, развитой удельной поверхностью 200–450 м²/г.

Концентрация УДА в электролите составляла 5 г/дм³. Процесс осаждения КЭП проводили при температуре 40–50°С и постоянном перемешивании электролита-суспензии для поддержания частиц УДА во взвешенном состоянии. Толщина покрытий не превышала 10 мкм.

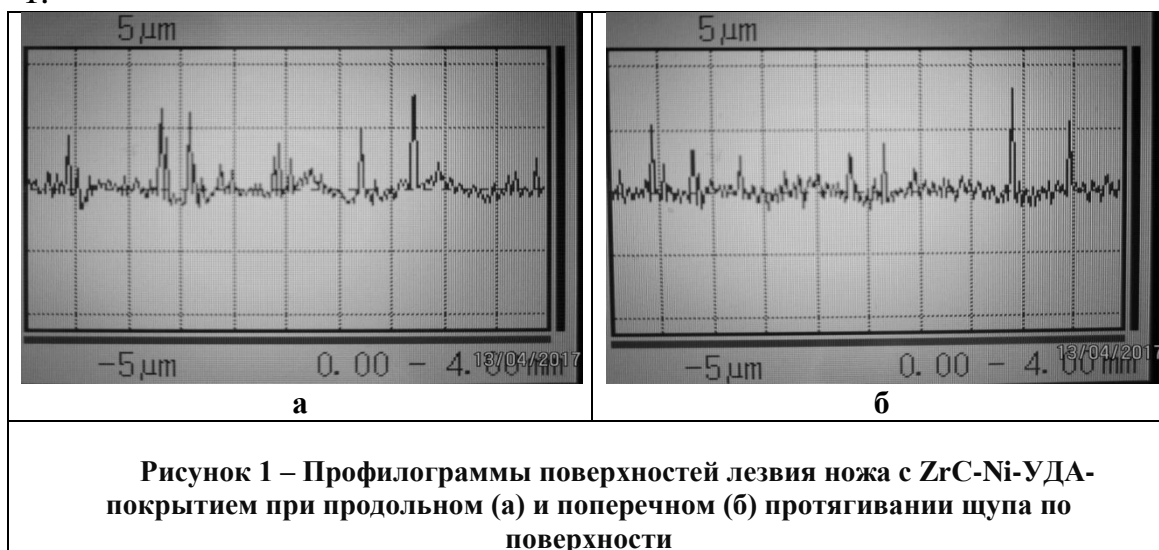
Ионно-плазменные ZrC-покрытия осаждались методом КИБ на поверхность ножей с Ni-УДА-покрытием на установке ВУ-1Б «Булат» по стандартной методике: с предварительной обработкой ионами циркония подложки в вакууме 10^{-3} Па при потенциале подложки, равном

–1 кВ, и последующим нанесением покрытий при токе горения дуги катода –100 А и опорном напряжении, равном –100 В, в атмосфере углеводорода CH_4 при давлении 10^{-1} Па. Температура при осаждении соответствовала 400–450°C. Толщина ZrC-покрытий не превышала 1,5 мкм.

Лабораторные испытания по определению параметра шероховатости Ra передней поверхности лезвия ножа с ZrC-Ni-УДА-покрытием были проведены вдоль, поперек и по диагонали поверхности согласно ГОСТ 2789-73 с помощью профилографа-профилометра Mitutoyo Surftest SJ-210 (Япония).

Измерение адгезионной прочности образцов с полученным покрытием проводилось на установке «скретч-тестер», разработанной и изготовленной БГУ – ЗАО «БМЦ». Принцип работы «скретч-тестера» основан на методе склерометрии – горизонтальном перемещении и царапании поверхности покрытия алмазным индентором (с радиусом закругления 0,5 мм), предварительно внедренным на определенную глубину при пропорционально возрастающей нагрузке и определении критической нагрузки, при которой происходит отрыв или разрушение покрытия [3]. Адгезионная прочность оценивалась по величине критической нагрузки.

Профилограммы поверхностей лезвия ножа показаны на рисунке 1.



Значение параметра шероховатости R_a были практически одинаковы по всем направлениям протягивания щупа по поверхности покрытия (рис. 2). Полученное значение среднего параметра шероховатости R_a составило 0,2724. Следовательно, морфология поверхности покрытия имеет равномерную и однородную структуру.

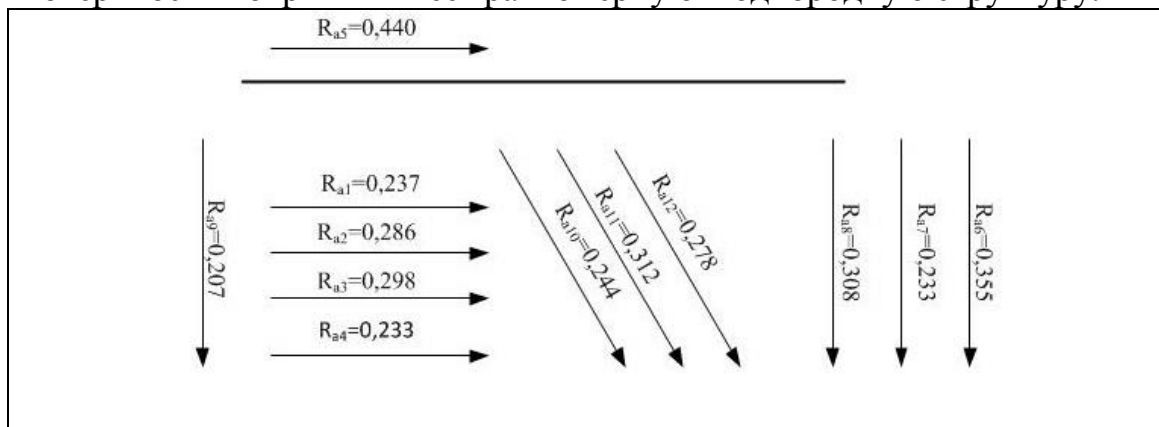


Рисунок 2 – Значения параметра шероховатости по различным направлениям на поверхности покрытия

Согласно полученным экспериментальным данным трибологических испытаний $ZrC-Ni$ -УДА покрытий на установке «скретч-тестер» при достижении нагрузки в 24 Н происходит разрушение покрытия, что соответствует изгибу и последующему одинаковому наклону кривой микроскрэтча (рис. 3).

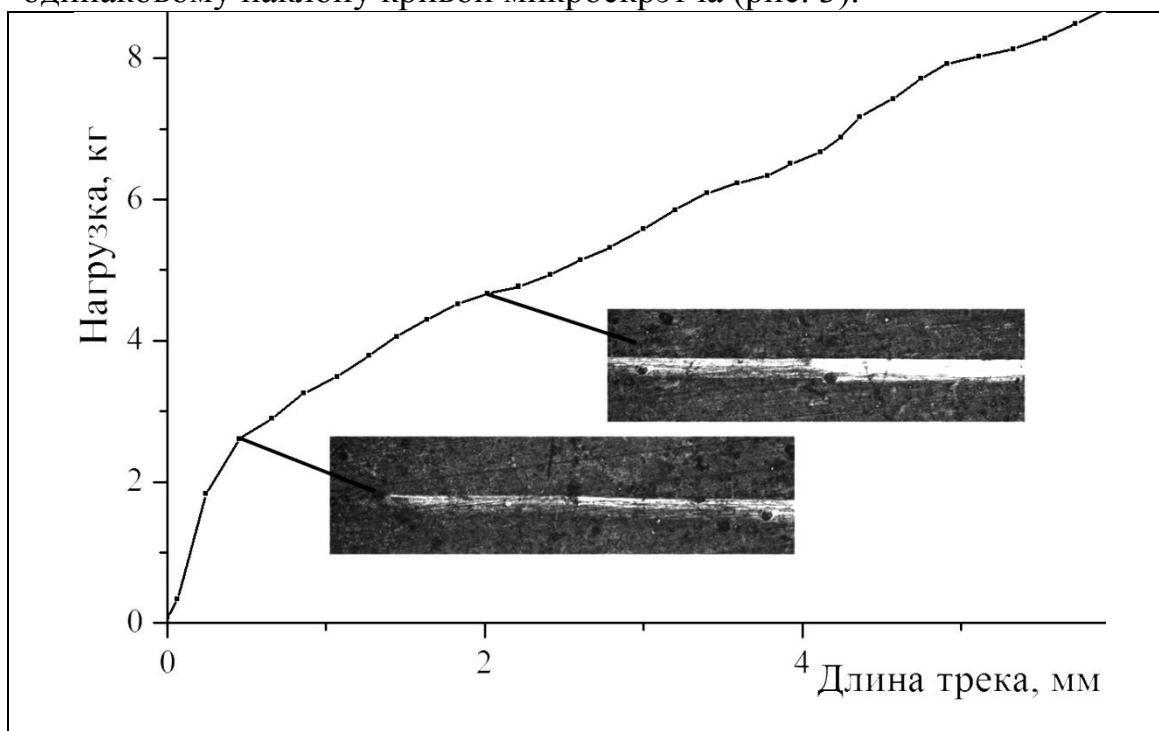


Рисунок 3 – Кривая микроскрэтча и снимки скрэтч-трека образца

В результате индентер при дальнейшем увеличении нагрузки начинает скользить по поверхности твердосплавной основы WC (рис. 4). Полученное значение критической нагрузки в 24 Н согласуется с литературными данными [4].

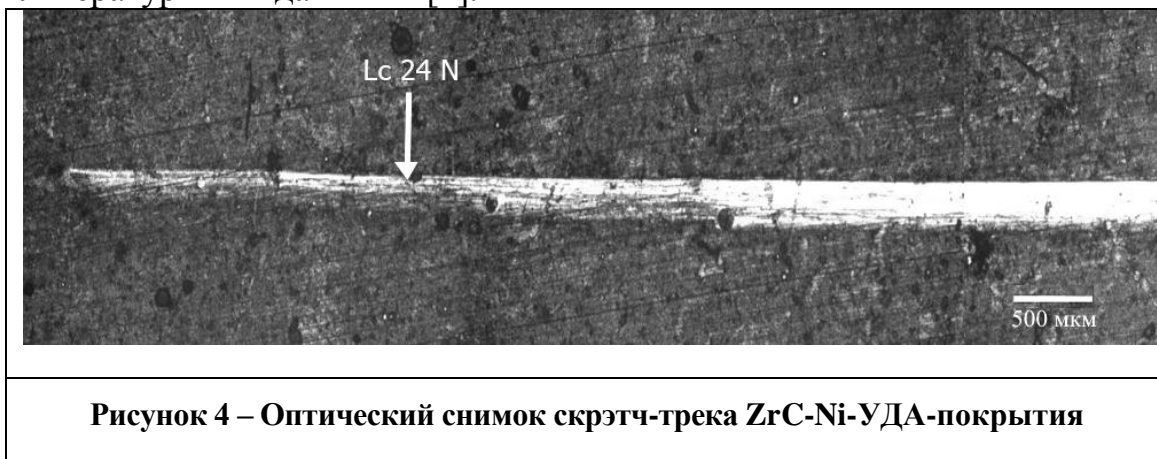


Рисунок 4 – Оптический снимок скрэтч-трека ZrC-Ni-УДА-покрытия

Таким образом, полученное среднее значение параметра шероховатости $R_a = 0,2724$ по всей поверхности ZrC-Ni-УДА-покрытия на лезвиях твердосплавных ножей фрезерного инструмента свидетельствует об однородности и равномерности сформированного покрытия. Критическая нагрузка на скрэтч-треке покрытия ZrC-Ni-УДА составила 24 Н, на основании чего можно утверждать об удовлетворительном сцеплении покрытия с твердосплавной основой.

ЛИТЕРАТУРА

1. Долматов В.Ю. Ультрадисперсные алмазы детанационного синтеза: свойства и применение // Успехи химии, 2001. – Т. 70, № 7. – С. 687-708.
2. Дисперсное упрочнение наночастицами алмазного композиционного электрохимического покрытия / Н.И. Полушин [и др.] // Известия высших учебных заведений. Порошковая металлургия и функциональные покрытия, 2011. – № 4. – С. 49-53.
3. Определение механических свойств и адгезионной прочности ионно-плазменных покрытий склерометрическим методом / В.М. Матюнин [и др.] // МИТОМ, 2002, № 3, С. 36–39.
4. The effects of the H/E ratio of various Cr-N interlayers on the adhesion strength of CrZrN coatings on tungsten carbide substrates / Hye-Kun Kim [et al.] // Surface & Coatings Technology. 2015. no. 284. P. 230–234.

УДК 546.212

Студ. А. О. Улитёнок,
 Науч. рук. доц. Л. Д. Яроцкая
 (кафедра высшей математики, БГТУ)

ИССЛЕДОВАНИЕ СОДЕРЖАНИЯ ДЕЙТЕРИЯ В ВОДЕ В ПРОЦЕССЕ ВЫПАРИВАНИЯ

Опасна ли вода из чайника? Среди населения бытует миф о том, что при кипячении природной воды в ней повышается содержание дейтерия – изотопов тяжёлого водорода, и это якобы может вредно сказаться на здоровье. «В процессе длительного кипячения из воды улетучиваются большая часть массы воды и таким путём увеличивается доля... «тяжёлой» воды D₂O ... Тяжёлая воды, естественно, осаждается внизу любого сосуда – чайника, титана. Поэтому если не выливать остатки кипячёной воды, а доливать к ней свежей, то при повторном кипячении процент тяжёлой воды в данном сосуде ещё больше увеличиться. При многократных добавлениях в остатки старой кипячёной воды новых количеств свежей воды может быть получена довольно большая концентрация тяжёлой воды. А это опасно для здоровья человека» [1]. Действительно ли это так?

Составим **математическую модель задачи**. Пусть общее количество воды в чайнике v молей. Обозначим через x на сколько увеличилась мольная доля тяжёлой воды после одного выпаривания. Проведем n операций выпаривания воды наполовину без и с доливанием воды до первоначального объема. Пусть x_n мольная доля тяжёлой воды после n таких операций. Известно, что мольная доля тяжёлой воды в питьевой воде $x_0 = 0,00016$, а коэффициент разделения, который показывает отношение концентраций тяжёлой воды в жидкой фазе и в паре, равен $\alpha = 1,03$.

Выпарим $v/2$ молей воды, тогда мольная доля тяжёлой воды в газовой фазе будет равна x/α , а общее количество тяжёлой воды в чайнике и в паре не измениться. Получаем уравнение:

$$\frac{v}{2}x + \frac{v}{2}\frac{x}{\alpha} = vx_n,$$

откуда следует, что $x = \frac{2x_n}{1+1/\alpha}$.

Таким образом, после каждого выпаривания наполовину без добавления свежей воды мольная доля тяжелой воды увеличилась бы в $2\alpha/(1+\alpha) \approx 1,015$ раз. Рассчитаем, например, сколько раз нужно

выпаривать воду наполовину без доливания свежей воды, чтобы увеличить концентрацию тяжёлой воды в 10 раз: $\left(\frac{2\alpha}{1+\alpha}\right)^n = 10$.

Решением этого уравнения является $n \approx 157$ раз.

Найдём соотношение между x_{n+1} и x_n в случае доливания $v/2$ молей свежей воды с содержанием тяжёлой воды \square_0 молей. Тогда мольная доля тяжёлой воды составит

$$x_{n+1} = \frac{x + x_0}{2} = \frac{x_n}{1+1/\alpha} + \frac{x_0}{2}.$$

Решая это уравнение, найдём x_n при любых значениях n

$$x_n = x_0 \frac{1+\alpha}{2} - x_0 \frac{\alpha}{2} \left(\frac{\alpha}{1+\alpha}\right)^n.$$

Переходя к пределу при $n \rightarrow \infty$, получаем, что сколько бы раз мы ни повторяли процедуру выпаривания и доливания свежей воды, мы не сможем получить мольное содержание тяжёлой воды больше чем

$$x_0 \frac{1+\alpha}{2} = 0,016 \cdot \frac{1+1,03}{2} = 0,0162\%.$$

Как изменятся результаты, если выпарить третью или k -ую часть воды? Аналогичные рассуждения приводят к следующим результатам. После каждого такого выпаривания без добавления свежей воды мольная доля тяжелой воды увеличилась бы в $k\alpha / (1+(k-1)\alpha)$ раз. Например, чтобы увеличить концентрацию тяжелой воды в 10 раз необходимо 236 раз выпаривать воду на треть без доливания свежей воды.

В случае выпаривания и доливания свежей воды, мольное содержание тяжёлой воды не превысит величину

$$x_0 \frac{1+(k-1)\alpha}{k}$$

и будет колебаться в пределах природного значения x_0 .

Таким образом, при разумных масштабах выпаривания концентрация тяжёлой воды увеличивается крайне незначительно – слишком мал коэффициент разделения, что не даёт достаточного количества тяжёлой воды, способной нанести вред человеку.

ЛИТЕРАТУРА

1. Похлебкин В. В. Чай, его свойства и употребление. М.: Центрполиграф, 2007. 121 с.

2. Еремин В. В. Теоретическая и математическая химия для школьников. Подготовка к химическим олимпиадам. М.: МЦНМО, 2007. 392 с.

УДК 514.87

Студ. В. С. Березовская,
Науч. рук. доц. Л. Д. Яроцкая
(кафедра высшей математики, БГТУ)

ФРАКТАЛЬНОСТЬ КРИСТАЛЛОВ, ДЕНДРИТОВ И ДЕНДРИМЕРОВ

Фрактал означает структуру, возникающую, когда части целого соединяются друг с другом так, что возникает инвариантность по отношению к масштабу. Про такие фигуры говорят, что они самоподобны, моделируют сами себя. Принято различать регулярные и нерегулярные фракталы, из которых первые являются плодом воображения, а вторые – продуктом природы или деятельности человека. Нерегулярные фракталы сохраняют способность к самоподобию в ограниченных пределах, определяемых реальными размерами системы.

Отметим, что фракталы обладают непривычными свойствами. В 1904 году швед Хельге фон Кох придумал непрерывную кривую, которая нигде не имеет касательной. Один из вариантов этой кривой носит название «снежинка Коха». Она обладает периметром бесконечной длины, хотя ограничивает конечную площадь. Эта фигура – один из первых исследованных учеными фракталов. Термин «фрактал» введен Б. Мандельбротом в 1975 году.

Фракталы являются могущественным средством описания сложных структур и процессов. Яркими примерами фракталов в химии являются кристаллы, дендриды и дендримеры. Целью данной работы является обсуждение особенностей фрактальных структур некоторых химических процессов и веществ.

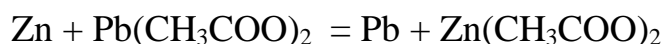
Кристалл – твердое тело, атомная структура которого обладает трансляционной периодичностью. Помимо периодичности кристаллы часто обладают и другими элементами симметрии (осевыми, плоскостными, инверсионными). Сами продукты кристаллизации обладают исключительным многообразием форм. Кристаллы могут расти из растворов, из расплавов и из пара. Катаплеит, очень редкий циркониевый минерал, образуется в щелочных магматических породах. Кристаллы дигидрата $\text{NaCl} \cdot 2\text{H}_2\text{O}$ образуются в соленых озерах в зимнее время, когда температура достаточно низкая для скопления минерала, получившего название гидрогалит. Кристаллы

искусственного дендрита магния получены осаждением паров магния на холодной поверхности. Первые кристаллики льда на стекле – это всегда шестилучевые звездочки или шестигранные стебельки, которые, разрастаясь и взаимодействуя друг с другом, образуют скелетные или ветвистые кристаллы, называемые дендритами.

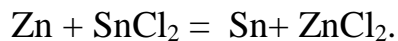
Дендрит представляет собой ветвящееся и расходящееся в стороны образование, возникающее при ускоренной или стеснённой кристаллизации в неравновесных условиях, при этом кристалл расщепляется по определённым законам. Когда начинается процесс кристаллизации в переохлажденном веществе, то энергия системы изменяется в двух противоположных направлениях. Образование новой поверхности раздела приводит к увеличению поверхностного натяжения и уменьшению выделения теплоты кристаллизации. Оба эти процесса нелинейны, и если при определенных условиях их характеристические времена оказываются близкими друг к другу, то возникают благоприятные условия для взаимосогласованного поведения частей системы в процессе кристаллизации и образования в ней дендритных структур с фрактальными свойствами.

Процесс образования дендрита принято называть дендритным ростом, при этом рост дендритов идет в строго определенных направлениях, а ветвление происходит приблизительно через одинаковые промежутки, которые становятся все короче с повышением порядка ветвей. В процессе дендритного развития объекта кристаллографическая закономерность изначального кристалла утрачивается по мере его роста. Дендриты могут быть трёхмерными объёмными (в открытых пустотах) или плоскими двумерными (например, если растут в тонких трещинах горных пород).

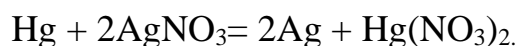
Приведем описание опытов получения некоторых дендритов металлов. Чтобы вырастить «Сатурново дерево», наливают в высокий стакан водный раствор 25 – 30 г ацетата свинца в 100 мл воды и погружают в него очищенную тонкой наждачной бумагой пластину или стержень из цинка. С течением времени на цинковой поверхности вырастают ветвистые и блестящие сросшиеся между собой кристаллы свинца. Их появление вызвано реакцией восстановления свинца из соли более активным в химическом отношении металлом.



Образование кристаллов олова на кусочках цинка называют «деревом Юпитера». Для этого в высокий стеклянный сосуд наливают водный раствор 30 – 40 г хлорида олова SnCl_2 в 100 мл воды и погружают цинковую пластинку.



Серебряное «дерево Дорфмана» получается, если в стеклянный стакан с каплей ртути на дне налить 10%-й водный раствор нитрата серебра AgNO_3 . Сначала ртуть покрывается серой пленкой амальгамы серебра, а через 5 – 10 секунд на ней быстро начинают расти блестящие игольчатые кристаллы серебра. Спустя несколько минут иглы начинают ветвиться, а через час в сосуде вырастает сверкающее серебряное дерево. Отметим, что здесь очень важно точно соблюсти рекомендованную концентрацию нитрата серебра: при более низком содержании AgNO_3 роста кристаллов металлического серебра не наблюдается, а при более высоком – кристаллизация серебра идет без образования ветвистых кристаллов. Но сохранить эти кристаллы не получится, т.к. они очень хрупкие и тонкие



Фрактальные крупные молекулы, синтезированные в органических и в неорганических системах, получили название дендримеров. Это макромолекулы с симметричной древообразной с регулярными ветвлениями структурой. Дендример становится похожим на дерево с шарообразной кроной, если в процессе роста полимерной молекулы не происходит соединения растущих ветвей.

Дендримеры активно исследуются с 1980-х годов, и с тех пор получены уже сотни различных типов подобных макромолекул. Дендримеры 3-го и более высоких поколений характеризуются высокой плотностью молекулярной структуры и имеют близкую к сферической форму. Растворы дендримеров обладают гораздо меньшей вязкостью, чем растворы других веществ с такой же молекулярной массой. Благодаря контролируемым и воспроизводимым с большой точностью размерам макромолекул, дендримеры уже используются для калибровки в масс-спектрометрии, электронной и атомной спектроскопии, ультрафильтрации. Дендримеры, содержащие тяжелые металлы, используются в медицинской диагностике в качестве рентгеноконтрастного вещества. А возможность прикрепления к дендримерам сразу нескольких различных молекул нашла свое применение при лечении злокачественных опухолей.

Таким образом, что фрактальные самоподобные структуры по существу описывают различные пути возникновения порядка в сложных системах. Во всем, что нас окружает, мы часто видим хаос, но на самом деле это не случайность, а идеальная форма, разглядеть которую нам помогают фракталы.

ЛИТЕРАТУРА

1. Пайтген Х., Рихтер П.Х. Красота фракталов. М.: Мир, 1993.
2. Третьяков Ю.Д. Дендриты, фракталы и материалы // Соросовский образовательный журнал, № 11, 1998г., с. 96-102.

УДК 519.173.5:547.022

Студ. П. В. Пашковский,
Науч.рук. канд. физ-мат наук Яроцкая Л.Д.
(кафедра высшей математики, БГТУ)

ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ МОЛЕКУЛ ДЛЯ АНАЛИЗА СВОЙСТВ ВЕЩЕСТВ НА ПРИМЕРЕ БУТАНА, ПЕНТАНА И ИХ ИЗОМЕРОВ

Одной из основных задач органической химии является установление связи между строением вещества и его свойствами. Методы математического моделирования позволяют найти количественные соотношения между структурой и анализируемыми свойствами вещества. В частности, способы описания структуры молекул сводятся к обработке методами теории графов соответствующих им абстрактных математических структур – молекулярных графов.

Целью данной работы является построение молекулярных графов для некоторых алканов и анализ связи между структурой молекул и физико-химическими свойствами посредством топологических индексов.

Граф – сложная геометрическая схема, состоящая из совокупности точек (вершин), соединённых линиями (рёбрами). Связный неориентированный граф, вершинами которого служат атомы углерода, а ребра связями между ними, является молекулярным графом. Если его вершины непомечены, то граф отражает только структуру, а если помечены – структуру и состав. Если рёбра молекулярного графа непомечены, то различия между одинарными и кратными химическими связями нет.

Как математические объекты графы описываются числами, которые в химии называют топологическими индексами. Топологические индексы находят разнообразное применение в структурной химии. В частности, они могут быть использованы для кодирования химической информации, при планировании химического эксперимента в теории электронного строения и реакционной способности молекул, для количественного описания химических структур при анализе связи между структурой молекулы и её свойствами.

Для построения многих топологических индексов используют матрицу расстояний $D = (d_{ij})$, $i, j = 1, \dots, n$, где d_{ij} – минимальное число ребер, которое надо пройти, двигаясь по ребрам графа из i -ой вершины в j -ую. Посредством элементов матрицы расстояний индекс Винера W определяется по формуле

$$W = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n d_{ij}.$$

Статистический анализ углеводородов показал [2], что индекс Винера коррелирует с некоторыми физическими свойствами алканов: молярным объемом, температурами кипения, теплотами испарения и другими. Например, для температуры кипения алканов получена следующая формула:

$$\log t_0 - \log t_0 = \frac{98(\log p - \log p_0)}{p^2} + 5,5 \cdot (\log p - \log p_0),$$

где $\log t_0 = 745,42 \cdot \log(p + 4,4) - 689,4$, $\log p_0 = \frac{1}{6} \cdot (p^3 - p)$, $\log p_0 = \log p - 3$,

W – индекс Винера, p – число полярности, t_0 , W_0 , p_0 – те же характеристики нормального алкана, n – число атомов углерода в молекуле.

Другой тип индексов зависит от степеней вершин молекулярного графа. Индекс молекулярной связности или индекс Рандича определяется по формуле:

$$W = \sum (\log p_i \cdot \log p_j)^{-\frac{1}{2}},$$

где $\log p_i$ – степень i -ой вершины, то есть число ребер, от нее отходящих.

Теплота образования алканов может быть приближенно описана формулой

$$\Delta \log p_0, \frac{\text{ккал}}{\text{моль}} = 12 \cdot n - 11 \cdot n - 9,$$

где n – число атомов углерода в молекуле.

Построим матрицу расстояний и рассчитаем топологические индексы для двух изомерных углеводородов C_4H_{10} : бутана и его изомера. Для этого изобразим их молекулярные графы и занумеруем вершины в произвольном порядке. Атомы водорода в таких графах не указываются, так как их расположение можно однозначно установить по структуре углеродного скелета. От каждой вершины графа может отходить не более четырех ребер, так как углерод в органических соединениях четырехвалентен. Молекулярные графы для бутана и изобутана имеют следующий вид соответственно:

$$G_1: \begin{array}{cccc} 0 & -0 & -0 & -0 \\ 1 & 2 & 3 & 4 \end{array}, \quad G_2: \begin{array}{ccc} 1 & 2 & 3 \\ 0 & -0 & -0 \\ & |4 & \\ & 0 & \end{array}.$$

Диагональные элементы матрицы расстояний для углеводородов равны 0. В первом графе вершина 1 связана с вершиной 2 одним ребром, поэтому элемент матрицы $d_{12}=1$. Аналогично, $d_{13}=2$, $d_{14}=3$. Полные матрицы расстояний для двух графов

$$D(G_1) = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix}, \quad D(G_2) = \begin{pmatrix} 0 & 1 & 2 & 2 \\ 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 2 \\ 2 & 1 & 2 & 0 \end{pmatrix}.$$

Для указанных выше графов, соответствующих молекулам $\square_4 \square_{10}$, индекс Винера принимает значения $W(G_1)=10$ и $W(G_2)=9$. Рассчитаем индекс Рандича:

$$\chi(G_1) = \frac{1}{\sqrt{v_1 v_2}} + \frac{1}{\sqrt{v_2 v_3}} + \frac{1}{\sqrt{v_3 v_4}} = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2 \cdot 2}} + \frac{1}{\sqrt{2}} = 1,914,$$

$$\chi(G_2) = \frac{1}{\sqrt{v_1 v_2}} + \frac{1}{\sqrt{v_2 v_3}} + \frac{1}{\sqrt{v_2 v_4}} = \frac{1}{\sqrt{1 \cdot 3}} + \frac{1}{\sqrt{1 \cdot 3}} + \frac{1}{\sqrt{1 \cdot 3}} = 1,732.$$

Построим матрицу расстояний и рассчитаем топологические индексы для трех изомерных углеводородов состава C_5H_{12} : пентана, изопентана и неопентана. Их молекулярные графы имеют вид:

$$G_3: \begin{array}{ccccc} 0 & -0 & -0 & -0 & -0 \\ 1 & 2 & 3 & 4 & 5 \end{array}, \quad G_4: \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 0 & -0 & -0 & -0 \\ & | & & \\ & 05 & & \end{array}, \quad G_5: \begin{array}{ccc} & 04 & \\ 1 & 2 & | & 3 \\ 0 & -0 & -0 & \\ & | & & \\ & 05 & & \end{array}.$$

Полные матрицы расстояний для этих графов

$$D(G_3) = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 1 \\ 4 & 3 & 2 & 1 & 0 \end{pmatrix}, \quad D(G_4) = \begin{pmatrix} 0 & 1 & 2 & 3 & 2 \\ 1 & 0 & 1 & 2 & 1 \\ 2 & 1 & 0 & 1 & 2 \\ 3 & 2 & 1 & 0 & 3 \\ 2 & 1 & 2 & 3 & 0 \end{pmatrix},$$

$$D(G_5) = \begin{pmatrix} 0 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 1 & 1 \\ 2 & 1 & 0 & 2 & 2 \\ 2 & 1 & 2 & 0 & 2 \\ 2 & 1 & 2 & 2 & 0 \end{pmatrix}.$$

Для указанных выше графов, соответствующих пентанам C_5H_{12} , индекс Винера принимает значения $W(G_3)=20$, $W(G_4)=18$ и

$W(G_5)=16$, а индекс Рандича – $\chi(G_3)=2,414$, $\chi(G_4)=2,270$, $\chi(G_5)=2,000$ соответственно.

Таким образом, сравнивая топологические индексы бутана, пентана и их изомеров, мы видим, что с увеличением степени разветвленности углеродного скелета индексы уменьшаются: наибольшие значения соответствуют наименее разветвленным углеводородам. С увеличением длины углеродного скелета топологические индексы увеличиваются, так как в матрице расстояний становится больше элементов. Рассчитав какой-либо топологический индекс, можно описать физико-химические свойства вещества, так как энергия межмолекулярных взаимодействий зависит от корректных оценок размеров молекулы и степени их разветвлённости.

ЛИТЕРАТУРА

1. Ерёмин В.В. Математика в химии. М.: МЦНМО, 2011. 64 с.
2. Химические приложения топологии и теории графов. Пер. с англ. Под ред. Р. Кинга. М.: Мир, 1987. 560 с.
3. Станкевич М. И., Станкевич И. В., Зефиоров Н. С. Топологические индексы в органической химии // Успехи химии. 1988. Т. 57, вып. 3. С. 337–366.

УДК 51-7:635

Магистрант А.А.Ярошук
Науч. рук. доц. В.В.Игнатенко
(кафедра высшей математики, БГТУ)

ОПТИМАЛЬНОЕ РАСПОЛОЖЕНИЕ РАСТЕНИЙ ПРИ ПОСАДКЕ

В последнее время всё больше земельных площадей отводится для выращивания различных культурных растений. С учётом того, что для этого необходимы очень плодородные почвы, найти эти площади не всегда представляется лёгкой задачей. В связи с этим, следует стремиться к рациональному использованию плодородных площадей. Необходимо использовать их максимально эффективно. Один из приёмов, позволяющих осуществить такое пользование – оптимальное расположение растений.

Под оптимальным расположением растений (при посадке) мы будем понимать такое расположение растений, при котором площадь, занимаемая некоторым количеством растений, будет минимальной и при этом достаточной для их нормального роста и развития. Здесь и далее мы будем рассматривать сплошную посадку без междурядий.

Для нахождения площади, которую занимают растения при посадке, необходимо составить математическую модель расположения растений. Важным является следующий факт: при однородном распределении питательных веществ в почве, растение развивается с одинаковой скоростью во всех направлениях, параллельных плоскости земли. Это даёт нам основание считать, что минимальная площадь, необходимая для нормального роста и развития отдельного растения может быть представлена в виде круга, где в центре находится само растение. Можно сказать, что для нормального роста и развития растений, эти круги не должны пересекаться друг с другом.

Таким образом, мы приходим к математической модели расположения растений. При обычной посадке, растения располагаются в рядах, которые можно считать перпендикулярными (рисунок 1).

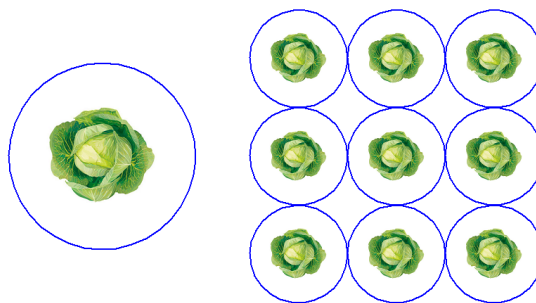


Рисунок 1 – Модель расположения растений при обычной посадке

Т. е. каждое растение находится в узле прямоугольной сетки. Для максимально эффективного использования площади следует стремиться к наиболее плотному заполнению плоскости кругами. Для этого число кругов, соприкасающихся с одним конкретным кругом (контактное число) должно быть максимальным. Возникает вопрос: «Как много одинаковых кругов

можно разместить вокруг одного, того же диаметра?» Ответ – 6. Потому что 1 круг «занимает» 60 градусов дуги, а окружность имеет 360 градусов. Т. е. $360:60 = 6$, что показано на рисунке 2. Таким образом, данное расположение растений будет являться оптимальным.

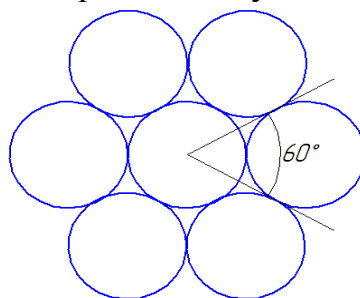


Рисунок 2 – Оптимальное расположение растений

Возникают следующие вопросы:

1) Какова величина экономии площади при посадке с оптимальным расположением растений по сравнению с обычной посадкой?

2) Какова степень использования площади при различных схемах посадки?

Располагать растения можно по-разному. Рассмотрим несколько форм посадки: квадрат, ромб, треугольник, шестиугольник (рисунок 3).

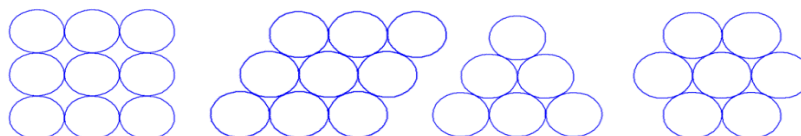


Рисунок 3 – Формы посадки

Площадь данных фигур представляет собой сумму площадей кругов и площадей лунок, т. е. фигур, которые располагается между кругами, площадь которых растениями не используется. Далее будем называть эти фигуры лунками: четырёхугольными лунками, в случае квадрата и треугольными лунками, в случае треугольника, ромба, шестиугольника. Как известно, площадь круга вычисляется по формуле 1.

$$S_{\text{круга}} = \pi r^2 \quad (1)$$

Где r – радиус круга. Можем записать формулы для вычисления площади лунок через радиусы образующих их окружностей. Для лунки в квадратной форме участка, площадь можно найти вычитанием площади четырёх секторов с центральным углом в 90° и радиусом r , из площади квадрата со стороной $2r$. В результате всех необходимых преобразований получим формулу 2.

$$S_{\text{кв.лунки}} = r^2(4 - \pi) \quad (2)$$

Для треугольной лунки площадь находим вычитанием площади трёх секторов с центральным углом в 60° и радиусом r , из площади равностороннего треугольника со стороной $2r$. В результате всех необходимых преобразований получим формулу 3.

$$S_{\text{тр.лунки}} = r^2 \left(\sqrt{3} - \frac{\pi}{2} \right) \quad (3)$$

Зная число рядов n в каждой из фигур, можем записать выражения для нахождения числа кругов и числа лунок в каждой конкретной фигуре. Зная число кругов и число лунок, можем найти площадь всей фигуры. Для квадрата эти формулы будут иметь следующий вид:

$$n_{\text{кругов(к)}} = n^2, \quad (4)$$

$$n_{\text{лунок(к)}} = (n - 1)^2, \quad (5)$$

$$S_{\text{квадрата}} = \pi r^2 n^2 + r^2 (n - 1)^2 (4 - \pi). \quad (6)$$

Где n – число рядов в фигуре.

Для ромба имеем следующие формулы:

$$n_{\text{кругов(р.)}} = n^2, \quad (7)$$

$$n_{\text{лунок(р.)}} = 2(n-1)^2, \quad (8)$$

$$S_{\text{ромба}} = \pi r^2 n^2 + 2r^2(n-1)^2 \left(\sqrt{3} - \frac{\pi}{2} \right). \quad (9)$$

Для треугольника получаем:

$$n_{\text{кругов(т.)}} = \frac{1}{2}(n^2 + n), \quad (10)$$

$$n_{\text{лунок(т.)}} = (n-1)^2, \quad (11)$$

$$S_{\text{треугольника}} = \frac{1}{2} \pi r^2 (n^2 + n) + r^2 (n-1)^2 \left(\sqrt{3} - \frac{\pi}{2} \right). \quad (12)$$

И для шестиугольника:

$$n_{\text{кругов(ш.)}} = \frac{1}{2}(n^2 + n) + 2(n^2 - n) + \frac{1}{2}(n-1)(n-2), \quad (13)$$

$$n_{\text{лунок(ш.)}} = 6(n-1)^2, \quad (14)$$

$$S_{\text{шестиугольника}} = \pi r^2 n_{\text{кругов(ш.)}} + 6r^2(n-1)^2 \left(\sqrt{3} - \frac{\pi}{2} \right). \quad (15)$$

Далее найдём площадь, которая приходится на одно растение в каждой фигуре:

$$S_{1\text{к.}} = \frac{S_{\text{квадрата}}}{n_{\text{кругов(к.)}}}, \quad (16)$$

$$S_{1\text{р.}} = \frac{S_{\text{ромба}}}{n_{\text{кругов(р.)}}}, \quad (17)$$

$$S_{1\text{т.}} = \frac{S_{\text{треугольника}}}{n_{\text{кругов(т.)}}}, \quad (18)$$

$$S_{1\text{ш.}} = \frac{S_{\text{шестиугольника}}}{n_{\text{кругов(ш.)}}}. \quad (19)$$

Тогда можем найти величину экономии площади E для каждой из фигур, по сравнению с обычной квадратной посадкой:

$$E_{\text{р.}} = \frac{S_{1\text{к.}} - S_{1\text{р.}}}{S_{1\text{к.}}} \cdot 100 \%, \quad (20)$$

$$E_{\text{т.}} = \frac{S_{1\text{к.}} - S_{1\text{т.}}}{S_{1\text{к.}}} \cdot 100 \%, \quad (21)$$

$$E_{\text{ш.}} = \frac{S_{1\text{к.}} - S_{1\text{ш.}}}{S_{1\text{к.}}} \cdot 100 \%. \quad (22)$$

Становится интересным вопрос: «Какова максимальная величина экономии площади при неограниченном возрастании числа рядов (растений)?»

Для ответа на него, найдём предельные значения величин экономии площади E_{max} :

$$E_{(max)р.} = \lim_{n \rightarrow \infty} \frac{S_{1к.} - S_{1р.}}{S_{1к.}} \cdot 100 \%, \quad (23)$$

$$E_{(max)т.} = \lim_{n \rightarrow \infty} \frac{S_{1к.} - S_{1т.}}{S_{1к.}} \cdot 100 \%, \quad (24)$$

$$E_{(max)ш.} = \lim_{n \rightarrow \infty} \frac{S_{1к.} - S_{1ш.}}{S_{1к.}} \cdot 100 \%. \quad (25)$$

Получаем следующий результат:

$$E_{(max)р.} = E_{(max)т.} = E_{(max)ш.} = \frac{(2 - \sqrt{3})}{2} \cdot 100 \% \approx 13,4 \%.$$

Оценим степень использования площади при различных схемах посадки. Под степенью использования площади мы будем понимать отношение площади, используемой растениями, ко всей площади участка, выраженное в процентном отношении. Найдём формулы, которые выражают площадь используемых участков, через количество рядов растений. А затем найдём отношение площади, необходимой растениям, к общей площади участков. Схемы участков для различных форм представлены на рисунке 4.

Находим выражения для площади этих участков $S_{у.}$, площади, используемой растениями (полезной площади) $S_{пол.}$ и степени использования пло-

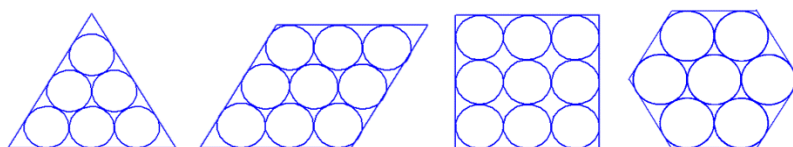


Рисунок 4 – Схемы участков для посадки

площади D . Для квадрата эти выражения будут иметь следующий вид:

$$S_{к.у.} = 4r^2n^2, \quad (26)$$

$$S_{пол.к.} = \pi r^2n^2, \quad (27)$$

$$D_{к.} = \frac{S_{пол.к.}}{S_{к.у.}} \cdot 100 \%. \quad (28)$$

Следует отметить, что для квадратного участка эта величина постоянна и равна $\frac{\pi}{4} \cdot 100 \% \approx 79 \%$. Выражения для участка в форме ромба принимают вид:

$$S_{p.y.} = \frac{2\sqrt{3} \left(2r + r\sqrt{3}(n-1) \right)^2}{3}, \quad (29)$$

$$S_{пол.р.} = \pi r^2 n^2, \quad (30)$$

$$D_p = \frac{S_{пол.р.}}{S_{p.y.}} \cdot 100 \%. \quad (31)$$

Для участка треугольной формы:

$$S_{т.у.} = \frac{\sqrt{3} \left(2\sqrt{3}r + 2r(n-1) \right)^2}{4}, \quad (32)$$

$$S_{пол.т.} = \frac{1}{2} \pi r^2 (n^2 + n), \quad (33)$$

$$D_t = \frac{S_{пол.т.}}{S_{т.у.}} \cdot 100 \%. \quad (34)$$

И для участка шестиугольной формы имеем:

$$S_{ш.у.} = \frac{3\sqrt{3} \left(\frac{2\sqrt{3}r}{3} + 2r(n-1) \right)^2}{2}, \quad (35)$$

$$S_{пол.ш.} = \pi r^2 \left(\frac{1}{2} (n^2 + n) + 2(n^2 - n) + \frac{1}{2} (n-1)(n-2) \right), \quad (36)$$

$$D_{ш.} = \frac{S_{пол.ш.}}{S_{ш.у.}} \cdot 100 \%. \quad (37)$$

Найдём предельные значения степени использования площади D_{max} :

$$D_{(max)р.} = \lim_{n \rightarrow \infty} \frac{S_{пол.р.}}{S_{p.y.}} \cdot 100 \%, \quad (38)$$

$$D_{(max)т.} = \lim_{n \rightarrow \infty} \frac{S_{пол.т.}}{S_{т.у.}} \cdot 100 \%, \quad (39)$$

$$D_{(max)ш.} = \lim_{n \rightarrow \infty} \frac{S_{пол.ш.}}{S_{ш.у.}} \cdot 100 \%. \quad (40)$$

Получаем следующий результат:

$$D_{(max)р.} = D_{(max)т.} = D_{(max)ш.} = \frac{\pi\sqrt{3}}{6} \cdot 100 \% \approx 90,7 \%$$

Вывод: При оптимальном расположении растений можно сэкономить до 13,4 % площади. Степень использования участка может достигать до 90,7 %. При достаточно большом числе растений, степень использования и экономия площади не зависят от геометрии используемого участка.

С точки зрения экологии благоприятно такое размещение, при котором конкуренция между растениями возникает как можно позже. Это возможно при равном расстоянии между растениями и квадратном или ромбовидном размещении. Однородное распределение семян с помощью сеялок точного высева обеспечивает равную силу роста отдельных растений [1]. Помимо применения для выращивания различных овощей, таких как свёкла, репа, морковь, лук, капуста, тыква и многие другие, данная модель может быть применена и при лесопосадке: такая модель считается самой лучшей для выращивания деревьев [2]. Эта модель может эффективно применяться при выращивании зерновых с помощью сеялок точного высева.

Модель также может использоваться и в ряде других случаев: в логистике, при размещении грузов цилиндрической формы, в экологии, при проектировании очистных сооружений, когда аппараты для очистки имеют круглое поперечное сечение (циклоны, отстойники).

ЛИТЕРАТУРА

1. Круг, Г. Овощеводство / Г. Круг ; пер. с нем. В. И. Леунова. - Москва : Колос, 2000. - 574 с.
2. Мирон, К. Ф. Посев и посадка леса / К. Ф. Мирон ; Общественный университет лесного хозяйства ВНИТОЛЕС. – Ленинград : Государственное лесотехническое издательство, 1947. - 80 с.

УДК 511.34

Студ. Н.В. Чистяков

Науч. рук. доц. Е.И. Ловенецкая
(кафедра высшей математики, БГТУ)

АДДИТИВНЫЕ ЦЕПОЧКИ

Рассмотрим следующую задачу: необходимо возвести число C в степень N наименьшим количеством действий.

Очевидно, что с помощью обычного последовательного возведения в степень потребуется $N - 1$ действие. Если количество данных операций в алгоритме велико, то это уменьшит производительность программы, поэтому желательно найти решение получше.

При решении задачи будем считать простыми операциями возведение числа в квадрат, а также умножение текущего значения на уже известное.

Рассмотрим рекурсивный алгоритм возведения в степень. В нем используется возведение в квадрат для получения четной степени и умножение на исходное основание C – для нечетной, поэтому потребуется всего 2 новые переменные: результат возведения в степень и текущая степень. Обозначим через $F_R(N)$ количество простых операций, необходимых для возведения исходного числа в степень N . Легко видеть, что

$$F_R(1) = 0; \quad F_R(2k) = F_R(k) + 1; \quad F_R(2k + 1) = F_R(2k) + 1.$$

Данный алгоритм не требует большого количества памяти и операций, однако он не дает наилучшего результата в данной задаче. Первым примером является $N = 15$. Рекурсивный метод дает решение за 6 действий:

$$C^{15} = ((C^2 * C)^2 * C)^2 * C,$$

однако возведение числа в 15-ю степень можно выполнить за 5 операций:

$$C^{15} = ((C^2)^2 * C)^3$$

(возведение в 3-ю степень представляет собой умножение исходного выражения на квадрат самого себя).

Для поиска лучшего результата будем использовать понятие аддитивной цепочки[1]. Аддитивность – свойство величин, состоящее в том, что значение величины, соответствующее целому объекту, равно сумме значений величин, соответствующих его частям.

Аддитивной цепочкой для числа N называется последовательность натуральных чисел $1, 2, a_2, a_3, \dots, a_l = N$, где каждый следующий член получается путем сложения каких-либо двух предыдущих:

$$a_i = a_j + a_k, \quad 0 \leq k \leq j < i, \quad i = 1, 2, \dots, l,$$

при этом число l называется длиной аддитивной цепочки. Будем обозначать $F(N)$ минимальное из чисел l , т.е. наименьшую длину аддитивной цепочки для числа N .

Как видно из примера для $N = 15$, рекурсивный метод не всегда позволяет получить цепочку наименьшей длины, т. е. $F(N) \leq F_R(N)$. Рассмотрим в качестве альтернативы рекурсивному методу бинарный (двоичный) метод решения поставленной задачи:

- 1) записать число N в двоичном представлении;
- 2) если стоит цифра 1, то вместо нее в итоговую строчку пишется SX, а если 0, то пишется S;
- 3) далее идет проход по получившейся строке: если X, то текущее значение умножается на исходное, а если S, то число возводится в квадрат.

Метод можно легко обосновать, проанализировав последовательность степеней при вычислении: если рассмотреть S как умножение на 2, а X как прибавление единицы, то получившаяся цепочка дает из единицы нужную степень N .

Несложно рассчитать количество операций бинарного метода $F_B(N)$, введя две вспомогательные функции: $\lambda(n) = \lceil \log_2 N \rceil$ – уменьшенная на единицу длина двоичной записи числа N , $\nu(n)$ – количество единиц в двоичном представлении N , которые могут быть вычислены рекурсивно:

$$\lambda(1) = 0; \quad \lambda(2N) = \lambda(2N + 1) = \lambda(N) + 1;$$

$$\nu(1) = 1; \quad \nu(2N) = \nu(N); \quad \nu(2N + 1) = \nu(N) + 1.$$

$$\text{Тогда } F_B(N) = \lambda(N) + \nu(N) - 1.$$

Бинарный метод требует память только для хранения C и текущего промежуточного результата, поэтому хорошо подходит для использования в программе. Однако можно убедиться в том, что $F_B(15) = F_R(15) > F(15)$, поэтому для уменьшения количества операций требуется рассмотреть другой метод.

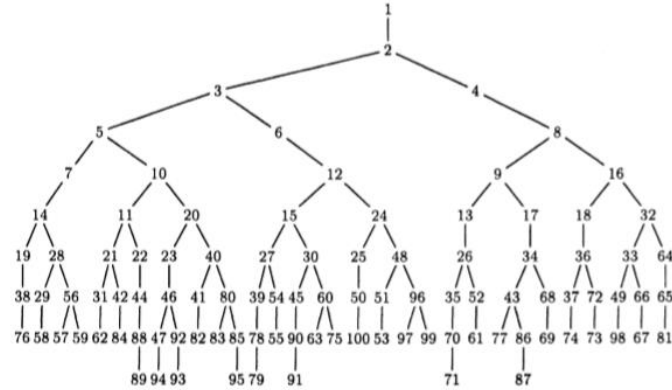
Метод множителей основан на разложении числа N на множители. Пусть $N = pq$. Тогда возведение числа C в степень N примет вид $C^N = (C^p)^q$. Например, если нужно возвести число в степень $N = 55$, то сначала вычисляем $y = C^5 = (C^2)^2 * C$ и $z = y^2$, а затем $C^{55} = y^{11} = (y^2)^5 * y = ((z^2)^2 * z) * y$.

Всего потребуется 8 действий против 9-ти у двоичного метода.

Метод множителей во многих случаях (но не всегда!) позволяет улучшить результат бинарного метода[2]. Таким образом, метод

множителей не гарантирует получение аддитивной цепочки наименьшей длины.

В [2] приводится следующее минимизирующее дерево для возведения числа в степень $N \leq 100$:



Это дерево позволяет для каждого значения N определить некоторую цепочку наименьшей длины и таким образом получить экономный способ вычисления C^N . Например, к числу $N = 15$ из вершины дерева ведет цепочка 1, 2, 3, 6, 12, 15. Поэтому для расчета C^{15} сначала вычисляем $y = C^3 = C^2 * C$, а затем $C^{15} = C^{12} * C^3 = (y^2)^2 * y$.

Цепочка для $N = 55$ имеет вид 1, 2, 3, 6, 12, 15, 27, 54, 55. Следовательно, $C^{55} = (C^{12} * C^3 * C^{12})^2 * C = (z * y * z)^2 * C$, где $y = C^3 = C^2 * C$, $z = (y^2)^2$.

Приведенные примеры показывают, что аддитивная цепочка наименьшей длины для заданного N не единственна. При этом неясно, как гарантированно получить цепочку минимальной длины за разумное время.

Поиск наиболее экономичного способа вычисления C^N и расчета $F(N)$ представляет собой нетривиальную теоретическую и алгоритмическую задачу, возникающую при изучении оптимальных методов вычислений. Проблема определения $F(N)$ впервые была поставлена в 1894 году Х. Деллаком и частично решена Э. де Жонкиэресом методом множителей. В своей работе Э. де Жонкиэрес перечислил значения $F(N)$ для всех простых чисел $N < 200$, но значения, полученные для $N = 107, 149, 163, 179$, были завышены.

Как указано в [1], задача о точном вычислении $F(N)$ до сих пор в общем случае не решена. Легко видеть, что $F(2^n) = n$. Методом множителей можно доказать, что

$$F(N) \leq F(p) + F(q), \quad \text{если } N = pq;$$

бинарный алгоритм обеспечивает оценку

$$F(N) \leq F_B(N) = \lambda(N) + \nu(N) - 1.$$

С другой стороны, можно доказать, что $F(N) \geq \lambda(N)$.

Неизвестно также, существует ли алгоритм полиномиальной сложности для расчета $F(N)$. Гарантировать точное значение может только полный перебор возможностей.

Целью данной работы является сравнение значений $F_R(N)$, $F_B(N)$ и $F(N)$ путем разработки программной реализации поиска необходимого для возведения числа в степень N количества операций тремя методами: рекурсивным, бинарным, полного перебора. Значения $F_R(N)$ и $F_B(N)$ вычисляются, как описано выше. Особенностью реализации алгоритма полного перебора является проверка всех путей к данной степени. Строится дерево, притом ветви друг о друге «не знают». Выписываются все возможные варианты перемножения текущей степени в вершине на известные. Если получается степень больше N , то дальше цепочка не строится. В итоге дерево остается как бы незаконченным в большинстве мест, но в одном или нескольких случаях получается нужная степень. В результате выбирается ветвь с минимальным количеством шагов.

Алгоритм полного перебора должен запоминать все возможные оптимальные способы получения каждой степени, а при проверке оптимальности новой цепочки – определять количество необходимых действий для возведения в степень N с учетом уже найденных аддитивных цепочек наименьшей длины для степеней n и m , где $N = n + m$. Например, для степени 5 есть 2 варианта получения: $N = 2 \cdot 2 + 1$ и $N = 2 + (2 + 1)$. В обоих случаях для получения степени требуется 3 операции. В первом случае в дальнейшем будут известны 1, 2, 4 степени, а во втором – 1, 2, 3 степени. Для возведения числа в 15-ю степень уже существует 4 оптимальных варианта, количество действий для каждого из вариантов равно 5. Так как при наращивании дерева используется умножение на уже известную степень, то необходимо проверять оба варианта получения базовой степени.

Реализация указанных алгоритмов для $N \leq 100$ показала, что при всех этих значениях $F_R(N) = F_B(N)$. Для $N = 15, 23, 27, 30, 31, \dots, 99$ (всего 33 значения) $F_R(N)$ и $F_B(N)$ на единицу больше, чем $F(N)$; при $N = 63$ и $N = 95$ значения $F_R(N)$ и $F_B(N)$ превышают $F(N)$ на 2. Ожидаемо, что полный перебор даст лучшее значение операций, однако он требует больше памяти и времени. Однако если в программе часто используются операции возведения в различные степени, имеет смысл изначально создать минимизирующее дерево операций, и использование полного перебора для составления данного дерева будет являться оптимальным решением.

ЛИТЕРАТУРА

1. Гашков, С.Б. Задача об аддитивных цепочках и ее обобщения / С.Б. Гашков // Математическое просвещение, 2011, выпуск 15. С. 138-153.
2. Кнут, Д.Э. Искусство программирования / Д.Э. Кнут – 3-е изд. – М.: Вильямс, 2012. – 788 с.

УДК 519.171

Студ. М. Ю. Радченко, А. М. Карпач
Науч. рук. Е. В. Терешко
(кафедра высшей математики, БГТУ)

ЧТО ТАКОЕ ГРАФ? ПРИМЕНЕНИЕ ГРАФОВ В ПРОГРАММИРОВАНИИ

Если студент имеет цель стать хорошим специалистом в сфере своей специализации, ему необходимо разбираться во всех дисциплинах, преподаваемых в университете, в не зависимости, являются они предметами по специальности или нет. Именно поэтому проводить параллели и находить нечто общее между, на первый взгляд, ничем не схожими дисциплинами — одна из важнейших задач обучения. Тема «Графов» широко представлена в программировании и большинство программ, которые создаются студентами уже на 1 курсе, основаны, в том числе, и на математических графах.

Граф G — это упорядоченная пара $G:=(V,E)$, где V — это непустое множество вершин или узлов, а E — множество вершин, называемых рёбрами.

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 E являются упорядоченными, например пары $(1, 3)$ и $(3, 1)$ это две разные связи). В свою очередь в неориентированном графе, связи ненаправленные, и поэтому если существует связь $(1, 3)$ то значит что существует связь $(3, 1)$.

Перед нами были поставлены вопросы: для чего нужны графы и что нужно знать, чтобы воспользоваться ими на практике при создании компьютерных программ?

Существует два способа представления графа: в виде списков смежности и в виде матрицы смежности. Оба способа подходят для представления ориентированных и неориентированных графов. И без труда представляются при помощи таких популярных языков программирования, как C++ или же JS.

При матрице смежности происходит заполнение квадратной матрицы A размером $|V| \times |V|$ (где V — количество вершин графа) следующим образом: $A[i][j] = 1$ (Если существует ребро из i в j) и $A[i][j] = 0$ (в обратном случае).

Вторым способом является список смежности. В данном представлении используется массив A , содержащий $|V|$ списков. В каждом списке $A[V]$ содержатся все вершины u , так что между V и u есть ребро.

В интернете, а также в специализированной литературе существует большое количество различных задач, решение которых представляется в виде графов. Часть из них основана на Гамильтоновом графе. Само название «гамильтонов цикл (граф)» произошло от задачи «Кругосветное путешествие», предложенной ирландским математиком Вильямом Гамильтоном в 1859 году. Нужно было, выйдя из исходной вершины графа, обойти все его вершины и вернуться в исходную точку.

В нашей работе была рассмотрена подобная задача и реализовано ее решение с использованием языка программирования C++.

В задаче имелось пять точек, соединенных определенных образом. Цель — посетить каждую только один раз.

Для написания кода была использована программа Visual Studio. После стандартного объявления используемых библиотек и основных переменных, в коде расположена первая основная часть программы — матрица смежности. Она составляется в виде двумерного массива A , в который входят целочисленные значения типа int — 1 или 0.

Основная функция кода `main` содержит небольшое меню приветствия. Здесь пользователю предоставляется возможность выбрать вершину, с которой будет начат путь. Далее на экран выводится матрица смежности для лучшего понимания задачи и выполняется переход на поиск решения. Этот переход обусловлен рекурсивной функцией `gamilton()`. Именно благодаря данному типу функции возможен поиск необходимого «пути движения» для решения поставленной задачи. Рекурсивная функция — это функция с такой организацией работы, при которой она вызывает сама себя.

Вся программа основана на условиях проверки и различных циклах, которые в конечном итоге выводят на экран оптимальное решение задачи, т.е. ряд чисел, каждое из которых значит определенную вершину гамильтонового графа.

Созданный код является универсальным, и, имея на руках любую задачу подобного типа, необходимо лишь изменить матрицу

смежности в самом начале программы. Дальнейшая работа уже не зависит от количества вершин и связей (ребер) между ними.

Из всего следует, что графы являются очень универсальным средством, которое помогает нам как в решении задач при изучении высшей математики, так и в программировании для написания наиболее оптимального кода. И умение сочетать различные дисциплины и изучать практические стороны различных тем — отличный способ совершенствоваться и достигать наилучшего результата.

ЛИТЕРАТУРА

1. Пацей, Н.В. Основы алгоритмизации и программирования: учеб.-метод. пособие для студентов специальности «Информационные системы и технологии (издательско-полиграфический комплекс)» / Н.В. Пацей. — Минск: БГТУ, 2010. — 289 с.

2. Новиков, Ф. А. Дискретная математика: Учебник для вузов. 3-е изд. Стандарт третьего поколения. — СПб.: Питер, 2017. — 334 с.

УДК 519.171

Студ. В. В. Хорхалёв

Науч. рук. И. К. Асмыкович

(кафедра высшей математики, БГТУ)

ЭЛЛИПТИЧЕСКИЕ КРИВЫЕ И ИХ ПРИЛОЖЕНИЯ В КРИПТОГРАФИИ

На протяжении более сотни лет эллиптические кривые исследовались лишь узкими специалистами в области теории чисел. Но в 1985 г. одновременно и независимо Н. Коблиц и В. Миллер предложили использовать эллиптические кривые для построения криптосистем с открытым ключом. После этого интерес к эллиптическим кривым стремительно возрос. Эллиптические кривые применяются в разных областях криптографии таких, как теория кодирования, генерация псевдослучайных последовательностей, алгоритмическая теория чисел для построения тестов на простоту и, для создания метода факторизации целых чисел Ленстры.

Пусть F_q , $q = p^k$, конечное поле характеристики $p \geq 2$. Эллиптической кривой над полем F_q называется множество точек $(x, y) \in F_q \oplus F_q$, удовлетворяющих уравнению Вейерштрассе ¹:

$$y^2 + ay + b = x^3 + cx^2 + dx + e.$$

К множеству точек эллиптической кривой также добавляется точка, называемая точкой в бесконечности и обозначаемая символом ∞ . Если характеристика поля $p \geq 3$ (а мы будем рассматривать именно этот случай), уравнение, путем замены переменных, может быть

преобразовано в уравнение, называемое сокращенным уравнением Вейерштрассе:

$$y^2 = x^3 + ax + b,$$

где $a, b \in \mathbb{F}_q$.

Для эллиптической кривой могут быть вычислены ее дискриминант Δ и инвариант j :

$$\Delta = -16(4a^3 + 27b^2);$$

$$j = \frac{1728(4a)^3}{b^2},$$

Заметим: $1728 = 2^6 \cdot 3^3$.

Кривая называется неособой, если $\Delta \neq 0$ и, как следствие, многочлен $x^3 + ax + b$, стоящий в правой части уравнения кривой, не имеет кратных корней. Мы будем рассматривать только неособые кривые. Инвариант j определяет изоморфизм кривых: две кривые с одинаковым инвариантом являются изоморфными.

Арифметические операции с точками на эллиптической кривой

На множестве точек E неособой эллиптической кривой может быть определена групповая операция суммирования $+$, с помощью которой это множество становится аддитивной абелевой группой. Нулем этой группы является бесконечно удаленная точка ∞ , а обратным элементом к точке $P = (x, y) \in E$ будет являться точка $-P = (x, -y)$. Опишем геометрическое определение операции суммирования. Пусть $P_1(x_1, y_1)$ и $P_2(x_2, y_2)$ – произвольные точки, и $P_3(x_3, y_3)$ обозначает сумму этих точек $P_3 = P_1 + P_2$. Через точки $P_1(x_1, y_1)$ и $P_2(x_2, y_2)$ проведем прямую L . Третью точку пересечения с эллиптической кривой обозначим через $P'(x'_3, y'_3)$. Такая точка обязательно существует, т.к. пересечение произвольной прямой с эллиптической кривой имеет либо одну, либо 3 точки пересечения. Определим сумму трех точек $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ и $P_3'(x'_3, y'_3)$ равной нулю (бесконечно удаленной точке):

$$P_1 + P_2 + P' = \infty$$

Тогда $P_3 = P_1 + P_2 = -P'$, откуда $x_3 = x'_3$, $y_3 = -y'_3$. Для вычисления координат точки P_3 , найдем параметры прямой $L : y = kx + d$:

$$k = \frac{y_2 - y_1}{x_2 - x_1}, \quad d = y_1 - kx_1$$

Подставляя выражение для L в уравнение, получим $x^3 + cx^2 + ax + b - (kx + d)^2 = 0$

Сумма координат $x_1 + x_2 + x_3$ должна быть равна коэффициенту при x^2 , взятому с противоположным знаком:

$$x_1 + x_2 + x_3 = k^2 - c = \left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2 - c,$$

откуда получим формулу для координат суммы ¹:

$$\begin{cases} x_3 = k^2 - x_1 - x_2 - c \\ y_3 = k(x_1 - x_3) - y_1 = k(2x_1 + x_2 - k^2 + c) - y_1 \end{cases}$$

где для сокращенного уравнения значение параметра c равно 0.

Если точки P_1 и P_2 совпадают, то прямая L является касательной в $t.P_1$ и угловой коэффициент прямой L можно найти, дифференцируя

уравнение по x . Общие формулы для коэффициента k получают вид ¹:

$$k = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{если } P_1 \neq P_2, \\ \frac{3x_1^2 + a}{2y_1}, & \text{если } P_1 = P_2 \end{cases}$$

Формулы для координат удвоенной точки можно получить, подставляя $x_2 = x_1, y_2 = y_1$:

$$\begin{cases} x_3 = k^2 - 2x_1 - c \\ y_3 = k(x_1 - x_3) - y_1 = k(3x_1 - k^2 + c) - y_1 \end{cases}$$

где опять для сокращенного уравнения значение параметра c равно 0. Группа точек эллиптической кривой над полем F_q обозначается символом $EC(F_q)$, а ее мощность (количество элементов) символом $\#EC(F_q)$.

Известно, что группа точек эллиптической кривой либо является циклической (т.е. найдется точка $P \in EC$ такая, что все точки являются кратными этой точки), либо $E(F_q) \cong Z_{n_1} \oplus Z_{n_2}$, где $n_1 | n_2$, и $n_1 | q - 1$.

Поскольку, арифметика эллиптических кривых не содержит прямых формул для вычисления кратного kQ для заданной точки $Q(x_1, y_1)$, то эту операцию выполняют с использованием операций сложения, вычитания и удвоения точки. Для этого надо представить число k в двоичной системе исчисления $k = b_t b_{t-1} \dots b_0$, $b_i \in \{0, 1\}$, потом вычислить все точки $2Q, 4Q, \dots, 2^t \cdot Q$ и подсчитать сумму тех точек $2^i \cdot Q$, для которых $b_i = 1$.

Пример

Выберем эллиптическую кривую, задаваемую $a = 1$, $b = 1$ над полем F_{23} . Уравнение кривой примет вид:

$$y^2 = x^3 + x + 1.$$

Выберем точки $A = (9, -7)$, $B = (6, -4)$.

Вычислим значение выражения $A - 3B$:

$$3B = B + 2B;$$

$$2B = (k^2 - 2x, k(3x - k^2) - y) = (-10, -7);$$

$$3B = (6, -4) + (-10, -7) = (-4, -11);$$

$$A - 3B = (9, -7) - (-4, -11) = (9, -7) + (4, 11) = (13, 4).$$

ЛИТЕРАТУРА

1. Ш.Т. Ишмухаметов, Р.Г. Рубцова Математические основы защиты информации: Электронное учебное пособие для студентов института вычислительной математики и информационных технологий. – Казань, 2012 г. – 138 с.
2. Ю. Г. Прохоров Эллиптические кривые и криптография: учеб. для вузов – Москва: Механико-математический факультет МГУ, 2007 г. – 144 с.

УДК 519.171

Студ. В.А. Андреюк, Е.А. Баран
 Науч. рук. А. А. Якименко
 (кафедра высшей математики, БГТУ)

ИСТОРИЯ МАТЕМАТИЧЕСКИХ ОБОЗНАЧЕНИЙ

Слово математика пришло к нам из древнегреческого, где означало "учиться", "приобретать знания. Математика – это первая наука, которую смог освоить человек. Самой древней деятельностью был счёт. Некоторые первобытные племена подсчитывали количество предметов с помощью пальцев рук и ног, а так же палочек. Можно сказать, что 1 палочка – это первый математический символ.

С древнегреческого «символ» (греч. *symbolon* – признак, примета, пароль, эмблема) – знак, который связан с обозначаемой им предметностью так, что смысл знака и его предмет представлены только самим знаком и раскрываются лишь через его интерпретацию.

С открытием математических правил и теорем ученые придумывали новые математические обозначения, знаки. Математические знаки - это условные обозначения, предназначенные для записи математических понятий, предложений и выкладок.

1. Знаки сложения, вычитания

История математических обозначений начинается с палеолита. Этим временем датируются камни и кости с насечками, использовавшимися для счета. Наиболее известный пример – кость Ишанго. Знаменитая кость датируемая примерно 20 тысяч лет до новой эры, доказывает, что уже в то время человек выполнял достаточно сложные математические операции. Насечки на кости использовались для сложения и наносились группами, символизируя сложения чисел.

В Древнем Египте была уже намного более продвинутая система обозначений. Например, в папирус Ахмеса в качестве символа сложения используется изображение двух ног, идущих вперед по тексту, а для вычитания - двух ног, идущих назад. Древние греки обозначали сложение записью рядом, но время от времени использовали для этого символ косой черты “/” и полуэллиптическую кривую для вычитания.

Считается, так же, что наш знак $+$ происходит от одной из форм слова “et”, которое по-латыни значит “и”. Выражение $a + b$ писалось на латыни так: **a et b**. Постепенно, из-за частого использования, от знака “et” осталось только “t”, которое, со временем превратилось в “+”. Первым человеком, который, возможно, использовал знак $+$ как аббревиатуру для et, был астроном Николь д’Орем в середине четырнадцатого века.

Обозначения вычитания были более запутанными, так как вместо простого знака “-” в немецких, швейцарских и голландских книгах иногда использовали символ “÷”, которым мы сейчас обозначаем деление. В нескольких книгах семнадцатого века использованы две точки “.” или три точки “...” для обозначения вычитания.

И даже после принятия знака $+$ не все использовали этот символ. Видман (первый использовал знаки «+», «-») сам ввел его как греческий крест (знак, который мы используем сегодня), у которого горизонтальная черта иногда немного длиннее вертикальный. Некоторые математики, такие как Рекорд, Харриот и Декарт, использовали такой же знак. Другие использовали латинский крест «†», иногда расположенный горизонтально, с перекладиной на одном конце или на другом. Наконец, некоторые использовали более декоративный вид «+».

2. Знак равенства

Знак равенства в математике и других точных науках пишут между двумя идентичными по своему размеру выражениями. Первым употребил знак равенства Диофант. Равенство он обозначил буквой i

(от греческого *isos* – равный). В античной и средневековой математике равенство обозначалось словесно, например, *est egale*, или использовали аббревиатуру “ae” от латинского *aequalis* - “равны”. На других языках также использовали первые буквы слова “равный”, но это не было общепринятым. Знак равенства “=” ввел в 1557 году уэльский врач и математик Роберт Рекорд. Математическим символом для обозначения равенства служил в некоторых случаях символ П. Рекорд ввел символ “=” с двумя одинаковыми горизонтальными параллельными отрезками, гораздо более длинными, чем те, что используются сегодня. Некоторое время распространению символа Рекорда мешало то обстоятельство, что такой же символ использовался для обозначения параллельности прямых; в конце концов было решено символ параллельности сделать вертикальным. Распространение знак получил только после работ Лейбница на рубеже XVII—XVIII веков, то есть более чем через 100 лет после смерти впервые использовавшего его для этого Роберт Рекорд. На его могильной плите нет слов – просто вырезан знак «равно».

Родственные символы для обозначения приблизительного равенства “ \approx ” и тождества “ \equiv ” являются совсем молодыми - первый введен в 1885 году Гюнтером, второй - в 1857 году Риманом.

3. Знаки умножения и деления

Знак умножения ввёл в 1631 году Уильям Отред (Англия) в виде косоугольного крестика. До него использовали букву M. Позднее Лейбниц заменил крестик на точку (конец XVII века), чтобы не путать его с буквой x; до него такая символика встречалась у Региомонтана (XV век) и английского учёного Томаса Хэрриота.

Для обозначения действия деления Отред предпочитал косую черту. Двоеточием деление стал обозначать Лейбниц. До них часто использовали также букву D. Начиная с Фибоначчи, используется также черта дроби, употреблявшаяся ещё в арабских сочинениях. Деление в виде обелюс (“ \div ”) ввел швейцарский математик Иоганн Ран.


4. Знак процента.

Символ процента появляется в середине XVII века сразу в нескольких источниках, его происхождение неясно. Есть гипотеза, что он возник от ошибки наборщика, который сокращение *cto* (*cento*, сотая доля) набрал как 0/0. Более вероятно, что это скорописный коммерческий значок, возникший лет на 100 раньше.

5. Знак бесконечности

Нынешний символ бесконечности “ ∞ ” ввел и придумал Джон Уоллис в 1655 году. До сих пор так и не известно, почему он

остановил свой выбор именно на этом знаке. Одна из наиболее авторитетных гипотез связывает происхождение этого символа с латинской буквой «М», которую римляне использовали для обозначения числа 1000. Символ бесконечности назван "lemniscus" (лат. лента) математиком Бернулли приблизительно сорок лет спустя.

Другая версия говорит о том, что рисунок «восьмерки» передает главное свойство понятия «бесконечность»: движение **без конца**. По линиям числа 8 можно совершать, как по велотреку, бесконечное движение. Для того, чтобы не путать введенный знак с числом 8, математики решили располагать его горизонтально. Получилось . Такое обозначение стало стандартным для всей математики, не только алгебры. Почему бесконечность не обозначают нулем? Ответ очевиден: цифру 0 как не поворачивай — она не изменится. Поэтому выбор и пал именно на 8.

Другой вариант - змей, пожирающий свой хвост, который за полторы тысячи лет до нашей эры в Египте символизировал различные процессы, не имеющие начала и конца.

6. Знак корня квадратного

Знак корня впервые употребил немецкий математик Кристоф Рудольф, в 1525 году. Происходит этот символ от стилизованной первой буквы слова *radix* (корень). Черта над подкоренным выражением вначале отсутствовала; её позже ввёл Декарт для иной цели (вместо скобок), и эта черта вскоре слилась со знаком корня.

7. Скобки

Круглые скобки появились в 1556 году у Тартальи (для подкоренного выражения) и позднее у Жирара. Одновременно Бомбелли использовал в качестве начальной скобки уголок в виде буквы L, а в качестве конечной — его же в перевёрнутом виде (1550); такая запись стала прародителем квадратных скобок. Фигурные скобки предложил Виет (1593). Всё же большинство математиков тогда предпочитали вместо скобок надчёркивать выделяемое выражение. В общем употреблении скобки ввёл Лейбниц.

8. Знаки угла и перпендикулярности

Символы «угол» и «перпендикулярно» придумал в 1634 году французский математик Пьер Эригон. Символ перпендикулярности у него был перевёрнут, напоминая букву T. Современную форму символ угла придал ему Уильям Отред.

9. Знак параллельности

Символ «параллельности» известен с античных времён. Сначала символ был похож на нынешний знак равенства, но с появлением

последнего, во избежание путаницы, символ был повернут вертикально.

10. Число π

Общепринятое обозначение числа, равного отношению длины окружности к ее диаметру (3,1415926535...), впервые образовал Уильям Джонс в 1706 году, взяв первую букву греческих слов окружность и периметр, то есть длина окружности. Это сокращение понравилось Эйлеру, труды которого закрепили обозначение окончательно.

11. Синус и косинус

Sinus с латинского - пазуха, впадина. Далеко в тригонометрии продвинулись индийские математики в районе 5 века. самого слова "тригонометрия" не было. То, что мы сейчас называем синусом, примерно соответствует тому, что индусы называли ардха-джия, в переводе - полутетива (т.е. полухорда). Для краткости называли просто - джия (тетива). Когда арабы переводили работы индусов с санскрита, они не стали переводить "тетиву" на арабский, а просто транскрибировали слово арабскими буквами. Получилась джиба. Но так как в слоговой арабской письменности краткие гласные не обозначаются, то реально остается дж-б, что похоже на другое арабское слово - джайб (впадина, пазуха). Когда Герард Кремонский в 12 веке переводил арабов на латынь, он перевел это слово как *sinus*, что по-латыни также означает пазуху, углубление.

Косинус появился автоматически, т.к. индусы называли его коти-джия, или сокращено ко-джия. Коти - изогнутый конец лука на санскрите. Современные краткие обозначения и введены Уильямом Отредом и закреплены в трудах Эйлера.

Обозначения тангенса/котангенса имеют намного более позднее происхождение (английское слово *tangent* происходит от латинского *tangere* - касаться). И даже до сих пор нет унифицированного обозначения - в одних странах чаще используется обозначение *tan*, в других – *tg*.

Математическая наука необходима для цивилизованного общества. Математика содержится во всех науках. Благодаря математическим открытиям прошлого, ученые создают новые технологии. Сохранившиеся открытия дают возможность решать сложные математически задачи. Благодаря математике Архимед, Платон, Ньютон открыли физические законы. Но математический язык не теряется среди физических формул. Наоборот, эти формулы нельзя написать без знания математики. Благодаря истории

сохраняются знания и факты для будущих поколений. Дальнейшее изучение математики необходимо для новых открытий.

ЛИТЕРАТУРА

1. Википедия – свободная энциклопедия [Электронный ресурс].– Режим доступа: <https://wikipedia.org>.– Дата обращения: 11.04.2017

2. NewConcepts [Электронный ресурс].– Режим доступа: <http://newconcepts.club>.– Дата обращения: 12.04.2017

История математических обозначений [Электронный ресурс].– Режим доступа: <http://dic.academic.ru>.– Дата обращения: 13.04.2017

УДК 378.6:519.83

Студ. Д.И.Синюк

Науч. рук. доц. О.Н.Пыжкова

(кафедра высшей математики, БГТУ)

ОБ ОПЫТЕ РЕАЛИЗАЦИИ ИГРЫ ГАРВАРД В БГТУ

Математическая теория игр берёт своё начало из неоклассической экономики. Впервые математические аспекты и приложения теории были изложены в классической книге 1944 года Джона фон Неймана и Оскара Моргенштерна «Теория игр и экономическое поведение». Этот труд был сразу же провозглашен одним из величайших научных открытий века. Авторы предложили систематический подход к пониманию того, как ведут себя игроки в ситуациях, когда результат каждого зависит от действия всех остальных. В этой книге предпринята попытка построения системы аксиом теории игр, игр двух лиц с нулевой суммой, игр n лиц с нулевой суммой [1].

Теория игр получила окончательное признание в 1994 году, когда Джон Нэш, Джон Харшань и Рейнхард Зелтен получили Нобелевскую премию в области экономики за «новаторский анализ равновесия в теории игр с противоположными интересами».

Просматривая в интернете видео обзор лекции по теме «Теория игр» Алексея Владимировича Савватеева я увидел одну некооперативную игру «Гарвард», которая меня заинтересовала. Условие данной игры, следующее: пусть мы имеем множество игроков $N = (x_1, x_2, x_3, \dots, x_n)$, каждый из которых должен выбрать одно целое число из промежутка $[1...100]$, таким образом, чтобы выбранное игроком число было как можно ближе к половине среднего арифметического из всех чисел названными игроками. Пусть среднее

арифметическое C_s , тогда половина среднего арифметического $C_{p.s.}$ и вычисляется по формуле [2]:

$$C_{p.s.} = \frac{x_{1c} + x_{2c} + x_{3c} \dots + x_{nc}}{2n},$$

где $x_{1c}, x_{2c}, x_{3c}, \dots, x_{nc}$ это числа, выбранные игроками $x_1, x_2, x_3, \dots, x_n$ количество которых n человек.

Под описание к данной игре было записано следующее: «В жизни подобный результат вы практически никогда не получите».

В своей работе я решил выяснить какие результаты в этой игре обычно удаётся получить и с чем этот результат коррелирует, что является важной частью теории игр.

Среднее значение может быть любым от 1 до 100. Следовательно, среднее $C_{p.s.}$ пополам тоже может быть в каком-то диапазоне, который будет занимать половину этих чисел. И соответственно, априори нельзя сказать, что какое-то число будет ближе к этому среднему. Чтобы это сказать нужно немного подумать. Давайте рассмотрим самое большое возможное значение этого выражения. Предположим, что все участники игры назвали число 100, тогда половина $C_{p.s.}$ среднего принимает значение 50. Значит, если игроки выбрали произвольные стратегии, то есть любые числа от 1 до 100, то значение $C_{p.s.}$ не превосходит 50. Из этого можно сделать вывод, что число 50 — это максимальное число, которое, имеет смысл называть. Потому что назвать 51, 52 и так далее вплоть до 100 — любая из этих стратегий будет заведомо больше $C_{p.s.}$, т.е. числа 50, а, следовательно, проигрышная.

Итак, мы будем вычеркивать слабодоминируемые стратегии (51,52,...,100) и согласимся, что игроки будут использовать только оставшиеся (49,48,...,1). Нам в данном случае придется признать, что в этих условиях среднее $C_{p.s.}$ пополам уже не превосходит число 25. То есть если все игроки поняли, что (51,52,...,100) применять не надо, то тогда самое максимальное число, которое будет написано у кого-либо, будет число 50. Тогда C_s — не больше 50, а $C_{p.s.}$ — не больше 25. В этом случае, целый ряд новых стратегий оказался слабодоминируемыми посредством стратегии 25. Применим постулат снова, итеративное применение того же самого постулата, и вычеркнем эти стратегии [3].

После очередного вычеркивания каждый игрок ограничится уже выбором (1,2,...,25). В этом случае C_s будет уже 12,5 (если ответы

будут равномерно распределены), а $C_{p.s.}$ соответственно 6,25. Если все игроки достаточно умны, то они продолжают этот процесс, который называется итеративное исключение доминируемых стратегий. И, в конце концов, мы приходим к единственному возможному исходу: а именно все стратегические множества оказываются одноэлементными и содержат только стратегию 1. Тем самым, исход игры, который мы предсказываем, если мы пользуемся тем постулатом об удалении слабодоминируемых стратегий, будет равен 1, и у каждого игрока будет написано число 1, и все будут одинаково близки к $C_{p.s.}$, равному 0,5. Классическая теория игр прогнозирует здесь такой исход. Однако в реальности мы никогда число 1 не получим [3].

Для проверки утверждения, что число 1 никогда не получится в ответе, я провел эту игру несколько раз, сначала в одной группе (27 человек) для отработки методики, а затем в нескольких группах одного потока двух различных факультетов. Чтобы все (студенты первого курса) игроки поняли правила я дважды повторил условие, написал на доске формулу вычисления половины среднего $C_{p.s.}$ и привел простой пример этой игры (для двух человек), а чтобы их ещё заинтересовать я пообещал дать победителям приз. В эксперименте приняло участие 148 человек (факультет ТОВ 96 человек и факультет ЛХ 52 человека). А также мне помогали 5 волонтеров, которые оперативно раздали и собрали листки с ответами.

Я выдвинул гипотезу, что чем ниже число, которое выберет игрок, тем выше сумма баллов на ЦТ и средний балл в школе (по сумме этих баллов в белорусских вузах проводится конкурс).

Результаты исследования:

1) факультет технология органических веществ: $C_s = 37.52$, $C_{p.s.} = 18.76$, балл – 278,6;

2) лесохозяйственный факультет: $C_s = 44.37$, $C_{p.s.} = 22.17$, балл – 177.

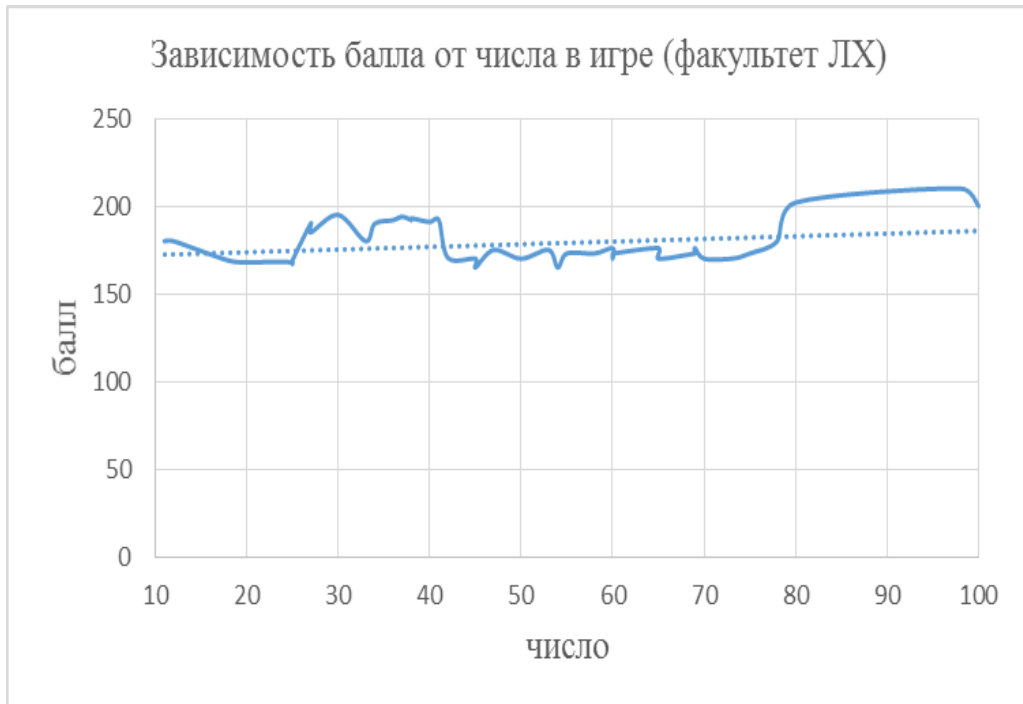


Рисунок 1 - Зависимость балла от числа в игре (ЛХ)



Рисунок 2 - Зависимость балла от числа в игре (ТОВ)

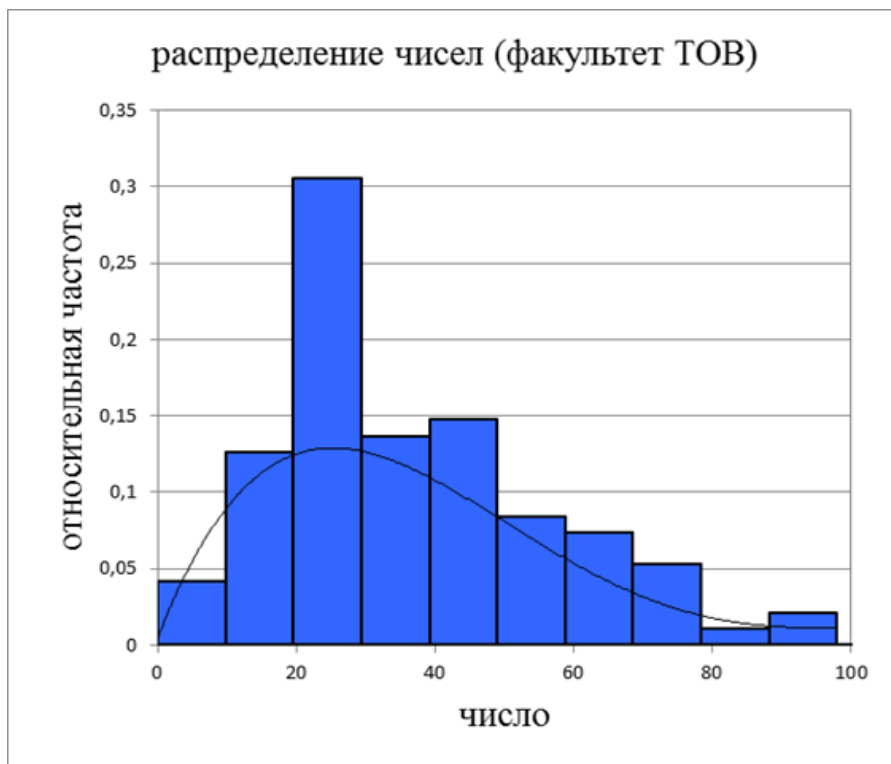


Рисунок 11 - Логнормальное распределение относительных частот чисел (ТОВ)

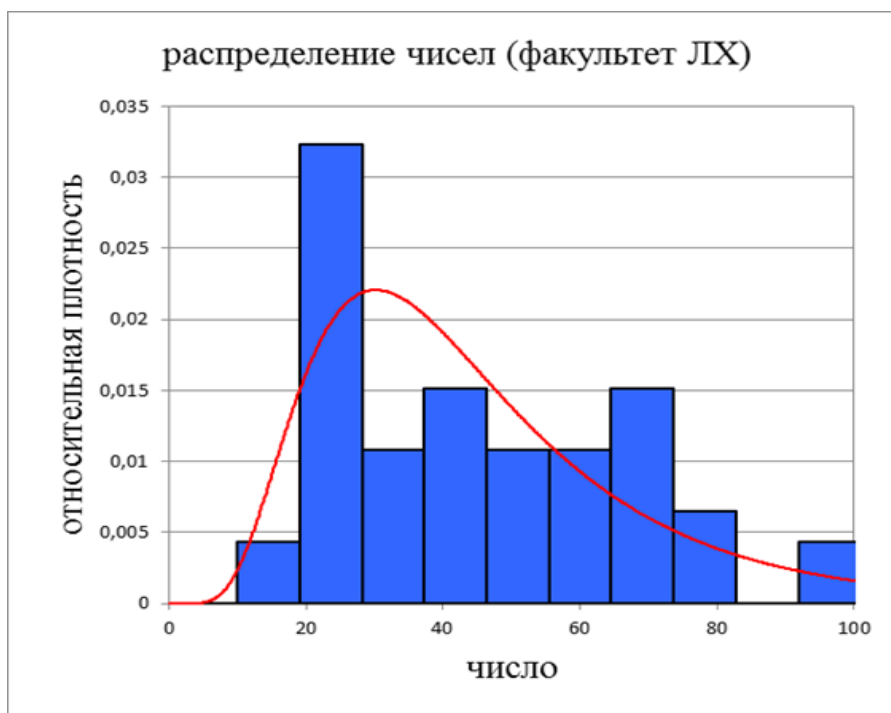


Рисунок 4 - Логнормальное распределение относительных частот чисел (ЛХ)

Почему же мы не получили результат 1? Причин здесь много. Перечислю основные из них:

во-первых, около трети участников не поняли условие игры или просто не захотели играть и писали числа наугад, не подумав (это видно из графиков);

во-вторых, часть студентов плохо продумала решение, так как даже при минимальном анализе становится ясно, что числа выше 50 (и даже 25) нет смысла называть.

Но более важная причина в том, что не все верят, что остальные будут достаточно рациональны.

Еще более важный результат который можно получить из этих графиков, это отсутствие очень сильной зависимости между суммой баллов при поступлении и числом в игре. Видимо результат в игре больше связан со специальностью, умением стратегически мыслить, и престижем, чем баллом при поступлении.

Подобные игры могут быть использованы в качестве оценки качества образовательного процесса в учебных заведениях и в виде вступительного испытания. Однако следует использовать не одну игру, а несколько подобных, чтобы исключить риск статистической погрешности. Также такие игры как Гарвард будут хорошо себя проявлять в виде практического задания на занятиях по теории игр.

А теперь почему эта игра называется Гарвард? Значит, наблюдения в Америке состоят в том, что в престижных университетах типа Гарварда, это число было значительно меньше, чем в других. В Гарварде $C_{p.s.}$ значение колебалось то в районе 7–8. Если игра проводилась в аудитории не очень интеллектуальной, то C_s , или $C_{p.s.}$, оказывалось больше, чем в аудиториях, где учатся более сильные студенты. И стали считать, что это своего рода индекс интеллекта данного университета. В Независимом университете результат данной игры дал $C_{p.s.}$ равным 4. А в Белорусском государственном технологическом университете на факультете ТОВ 1 курса значение $C_{p.s.}$ оказалось равным 19, а на факультете ЛХ $C_{p.s.}$ – 22.

ЛИТЕРАТУРА

1. Петросян Л. А., Зенкевич Н.А., Семина Е.А. Теория игр: Учеб. пособие для ун-тов. — М.: Высш. шк., Книжный дом «Университет», 1998. — С. 304.

2. <https://www.coursera.org/learn/gametheory/lecture/v3WYR/ighragarvard-dominiruiemyie-stratieghii>

3. Захаров, А. В. Теория игр в общественных науках: учебник для вузов / А. В. Захаров ; Нац. исслед. ун-т «Высшая школа экономики». — М. :Изд. Дом Высшей школы экономики, 2015. — (Учебники высшей школы экономики). — С. 304.

УДК 378.6:519.83

Студ. Н.Ю. Савось

Науч. рук. доц. И. К. Асмыкович
(кафедра высшей математики, БГТУ)**ПРОБЛЕМА КЛАССОВ P И NP**

Вопрос о равенстве классов сложности P и NP (в русских источниках также известный как проблема перебора) — это одна из центральных открытых проблем теории алгоритмов уже более трёх десятилетий (проблема института Клея, за решение которой дают вознаграждение). Если на него будет дан утвердительный ответ, это будет означать, что теоретически возможно решать многие сложные задачи существенно быстрее, чем сейчас.

P класс

В теории алгоритмов классом P (от англ. polynomial) называют множество задач, для которых существуют «быстрые» алгоритмы решения (время работы которых полиномиально зависит от размера входных данных).

Формальное определение

Алгоритм отождествляется с детерминированной машиной Тьюринга, которая вычисляет ответ по данному на входную ленту слову из входного алфавита Σ . Временем работы алгоритма $T_m(x)$ при фиксированном входном слове x называется количество рабочих тактов машины Тьюринга от начала до остановки машины. Сложностью функции $f: \Sigma^* \rightarrow \Sigma^*$, вычисляемой некоторой машиной Тьюринга, называется функция $C: N \rightarrow N$, зависящая от длины входного слова и равная максимуму времени работы машины по всем входным словам фиксированной длины

$$C_M(n) = \max_{x: |x|=n} T_M(x).$$

Если для функции f существует машина Тьюринга M такая, что $C_M(n) < n^c$ для некоторого числа c и достаточно больших n , то говорят, что она принадлежит классу P, или полиномиальна по времени.

Задачи, принадлежащие классу P

Примерами задач из класса P являются целочисленное сложение, умножение, деление, взятие остатка от деления, умножения матриц, выяснение связности графов, сортировка множества из n чисел, нахождение эйлерова цикла на графе из m рёбер, обнаружение в тексте длиной n некоторого слова, построение покрывающего дерева

минимальной стоимости, линейное программирование и некоторые другие.

NP класс

В теории алгоритмов классом NP (от англ. non-deterministic polynomial) называют множество проблем разрешимости, решение которых возможно проверить на машине Тьюринга за время, не превосходящее полинома от размера входных данных, при наличии некоторых дополнительных сведений (так называемого сертификата решения).

Эквивалентно класс NP можно определить как содержащий задачи, которые можно за полиномиальное время решить на недетерминированной машине Тьюринга.

Задачи, имеющие полиномиальные по времени алгоритмы решения, можно решать с помощью компьютера значительно быстрее, чем путём прямого перебора, время которого экспоненциально. Это обуславливает практическое значение проблемы о равенстве классов P и NP.

Определения

Класс сложности NP определяется для множества языков, то есть множеств слов над конечным алфавитом. Язык L называется принадлежащим классу NP, если существуют двуместный предикат $R(x,y)$ из класса P (то есть вычислимый за полиномиальное время) и константа $c > 0$ такие, что для всякого слова x условие «x принадлежит L» равносильно условию «найдётся у длины меньше $|x|^c$ такой, что верно $R(x,y)$ » (где $|x|$ — длина слова x). Слово y называется сертификатом принадлежности x языку L. Таким образом, если у нас есть слово, принадлежащее языку, и ещё одно слово-свидетель ограниченной длины (которое бывает трудно найти), то мы быстро сможем удостовериться в том, что x действительно принадлежит L.

Эквивалентное определение можно получить, используя понятие недетерминированной машины Тьюринга (то есть такой машины Тьюринга, у программы которой могут существовать разные строки с одинаковой левой частью). Если машина встретила «развилку», то есть неоднозначность в программе, то дальше возможны разные варианты вычисления. Предикат $R(x)$, который представляет данная недетерминированная машина Тьюринга, считается равным единице, если существует хоть один вариант вычисления, возвращающий 1, и нулю, если все варианты возвращают 0. Если длина вычисления, дающего 1, не превосходит некоторого многочлена от длины x, то предикат называется принадлежащим классу NP. Если у языка существует распознающий его предикат из

класса NP, то язык называется принадлежащим классу NP. Это определение эквивалентно приведённому выше: в качестве свидетеля можно взять номера нужных веток при развилках в вычислении. Так как для x принадлежащему языку длина всего пути вычисления не превосходит многочлена от длины x , то и длина свидетеля также будет ограничена многочленом от длины x .

Примеры задач принадлежащих NP классу

Можно привести много задач, про которые на сегодняшний день неизвестно, принадлежат ли они P, но известно, что они принадлежат NP. Среди них:

- Определение наличия в графе гамильтонова цикла. Сертификат — последовательность вершин, образующих гамильтонов цикл.
- Неоптимизационный вариант задачи о коммивояжёре (существует ли маршрут не длиннее, чем заданное значение k) — расширенный и более приближенный к реальности вариант предыдущей задачи. Сертификат - такой маршрут.
- Существование целочисленного решения у заданной системы линейных неравенств. Сертификат — решение.
- Кратчайшее решение «пятнашек» размера $n \times n$

Среди всех задач класса NP можно выделить «самые сложные» — NP-полные задачи. Если удастся решить любую из них за полиномиальное время, то все задачи класса NP также можно будет решить за полиномиальное время.

NP-полная задача — в теории алгоритмов задача с ответом «да» или «нет» из класса NP, к которой можно свести любую другую задачу из этого класса за полиномиальное время (то есть при помощи операций, число которых не превышает некоторого полинома в зависимости от размера исходных данных). Таким образом, NP-полные задачи образуют в некотором смысле подмножество «типовых» задач в классе NP: если для какой-то из них найден «полиномиально быстрый» алгоритм решения, то и любая другая задача из класса NP может быть решена так же «быстро».

Равенство $P=NP$ может означать, что задачи, решение которых раньше считалось очень сложным, теперь решаются за полиномиальное время.

Любая криптосистема с открытым ключом базируется на предположении существования односторонних функций и/или крайней затратности решения некоторой задачи (например, для алгоритма RSA это разложение на множители очень больших чисел). В зависимости от того будет ли доказана принадлежность задач NP класса к P классу будет зависеть безопасность всех систем базирующихся на криптосистемах и задачах которые принадлежат к NP классу. То есть текущие системы безопасности базирующиеся на NP задачах построены на вере в неупрощаемость этих задач. Но это является лишь неполным условием. Иными словами, в NP-задачи заложена мера сложности «в худшем случае», но для стойкости криптографической системы необходимо, что бы задача была сложной «почти всюду». Таким образом, нам видно что для криптографической стойкости необходимо существенно более сильное предположение, чем $P \neq NP$ (хоть и в большинстве случаев оно является достаточным, кроме случаев с криптосистемами с открытым ключом). А именно, предположение о существовании односторонних функций.

ЛИТЕРАТУРА

1. Введение в криптографию, В.В. Яценко 2012.- С. 348.
2. Википедия, свободная энциклопедия(<https://www.wikipedia.org>)
3. Д. Хопкрофт, Р. Мотвани, Д. Ульман., Введение в теорию машин Тьюринга // Введение в теорию автоматов, языков и вычислений 2002. – С. 528.

УДК 519.83

Студ. П.С. Шенец

Науч. рук. асс. Т.Г. Шагова

(кафедра высшей математики, БГТУ)

ИГРА «НИМ»

Ним – это конечная игра с полной информацией, которая является фундаментом математической теории комбинаторных игр. Эта древняя китайская игра пришла в Европу только в 16 веке, а известное нам название, одновременно с доказательством того, что у неё есть выигрышная стратегия, получила только в 20 веке. И то, и другое сделал математик Чарльз Бутон.

В 30-х годах XX в. независимо друг от друга два математика — Р. Шпраг и П.М. Гранди — разработали теорию, описывающую равноправные игры. И Ним имеет фундаментальное значение для этой теории, так как в ней утверждается, что любая равноправная игра двух игроков эквивалентна обычному Ниму.

Рассмотрим правила игры. Два игрока поочередно берут предметы из кучек. Число кучек и число предметов может быть произвольным, и выкладываются они заранее, до начала игры. Взять разрешается любое число предметов из любой кучки, даже всю кучку целиком, но хотя бы один предмет взять обязательно, и брать предметы можно только из одной кучки. Игрок, взявший последний предмет, выигрывает игру.

Какова же выигрышная стратегия этой игры? Рассмотрим все возможные случаи.

Если у нас всего одна кучка, то стратегия проста: забрать всю кучу и выиграть.

Если их две, то стратегия следующая: одним ходом нужно уравнивать количество камушков в кучках, а потом, после хода противника, просто копировать его ход на другой кучке. И тогда в один момент игра сведётся к одной кучке.

Если количество кучек больше 2, то описать выигрышную стратегию не так просто. Для этого нам понадобится двоичная система счисления. Из теоремы, представленной Ч. Бутоном, текущий игрок имеет выигрышную стратегию тогда и только тогда, когда XOR-сумма размеров кучек отлична от нуля. В противном случае текущий игрок находится в проигрышном состоянии. Из доказательства теоремы можно получить следующий алгоритм выигрышной стратегии. Рассмотрим его на конкретном примере.

Пусть у нас есть 3 кучки, в них 5, 6 и 7 камушков соответственно.

1. Перевести размеры кучек в двоичную систему счисления.

Таким образом, $5_{10} = 101_2$, $6_{10} = 110_2$, $7_{10} = 111_2$.

2. Найти XOR – сумму получившихся чисел.

XOR – это бинарная операция, результат выполнения которой будет равен 1 тогда, когда количество аргументов, равных 1, составляющих текущий набор, — нечетное, в противном случае, результата равен 0. (рис.1)

Применив эту операцию к нашим числам, получим XOR-сумму(Ним-сумму) 100 (рис.2).

3. Найти первый ненулевой бит в Ним-сумме. (рис.3)

4. Найти число с таким же ненулевым битом. (рис.4)

Выбираем любое из них и применяем к нему XOR операцию с полученной ранее Ним-суммой. Полученное число – это то количество камушков, которое мы должны оставить в той кучке, с которой только что проводили вычисления. Заменяя это число, мы должны получить нулевую Ним-сумму.

A	B	XOR				
			101	101	Нужный бит	101
0	0	0	110	110		110
0	1	1	<u>111</u>	<u>111</u>		<u>111</u>
1	0	1	100	100		100
1	1	0				

Рисунок 1 Рисунок2
Рисунок 3 Рисунок 4

Отсюда мы видим, что из любого числа с таким же ненулевым битом мы можем сделать нулевую Ним-сумму.

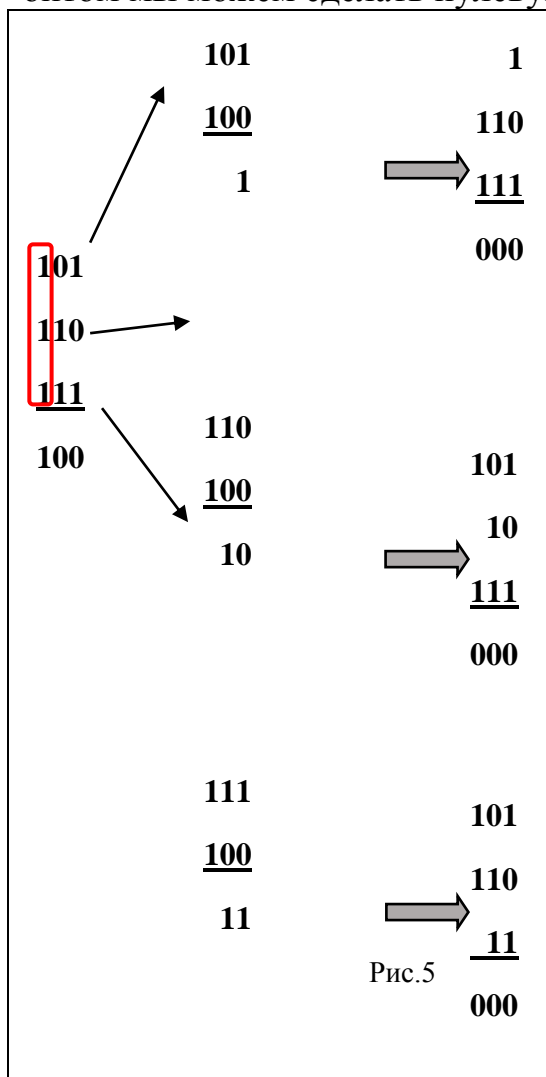


Рис.5

Далее противник, находясь в проигрышной позиции, делает абсолютно любой ход, так как гарантированно он переведет игру в состояние с ненулевой Ним-суммой. После его хода мы снова повторяем свои действия. В конечном итоге игра сведётся к случаю с двумя кучками, а затем с одной.

Так же мы должны отметить, что не всегда выгодно ходить первым. Если на начало игры Ним-сумма уже равна 0, то стоит предложить начать игру сопернику, тогда вы, заранее зная выигрышную стратегию, обеспечиваете себе победу, либо придётся ждать оплошности соперника.

Игра — это высшая форма исследования. Наша жизнь — это тоже игра. У которой пока что нет выигрышной стратегии, но кто его знает, может, и эта игра в конечном

итоге сведётся к Ниму.

ЛИТЕРАТУРА

1. Гашков, С. Б. Системы счисления и их применение. М.: МЦНМО, 2004. 52 с.
2. Шень, А. Игры и стратегии с точки зрения математики. М.: МЦНМО, 2008. 40 с.
3. Теория Шпрага-Гранди. Нем. [Электронный ресурс] / MAximal. – 2014. / Режим доступа: http://e-maxx.ru/algo/sprague_grundy#2. Дата доступа: 25.03.2017.
4. Фролов И.С. Введение в теорию комбинаторных игр. 2012. 202 с.

УДК 004.021

Студ. А.Н. Зайцев

Науч. рук. асс. Т.Г. Шагова

(кафедра высшей математики, БГТУ)

МЕТОДЫ СГЛАЖИВАНИЯ ИЗОБРАЖЕНИЙ

В современном мире огромная часть информации представлена в виде изображений. И часто бывает, что на изображениях имеются различного вида искажения. Одним из видов искажения является эффект «зубчатости». Мало того, что данный эффект портит впечатление от изображения из-за неровных краёв, зубчатость крайне негативно сказывается, например, на оптическом распознавании символов. Именно поэтому в данном случае следует применять алгоритмы сглаживания.

Для сглаживания применяют следующие сглаживающие фильтры: фильтр Гаусса, медианный фильтр, обобщённый медианный фильтр, билатеральный фильтр, нелокальный фильтр, морфологический фильтр.

Самые простые фильтры - основанные на матрицах свёртки. Матрица свёртки – это матрица коэффициентов, на которую «умножается» значение пикселей изображения для получения требуемого результата. Пример свёрточного фильтра следующий:

$$\begin{array}{l}
 \text{Входное изображение} \qquad \qquad \qquad \text{Матрица} \\
 \begin{pmatrix} 12 & 14 & 41 \\ 43 & 84 & 24 \\ 2 & 1 & 43 \end{pmatrix} \times \begin{pmatrix} 0,5 & 0,75 & 0,5 \\ 0,75 & 1,0 & 0,75 \\ 0,5 & 0,75 & 0,5 \end{pmatrix} = \\
 = \begin{pmatrix} 12 * 0,5 + 14 * 0,75 + 41 * 0,5 + \\ 43 * 0,75 + 84 * 1,0 + 24 * 0,75 + \\ 2 * 0,5 + 1 * 0,75 + 43 * 0,5 \end{pmatrix} \times \frac{1}{\text{div}} = 31,41667,
 \end{array}$$

где $\text{div} = 6$.

Фильтр Гаусса основывается на том, что матрица свёртки заполняется по закону распределения Гаусса. Стоит отметить, что результат работы фильтра будет напрямую зависеть от размера ядра фильтра, то есть матрицы свертки: чем больше размер ядра, тем качественнее будет размытие, но тем дольше работает фильтр. Рассмотрим пример матрицы свёртки, заполненной по закону распределения Гаусса (табл. 1):

0,000789	0,006581	0,013347	0,006581	0,000789
0,006581	0,054901	0,111345	0,054901	0,006581
0,013347	0,111345	0,225821	0,111345	0,013347
0,006581	0,054901	0,111345	0,054901	0,006581
0,000789	0,006581	0,013347	0,006581	0,000789

Таблица 3

Асимптотика фильтра Гаусса: $O(h_i \cdot w_i \cdot n \cdot n)$, где h_i , w_i – размеры изображения, n – размер матрицы (ядра фильтра). Наиболее часто используются ядра размером 3×3 , 5×5 , 9×9 , 11×11 . Применение ядер с более крупным размером не даёт значительного прироста качества, но приводит к существенному увеличению ресурсоёмкости вычислений. Фильтр Гаусса может быть представлен в следующем виде:

$$G_0(x, y) = A e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2} - \frac{(y-\mu_y)^2}{2\sigma_y^2}},$$

где μ – это математическое ожидание, σ^2 является дисперсией (по каждой из переменной x и y).

Медианный фильтр.

Медианный фильтр обычно используется для уменьшения шума или «сглаживания» изображения. Фильтр работает с матрицами различного размера, но в отличие от матрицы свёртки, размер матрицы влияет только на количество рассматриваемых пикселей.

Алгоритм медианного фильтра следующий:

Для текущего пикселя, пиксели, которые «падают» в матрицу, сортируются, и выбирается среднее значение из отсортированного массива. Это значение и является выходным для текущего пикселя. Особенно хорошо медианный фильтр показывает себя на импульсных шумах: если в целом у нас изображение качественное, но только на некоторых пикселях наше изображение

искажено, то медианный фильтр просто усреднит данный пиксель согласно его соседям. Асимптотика медианного фильтра: $O(h_i * w_i * n * n)$, где h_i , w_i – размеры изображения, n – размер матрицы (ядра фильтра). Рекомендуемый размер ядер такой же, как и у фильтра Гаусса.

Билатеральный фильтр – нелинейный фильтр, выполняющий пространственное усреднение в пределах своей маски. Идея крайне похожа на предыдущие фильтры, но билатеральный фильтр также учитывает разность весов пикселей. Он может быть выбран в качестве эффективной техники удаления шума и ряда других искажающих факторов. Важным вопросом при работе с билатеральным фильтром является выбор его параметров, которые влияют на эффективность фильтрации. Пример работы билатерального фильтра можно видеть на (Рис. 1 и Рис. 2):



Рисунок 13



Рисунок 12

На практике в первую очередь выбор фильтра зависит от требований к ресурсоёмкости вычислений. Если требуется простой фильтр, который будет работать быстро, пусть и с меньшим качеством, то выбирается фильтра Гаусса или медианный фильтр с малым размером ядра. Шум в основном пропадает и обеспечивается приемлемое качество изображения. Если же вычислительная сложность не имеет большого значения (например, обработка фотографий в профессиональных программах), то выбирают в основном различные модификации билатеральных фильтров с большим размером ядра или даже нелокальные билатеральные фильтры.

ЛИТЕРАТУРА

1. Матричные фильтры обработки изображений [Электронный ресурс] / Habrahabr. — 2017. / Режим доступа: <https://habrahabr.ru/post/142818/>. — Дата доступа: 25.03.2017.
2. Median filter [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Median_filter. — Дата доступа: 25.03.2017.
3. Image analysis [Электронный ресурс] / Wikipedia. — 2017. / Режим доступа: https://en.wikipedia.org/wiki/Image_analysis. — Дата доступа: 25.03.2017.
4. Bilateral Filtering for Gray and Color Images [Электронный ресурс] / homepages.inf.ed.ac.uk. — 2017. / Режим доступа: [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUC N1/Bilateral_Filtering.html](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUC%20N1/Bilateral_Filtering.html). — Дата доступа: 25.03.2017.

УДК 514.74

Студ. Д.А.Байгазин
Науч. рук. доц. И.М.Борковская
(кафедра высшей математики, БГТУ)

ЗАМЕЧАТЕЛЬНЫЕ КРИВЫЕ

Человека всегда окружали линии: линия горизонта, береговая линия, линия изгиба реки и т.д. Кривые можно увидеть в разнообразных объектах нашего мира – от листьев и цветов, расположения семечек в подсолнухе до траекторий движения точек в природе и технике. Линии могут описываться в декартовой прямоугольной системе координат алгебраическими уравнениями первой, второй и некоторых высших степеней. Часто линии удобно задавать параметрическими уравнениями либо уравнениями в полярной системе координат. Аристотель говорил: «Математика выявляет порядок, симметрию и определенность, а это – важнейшие виды прекрасного». Глядя на изображения некоторых кривых, их симметрию и красоту, убеждаешься в том, что математика раскрывает красоту окружающего нас мира. Остановимся на некоторых замечательных кривых, которые в силу своих свойств получили известность. Итак, начнем.

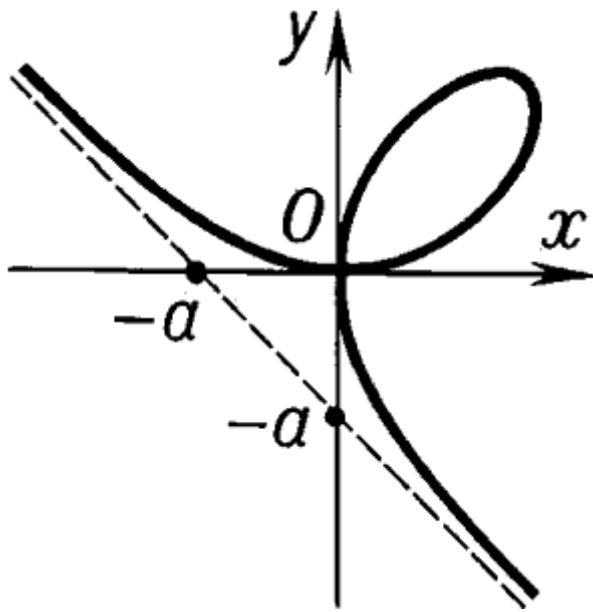


Рисунок 1- Декартов лист

Декартов лист – плоская линия, уравнение которой в декартовой системе координат имеет вид: $x^3 + y^3 - 3axy = 0$.
 Параметрические уравнения линии:

$$x = \frac{3at}{1+t^3},$$

$$y = \frac{3at^2}{1+t^3}.$$

Полярное уравнение линии:

$$r = \frac{a \sin \varphi \cos \varphi}{\sin^3 \varphi + \cos^3 \varphi}.$$

Прямая $x + y + a = 0$ является асимптотой декартова листа.

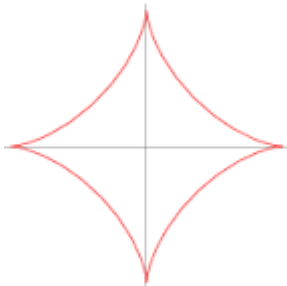


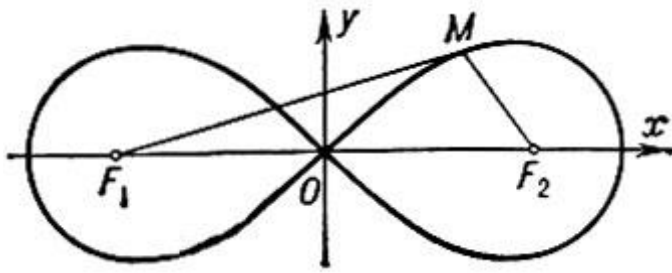
Рисунок 2 - Астроида

Астроида – от греческого «звездообразная» – это траектория точки, лежащей на окружности, катящейся по внутренней стороне другой окружности, радиус которой в 4 раза больше. Линия является частным случаем гипоциклоиды.

Уравнение астроида в декартовой системе координат: $x^{\frac{2}{3}} + y^{\frac{2}{3}} = a^{\frac{2}{3}}$.

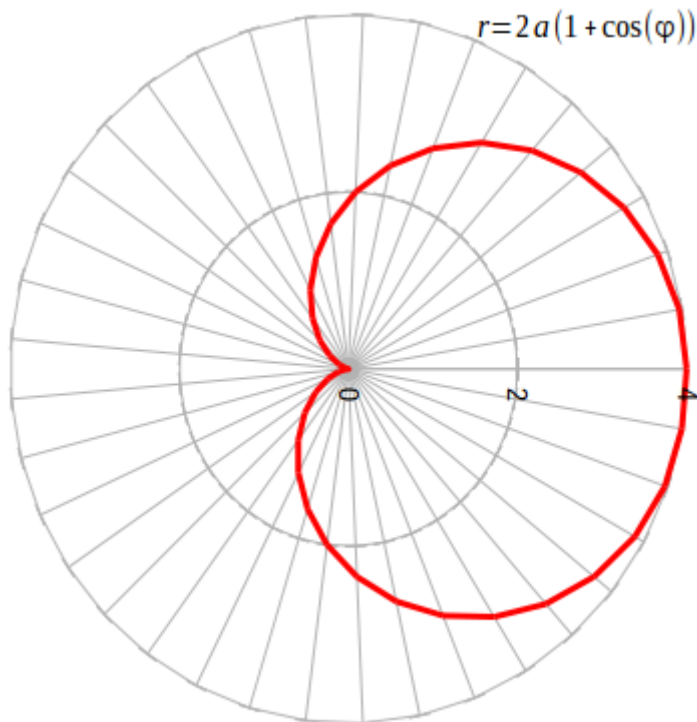
Параметрические уравнения линии:
 $x = a \cos^3 t,$
 $y = a \sin^3 t.$

Лемниската Бернулли – кривая, описываемая точкой на плоскости так, что остается неизменным произведение расстояний от этой точки до двух определенных точек той же плоскости. Впервые была рассмотрена Якобом Бернулли в 1694 году. Лемниската по-гречески означает «ленточная». Уравнение в полярных координатах: $r^2 = 2a^2 \cos 2\varphi$.



Лемниската Бернулли используется в качестве переходной линии на закруглениях малого радиуса (например, на трамвайных путях).

Рисунок 3 - Лемниската Бернулли



Кардиоида (от греческого «сердце») получила свое название из-за схожести своих очертаний со стилизованным изображением сердца.

Кардиоида описывается точкой окружности, катящейся по окружности с таким же радиусом.

Является эпициклоидой. В технике эта прямая применяется для устройства кулачковых механизмов.

Рисунок 4 - Кардиоида

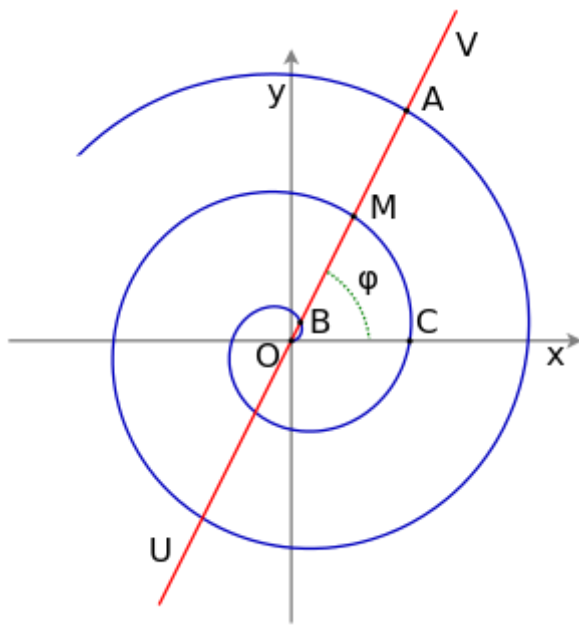


Рисунок 5 - Спираль Архимеда

Архимедова спираль – плоская кривая, траектория точки, движущейся из полюса с постоянной скоростью по лучу, вращающемуся около полюса с постоянной угловой скоростью. Уравнение линии: $r = a\varphi$.

Архимедова спираль относится к алгебраическим спиральям. Кривая названа по имени Архимеда (3 в до н.э.), который изучал ее свойства. Геометрическим свойством, характеризующим спираль Архимеда, является постоянство расстояний между витками, каждое из которых равно $r = 2\pi a$, где a – некоторое число.

Логарифмическая спираль $r = a^\varphi$, $a > 0$ – это кривая, пересекающая все радиус-векторы под одним и тем же углом. Полюс – асимптотическая точка. Линия широко используется в технике: по ней выполняются профили вращающихся ножей и фриз, зубчатых передач. По логарифмической спирали очерчены некоторые раковины, по дугам, близким к логарифмической спирали, расположены семечки в подсолнухе, чешуйки в шишках.

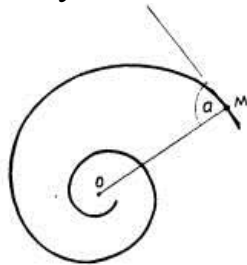


Рисунок 6 - Логарифмическая спираль

Эту линию можно было бы назвать по имени Декарта, так как впервые о ней говорится в одном из его писем (1638 г.). Однако подробное изучение ее свойств было проведено только полвека спустя Якобом Бернулли. На каменной плите, водруженной на могиле этого знаменитого математика, изображены витки логарифмической

Кривые, и те, которые рассмотрены выше, и не рассмотренные, отражают красоту окружающего нас мира и представляют собой интереснейший объект для изучения.

ЛИТЕРАТУРА

1. Райхмист, Р. Б. Графики функций: Справ. Пособие для вузов. – М.: Высшая школа, 1991. – 160 с.

2. Воднев, В. Т. Математический словарь высшей школы / В. Т. Воднев, А. Ф. Наумович, Н. Ф. Наумович. – Минск: «Вышэйшая школа», 1984. – 527 с.

3. Кудрявцев, Л. Д. Сборник задач по математическому анализу / Л. Д. Кудрявцев, А. Д. Кутасов, В. И. Чехлов, М. И. Шабунин. – М.: «Наука», 1984. – 592 с.

УДК 339.138

Студ. Д. Д. Тарасенко

Науч. рук. доц. И.Ф. Соловьева

(кафедра высшей математики, БГТУ)

КРАСОТА МНОГОГРАННИКОВ В МАТЕМАТИКЕ, ИСКУССТВЕ И АРХИТЕКТУРЕ

Трехмерные многогранники значительно отличаются от более привычных нам плоских фигур. Рассмотрим хорошо известный пример: на плоскости существует бесконечно много правильных многоугольников, в пространстве – всего пять правильных многогранников. И тот факт, что при переходе из двухмерного в трехмерном пространстве число вариантов уменьшается с бесконечности всего до пяти, настораживает.

Рассмотрим правильные пятиугольники: из них невозможно составить мозаику (их углы не стыкуются друг с другом), однако из двенадцати правильных пятиугольников в пространстве можно составить прекрасный многогранник – додекаэдр.

Формула Эйлера

Леонард Эйлер (1707-1783)

Эйлер был одним из величайших математиков всех времен. Он родился в Швейцарии, но большую часть жизни проработал в Берлинской и Петербургской академии наук. Ему принадлежит 886 трудов, полное собрание его сочинений насчитывает 87 томов. Особо важны его работы по алгебре, теории чисел, геометрии, математическому анализу, механике, астрономии и физике.

Что общего у куба, пирамиды, шестиугольной призмы, додекаэдра и ромбододекаэдра? Да, все эти фигуры многогранники, но их характеристики заметно отличаются. Существует удивительное соотношение, которое выполняется для всех этих фигур:

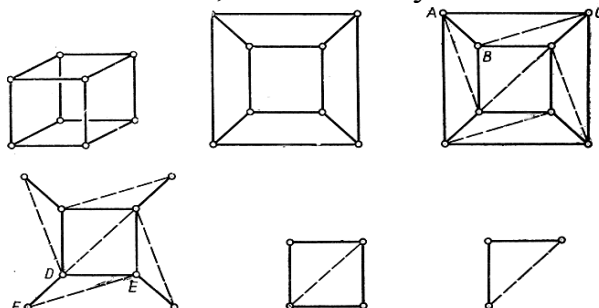


$$\text{Грани} + \text{Вершины} = \text{Ребра} + 2$$

Эта формула справедлива для всех выпуклых многогранников.

Рассмотрим выпуклый n -угольник с вершинами v_1, v_2, \dots, v_n и соедующими ребрами v_1v_2, \dots, v_nv_1 . Вне зависимости от длин сторон, величин углов, кривизны ребер и т.д. число ребер будет всегда равно числу вершин многоугольника.

Перейдем в трехмерное пространство и рассмотрим произвольный выпуклый многогранник, который имеет V вершин, P ребер и Γ граней. Если посмотреть на этот многогранник изнутри и спроецировать его на большую сферу, внутри которой он находится, то его линии и соответствующие вершины окажутся нанесены на эту сферу так, что его значения V , P и Γ останутся неизменными.



Многограннику также можно поставить в соответствие плоский граф, который будет иметь то же число ребер P , то же число вершин V и то же число граней Γ : $\Gamma - 1$ – число многоугольников, 1 – внешняя часть плоскости (грань, состоящая из части плоскости, ограничивающей плоскости, ограничивающей граф). Тогда, по индукции, видим, что при $\Gamma = 2$ получится единственный многоугольник и $V = P$, или, что аналогично, $\Gamma + V = P + 2$.

Если при $\Gamma = n$ число вершин = V , число ребер = P , и мы предположим (по индукции), что $n + V_n = P_n + 2$, тогда при $\Gamma = n + 1$ нужно заострить внимание на $(n + 1)$ -й грани. Когда число граней станет равным $n + 1$, к графу с n гранями добавится некоторое число вершин K и $K + 1$ ребро. Следовательно,

$$\begin{aligned} \Gamma + V_{n+1} &= n + 1 + V_n + K = (n + V_n) + (K + 1) = (P_n + 2) + (K + 1) \\ &= (P_n + K + 1) + 2 = P_{n+1} + 2. \end{aligned}$$

Получили знаменитую теорему: В любом выпуклом многограннике выполняется формула Эйлера: $\Gamma + V = P + 2$.

Эта теорема означает, что соотношение выполняется для любого выпуклого многогранника независимо от формы его граней, углов на гранях и углов между гранями, от длин ребер и т.д.

Формула, которая корректна для бесконечно большого числа разнообразных фигур, не может не привлекать внимание. Подобных формул, справедливых для столь непохожих фигур, практически не существует.

Формула Эйлера и Декарта

Рене Декарт (1596-1650)

Французский философ, математик, механик и физик, создатель аналитической геометрии и современной алгебраической символики, автор метода радикального сомнения в философии, механицизма в физике. Главный философско-математический труд Декарта, «Рассуждение о методе». В приложении «Геометрия» к этой книге излагались аналитическая геометрия, многочисленные результаты в алгебре и геометрии.



Рассмотрим выпуклый многогранник P . Декарта. Для каждой вершины v_i рассчитывается угловой дефект Δ_i , определяемый как разность между 2π и суммой углов, сходящихся в вершине V_i , которую мы обозначим через S . В произвольной вершине тетраэдра угловой дефект будет равен $2\pi - 3\frac{\pi}{3} = \pi$, в вершине куба – $2\pi - 3\frac{\pi}{2} = \frac{\pi}{2}$. Через $\Delta = \Delta_1 + \Delta_2 + \dots$ если P имеет n вершин V_1, V_2, \dots . Чему равно Δ ? Рене обнаружил, что Δ неизменно равняется 4π радианам. Отсюда следует справедливость соотношения: $\Delta = 2\pi * (\Gamma + B - P)$.

Обратим внимание, что если теорема Декарта верна, но $\Delta = 4\pi$ и, исходя из вышеизложенного, $\Gamma + B - P = 2$, то $\Delta = 4\pi$. Таким образом, Эйлера и Декарт независимо друг от друга доказали эквивалентные утверждения.

Представим в виде таблицы следующие закономерности вершин, ребер и граней для некоторых многогранников.

Закономерность Рене Декарта:
 $V - P + \Gamma = 2$.

Тела Платона	V	P	Γ
Тетраэдр	4	6	4
Куб	8	12	6
Октаэдр	6	12	8
Додекаэдр	20	30	12
Икосаэдр	12	30	20

В 1755 г. Л. Эйлер доказал, что это замечательное равенство справедливо для произвольного выпуклого многогранника.

$\chi = V - P + \Gamma$ – эйлерова характеристика многогранника.

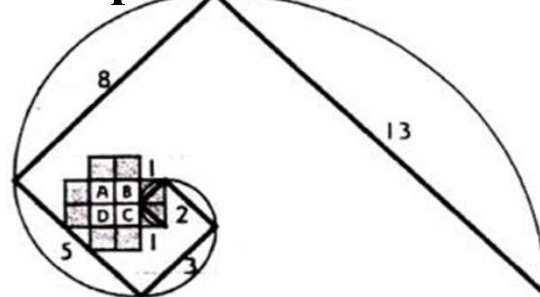
Золотое сечение и правильные многоугольники

Золотое сечение (золотая пропорция, деление в крайнем и среднем отношении, гармоническое деление) — соотношение двух величин b и a , $a > b$, когда справедливо $a/b = (a+b)/a$.

Золотое сечение, величина которого равна $\Phi = (1 + \frac{\sqrt{5}}{2}) = 1,618\dots$ – это отношение между диагональю правильного пятиугольника и его стороной. Это объясняет, почему золотому сечению подчиняются размеры икосаэдров и додекаэдров.

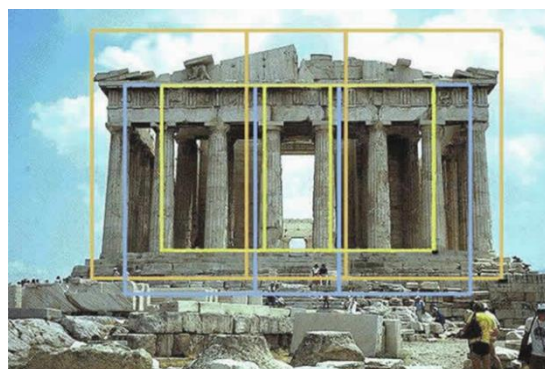
С древних времен золотое сечение считалось одним из красивейших соотношений. Знаменитая последовательность Фибоначчи, в свою очередь определяется так: два ее первых члена равны 1, а каждый последующий равен сумме двух предыдущих (1, 1, 2, 3, 5, 8, 13...). Отношение между соседними числами Фибоначчи стремится к числу, равному величине золотого сечения. Сегодня золотое сечение можно встретить во множестве объектов, например в строительстве домов, футбольных мечах, пачках для сигарет и банковских картах.

Спираль Фибоначчи



Приложение метода «Золотого сечения»

Понятие сущности «Золотого сечения» заключается в делении целого предмета на две части таким образом, что весь предмет так относится к большей своей части, как большая часть – к меньшей. «Золотое сечение» – это



Парфенон. Западный фасад(447-438 до н.э.)

иррациональное число, приблизительно равное 1,618.

Существует так же понятие о втором «Золотом сечении», которое вытекает из основного сечения, дает другое отношение 44:56. Такая пропорция обнаружена в архитектуре, а также имеет место при построении композиций изображений удлиненного горизонтального формата.

Принцип «Золотого сечения» – высшее проявление структурного и функционального совершенства целого и его частей в искусстве, науке, технике и природе.

Понятие о «Золотом сечении» впервые было введено Пифагором (VI в. до н.э.). Любой предмет, геометрическая фигура, вещь, построение которых соответствует «Золотому сечению», отличаются строгой пропорциональностью, и производят приятное зрительное впечатление.

Замечательными примерами «Золотого сечения» в живописи, архитектуре и скульптуре являются статуи Аполлона Бельведерского, статуя Зевса Олимпийского, статуя Афины, Парфенон, Собор Парижской Богоматери, «Черный квадрат» Малевича, «Джоконда» Леонардо да Винчи и т.д.

Многогранники в архитектуре

Антонио Гауди (1852-1926)

Известнейший архитектор XX века родом из Испании. Преодолев нелегкий путь от простого чертежника, он становится одним из самых дорогих, модных и востребованных зодчих своего времени.

Все богачи Барселоны мечтали владеть необыкновенной усадьбой, созданной Антонио Гауди. Ориентируясь на опыт неоготики, архитектор приходит к своему особому, ни на что не похожему стилю, который лишь приближенно можно отнести к модерну.

Последний и самый значимый для автора - проект Собора Саграда Фамилия, который он так и не заканчивает.



Храм Святого Семейства

Антонио Гауди познакомился с многогранниками, изучая архитектуру, начертательную геометрию и естественные науки, а также читая книгу Леруа, где они подробно описывались.

Соотношение всех архитектурных элементов храма Святого Семейства описываются делителями числа 12 (1:1, 1:4, 1:2, 3:4, 1:3, 2:3). Неудивительно, что при проектировании двенадцати колоколен храма Гауди выдвинул на первый план некоторые правильные многогранники, так как у куба и октаэдра двенадцать ребер, у додекаэдра – двенадцать граней, а у икосаэдра – двенадцать вершин. Кроме того, двенадцать колоколен оканчиваются изображениями епископских перстней, так как Гауди было прекрасно известно, что ювелиры при огранке бриллиантов и драгоценных камней всегда использовали как образец правильные многогранники.



Логично думать, что Гауди, любитель оригами, конструировал модели многогранников из бумаги. В его мастерской, а также в крипте храма Святого Семейства и соборе Пальма-де-Майорка можно увидеть модели многогранников, подвешенных к потолку.

Удивительным образом Гауди использовал многоугольники в колоннах нефа храма Святого семейства. Колонны – результат тонкой геометрической игры, состоящей в перемещении многоугольников и пересечении фигур. Два равных правильных многоугольника смещены вверх и повернуты в противоположных направлениях, образуя колонну, сечениями которой на разной высоте являются последовательности многоугольников со все большим числом сторон.

Многогранники в искусстве

Кубизм (фр. Cubisme) — модернистское направление в изобразительном искусстве, прежде всего в живописи, зародившееся в начале XX века во Франции и характеризующееся использованием подчеркнута геометризованных условных форм, стремлением «раздробить» реальные объекты на стереометрические примитивы.



I стадия: сезанновский кубизм (1907 - 1909) – выделение геометрических форм фигур и предметов, отделение формы от пространства и плоскости.

II стадия: аналитический кубизм (1909-1912) – дробление форм на грани и срезы, построение композиции при помощи коллажа из пересекающихся срезов и плоскостей, стирание граней между формой и пространством, визуальное взаимодействие формы и пространства.

III стадия: синтетический кубизм (1913 – 1914) – с помощью геометрических форм и их фрагментов конструируются новые объекты, которые обладают реальностью сами по себе. Коллажи создаются, в том числе, с помощью аппликаций, которые наиболее часто представляют собой фрагменты газетного листа, вклеенного в композицию.

По словам Холлингсворт, "изображение дробится на мелкие плоскости, объемность вообще не принимается в расчет, а объем существует как бы параллельно плоскости изображения. Таков по определению испанского художника Хуана Гриси, этап "аналитического кубизма". Изображение перестает быть представлением, а становится самоценной вещью".

Многогранники в архитектуре Беларуси Национальная библиотека Беларуси

В 1989 году был проведен всесоюзный конкурс на лучший проект будущего сооружения. Его победители – архитекторы Виктор Крамаренко и Михаил Виноградов – предложили модель "белорусского алмаза", в котором сочетались функциональность и современный дизайн.



Идея предполагала возведение оригинального здания в виде *ромбокубооктаэдра* – сложного многогранника из 18 квадратов и 8 треугольников, расположенного на подставке-подиуме (стилобат).

Окружающий нас мир полон изумительно красивых и необычайно сложных фигур. Примерами их можно считать обычный цветок, звезды, листья, горы камней, каждый из которых имеет свою уникальную форму. Среди них отдельное место занимают многогранники, которые на протяжении веков привлекали внимание художников, скульпторов и ювелиров.

ЛИТЕРАТУРА

1. К. Альсина / Тысяча граней геометрической красоты. Москва 2014. С.143.
2. Аракелян Г. Б. Математика и история золотого сечения. — М.: Логос, 2014, 404 с.
3. Шмигевский Н. В. Формула совершенства // Страна знаний. — 2010. — № 4. — С. 2-7.

УДК 004.9

аспирант А. В. Мирончик

Науч. рук. д.т.н., профессор П.П. Урбанович
(кафедра ИСиТ, БГТУ)

ГЕОЛОКАЦИЯ ОБЪЕКТОВ ВНУТРИ ПОМЕЩЕНИЙ

В настоящее время проблема навигации внутри помещений является актуальной. GPS-технология решает множество задач, связанных с определением местоположения объекта вне помещений, однако её невозможно использовать для навигации внутри помещений в связи со значительным ослаблением сигнала в стенах и перекрытиях зданий и недостаточной точностью позиционирования (порядка 5-10 метров). Поэтому для решения данной проблемы необходимо использовать альтернативные способы позиционирования и навигации, например RTLS [1].

RTLS (Real-time Locating Systems — система позиционирования в режиме реального времени) — это автоматизированная система, обеспечивающая идентификацию, определение координат, отображение на плане местонахождения контролируемых объектов в пределах территории, охваченной необходимой инфраструктурой. RTLS накапливает, обрабатывает и хранит информацию о местонахождении и перемещениях людей, предметов, мобильных механизмов и транспортных средств с целью мониторинга технологических и бизнес-процессов, сигнализации об отклонениях от регламентов, а также с целью ретроспективного анализа различных процессов и ситуаций. [1]

Существуют RTLS на базе:

- Cellular-based — в качестве сигнала выступает мощность ближайших базовых станций сотовых операторов телефона. Плюсы в простоте, минусы — точность (50-200м). Плюсы — глобальный масштаб (меньше, чем у GPS, но точность в помещении больше);

- RFID — радио идентификационные метки имеют малый радиус действия (1-2 м для пассивных, десятки метров для активных меток), но этого может быть вполне достаточно чтобы поставить метки в каждый офисный стол. Большая стоимость внедрения такой системы и малый радиус действия;

- WLAN — идентификация с помощью Wi-Fi передатчиков. Не нужно внедрять новое оборудование, если оно уже используется. Довольно хорошая точность. Подходит местам с развитой инфраструктурой. Потребляет большое количество энергии и точно определения 3-5 метров;

- Bluetooth — большое распространение этой технологии в большинстве мобильных телефонов. Меньший радиус действия по сравнению с Wi-Fi;

- Ultra Wide Band — это беспроводная технология связи на малых расстояниях при низких затратах энергии, использующая в качестве несущей сверхширокополосные сигналы. Технология похожа на RFID. UWB метки потребляют меньшую мощность чем RFID. Недостатки те же что и у RFID;

- Infrared — передатчики на основе инфракрасного излучения. Обеспечивают довольно большую точность, но требуют прямую видимость объекта наблюдения.

Более подробно рассмотрим RTLS-систему для геолокации объектов внутри помещений на базе bluetooth сигнала.

Bluetooth — производственная спецификация беспроводных персональных сетей. Bluetooth обеспечивает обмен информацией между такими устройствами, как персональные компьютеры (настольные, карманные, ноутбуки), мобильные телефоны, принтеры, цифровые фотоаппараты, мышки, клавиатуры, джойстики, наушники, гарнитуры на надёжной, бесплатной, повсеместно доступной радиочастоте для ближней связи.

Качество сигнала определяется параметром — RSSI. RSSI (англ. received signal strength indicator) — полная мощность принимаемого приёмником сигнала. Измеряется приёмником по логарифмической шкале в дБм (dBm) [2].

Существует большое количество источников помех влияющих на качество сигнала:

- Wi-Fi сети, которые находятся в зоне действия bluetooth-устройства. Bluetooth и Wi-Fi работают в диапазоне 2,4 ГГц, поскольку радио сигналы обоих устройств мешают друг другу, создается интерференция волн;

- другие bluetooth-устройства;
- большие расстояния между bluetooth-устройствами;
- препятствия (стены, окна, металлические предметы и т.д.);
- различная бытовая техника (СВЧ-печи, передатчики видеосигнала, работающие на частоте 2,4 ГГц).

Существует метод для определения местоположения bluetooth-устройства на основе RSSI сигнала. Но практически в любых помещениях большое количество факторов влияющих на RSSI сигнал, поэтому даже в условиях прямой видимости с bluetooth-устройством полученный RSSI хаотично меняется, даже если устройство не меняет свое местоположение. В результате чего без применения

математического аппарата определить расстояние до маячка становится делом затруднительным [3].

Таким образом, требуется дальнейшая математическая обработка полученных результатов. В данном случае можно применить один из самых популярных алгоритмов фильтрации данных — фильтр Калмана. Фильтр убирает шумы измерения (случайные всплески) и выдаёт результат как с учетом результатов текущих измерений, так и с учётом предсказанных результатов на основе прошлых измерений. Фильтр использует динамическую модель системы (закон движения) и 2 повторяющиеся циклически стадии: предсказание и корректировка [4]. На первом этапе, предсказании, мы рассчитываем состояние системы в следующий момент времени, а на втором, корректировке, корректируем наш прогноз, используя результат очередного измерения.

Для уменьшения погрешности в фильтре Калмана имеется возможно учитывать управляющее воздействие. Таким образом можно применять сразу 2 системы — помимо основного определения положения на основе RSSI-сигнала, использовать для коррекции этого положения инерциальную навигационную систему, состоящую из акселерометра и гироскопа, что значительно улучшает результат. В таком варианте погрешность составляет уже не 3 метра, а 1.

ЛИТЕРАТУРА

1. RTLS – Википедия [Электронный ресурс] / Wikimedia Foundation, Inc. – Wikimedia Foundation, Inc., 2017. – Режим доступа: <https://ru.wikipedia.org/wiki/RTLS>. – Дата доступа: 10.04.2017.

2. Показатель уровня принимаемого сигнала – Википедия [Электронный ресурс] / Wikimedia Foundation, Inc. – Wikimedia Foundation, Inc., 2017. – Режим доступа: <https://ru.wikipedia.org/wiki/WebSocket>. – Дата доступа: 10.04.2017.

3. An Indoor Positioning Algorithm Using Bluetooth Low Energy RSSI [Электронный ресурс] / International Conference on Advanced Material Science and Environmental Engineering (AMSEE 2016), 2016. – Режим доступа: www.atlantispress.com/php/download_paper.php?id=25858154. – Дата доступа: 15.04.2017.

4. Навигация в помещениях с iBeacon и ИНС [Электронный ресурс] / Habrahabr. – Habrahabr, 2017 – <https://habrahabr.ru/post/245325/>. – Дата доступа: 15.04.2017.

секция
ПЕРВЫЙ ШАГ В НАУКУ

УДК 574

Учащ. К. П. Волова

Науч. рук. преп. В.И. Петрушина

(филиал БГТУ «Бобруйский государственный лесотехнический колледж»)

**ЗАЩИТА ОКРУЖАЮЩЕЙ СРЕДЫ
ОТ ВРЕДНОГО ВОЗДЕЙСТВИЯ ХИМИЧЕСКИХ ОТХОДОВ
ПРОМЫШЛЕННЫХ ПРЕДПРИЯТИЙ ГОРОДА БОБРУЙСКА**

Город Бобруйск является одним из крупнейших индустриальных центров Беларуси. Естественно, осуществление больших экономических проектов, постоянный рост объемов выпускаемой продукции с использованием современных химических материалов и сырья, приводит к нарушению химического равновесия в биосфере, что в свою очередь чревато самыми серьезными последствиями для здоровья человека. Поэтому насущной задачей сегодняшнего дня является создание производств, действующих на основе безотходной технологии.

В данной работе показано вредное влияние химических отходов промышленных предприятий на окружающую среду и человека. Но вместе с тем, отмечено, что использование химических веществ на промышленных предприятиях города является неотъемлемой частью выпуска промышленной продукции, повышения благосостояния людей и улучшения качества жизни в целом.

В связи с этим в работе определены приоритетные направления по защите окружающей среды и проведен контроль выбросов промышленных предприятий в городе Бобруйске.

С этой целью осуществлялось знакомство с работой УЗ «Региональный центр гигиены и эпидемиологии» г. Бобруйска, Бобруйской горрайинспекцией природных ресурсов и охраны окружающей среды, государственным учреждением «Могилевский областной центр по гидрометеорологии и мониторингу окружающей среды им. О. Ю. Шмидта».

Определена динамика экологической обстановки в городе Бобруйске за последний год, осуществилось ознакомление с выданными Министерством природных ресурсов и охраны окружающей среды Республики Беларусь разрешениями предприятиям города Бобруйска на выбросы загрязняющих веществ в атмосферный воздух и специальное водопользование и установленными при этом нормативами допустимых выбросов.

В работе предложены меры для достижения максимально полного объема использования химического сырья, минимального коли-

чества отходов и максимальной защиты человека и окружающей среды от вредного воздействия химических веществ.

В работе отмечено, что защита окружающей среды и человека от вредного воздействия промышленных выбросов – это не только региональная проблема города Бобруйска, это проблема международная и роль Беларуси в ней велика.

ЛИТЕРАТУРА

- 1 Э. Иванова «Химические вещества в окружающей среде: проблемы, приоритеты, решения» Минск, УП «Орех», 2017
- 2 Информация, предоставленная Бобруйской горрайинспекцией природных ресурсов и охраны окружающей среды
- 3 А.А. Челноков, Л.Ф. Ющенко «Основы промышленной экологии», Минск, «Вышэйшая школа», 2001
- 4 А.А. Челноков, Л.Ф. Ющенко «Основы окружающей среды и энергосбережение», Минск, «РИПО», 2011

УДК 81

Учащ. К.П.Казакова

Науч. рук. преп. С.Н.Шах

(филиал БГТУ «Бобруйский государственный лесотехнический колледж»)

ЛИНГВИСТИЧЕСКИЙ АНАЛИЗ НАЗВАНИЙ УЛИЦ ГОРОДА БОБРУЙСКА

Бобруйск — один из белорусских древних городов, который расположился на живописном берегу Березины.

Историческое прошлое и географическое положение Бобруйска нашли своё воплощение в названиях его улиц. Названия улиц являются неотъемлемой частью нашего исторического наследия.

В 2017 году Бобруйску исполняется 630 лет. Его улицы на своем веку повидали немало, они могут многое рассказать о своей судьбе, о легендарных людях и событиях.

Актуальность исследования обусловлена тем, что названия улиц города Бобруйска рассматривались только с исторической точки зрения, но не были предметом лингвистического наблюдения, существует необходимость их целостного осмысления: семантической организации, фонетических и словообразовательных особенностей, морфологических свойств.

Было выдвинуто предположение, что названия улиц тесно связаны с социально-экономической историей города и всей страны, с природной средой, с отечественной культурой и наукой; что через

изучение возникновения названий улиц можно пополнить знания по истории родного города.

В ходе исследования были использованы метод наблюдения, метод количественного анализа, метод опроса, исследовательский и пояснительный метод, метод лингвистического описания, систематизация, сопоставление, обработка, классификация и интерпретация материала.

Используя электронную карту города Бобруйска, документы краеведческого музея, был составлен список топонимов. В Бобруйске более 500 улиц, переулков, набережных, проездов. В названиях многих увековечены значительные события из жизни страны, города, имена выдающихся деятелей государства; людей, оставивших заметный след в истории города.

Крупный блок названий представляют «именные» улицы, то есть носящие имена различных известных деятелей. Как и во всяком Белорусском городе или деревне, есть названия улиц, связанные с идеологией, политическими символами.

В оформлении названий улиц города Бобруйска используются имена существительные, прилагательные, числительные.

Анализ показал, что названия улиц города Бобруйска образованы различными способами: субстантивационным, суффиксальным, приставочно-суффиксальным, сложением.

ЛИТЕРАТУРА

1. Голанова Е.И. Как возникают названия. – М.: Просвещение, 1989.
2. Горбаневский М.В. В мире имен и названий. – М.: Знание, 1987.
3. Голубева Т.Ф. Город мой, ты песня и легенда! Бобруйск, 2007.
4. Алексеева С.С. Улицы Бобруйска. «Типография «Дикович и К», Бобруйск, 2005.

УДК 070

Учащ. М. А. Хурсан

Навук. кір. Н. М. Мяркулава

(филиал БГТУ «Бобруйский государственный лесотехнический колледж»)

СРОДКІ ПРЫЦЯГНЕННЯ ЁВАГІ 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 загатоўкаў шматразова ўзрастае — менавіта ад іх залежыць, будзе прачытаны матэрыял ці не.

Мэтай дадзенага даследавання з’яўляецца выяўленне розных сродкаў прыцягнення ўвагі чытачоў у газетных загатоўках; выявіць, артыкулы з якімі загатоўкамі больш прыцягваюць увагу чытачоў. Для дасягнення дадзенай мэты былі выкарыстаны наступныя метады:

1 метады суцэльнага выбару (сабраць газетныя загатоўкі);

2 метады класіфікацыі (класіфікаваць газетныя загатоўкі па выкарыстаных у іх сродках прыцягнення ўвагі);

3 імперычны метады даследавання (праанкеціраваць навучэнцаў з мэтай выяўлення цікавасці да артыкула).

На аснове разгледжанага матэрыялу зроблены наступныя вывады:

1 У загатоўках газеты “Звязда” найчасцей сустракаюцца такія сродкі прыцягнення ўвагі як лексічныя.

2 Нямнога радзей сустракаліся загатоўкі, утвораныя шляхам трансфармацыі назваў мастацкіх твораў, назваў мастацкіх фільмаў і радкоў з песень і загатоўкі, утвораныя пры дапамозе стнаксічных сродкаў.

3 Найбольш рэдка сустракаліся назвы загатоўкаў, 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. Дыскур с 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. Ютертычный, А.А. Заголовок – слово главное / А.А. Тертычный // Журналист. – 2004. – № 1. – С. 80.

УДК 658

Учащ. Д. Гавриленко
Науч. рук. преп. И. Л. Лаптева
(филиал БГТУ «Бобруйский государственный лесотехнический колледж»)

НОВАЯ КОНЦЕПЦИЯ ПОСТРОЕНИЯ ЛИНИИ ДЛЯ АВТОМАТИЧЕСКОГО ВЫПОЛНЕНИЯ ИНДИВИДУАЛЬНЫХ ЗАКАЗОВ

Доля объема экспорта мебели составляет около 40%. Возникает вопрос: есть ли возможность увеличить долю экспорта мебельной продукции и среднестатистическое потребление мебели белорусами. За счет чего мы можем увеличить конкурентоспособность белорусской мебели сейчас, в условиях экономического кризиса.

Одним из путей решения проблемы может быть новый концептуальный подход к построению производства на основе автоматизированных информационных технологий и систем.

Объект исследования — организация производства мебели на современном этапе.

Предмет исследования — многоаспектное безошибочное проектирование и производство мебели на основе САПР — специального комплекса систем автоматизированного проектирования.

Цель данной работы — выявить преимущества многоаспектного безошибочного проектирования для перехода от серийного и массового производства к позаказному на автоматических производственных линиях.

Переход от серийного и массового производства к позаказному требует реинжиниринга бизнес-процессов. Реинжиниринг – это инновационный процесс, направленный на перепроектирование бизнеса — изобретение, позволяющее увеличить показатели в 5–10 раз и более за счет инноваций и внедрения в производство новейших достижений НТП.

Позаказное производство должно сочетать в себе широкий выбор моделей, качество, цены и сроки изготовления заказов, быть высокотехнологичным, сохранять большой объем выпуска. Для такого типа производства следует использовать САПР – специальную комплексную систему автоматизированного проектирования как важнейший элемент CALS-технологий.

CALS-технологии – непрерывная информационная поддержка поставок и жизненного цикла изделий– подход к проектированию и производству высокотехнологичной и наукоёмкой продукции, заключающийся в использовании компьютерной техники и информационных технологий на всех стадиях производства изделия.

Новая парадигма проектирования САПР названа концепцией безошибочного проектирования и производства (БОПП).



Рисунок

На международной выставке «Деревообработка – 2016» в этом году была представлена линия «INDASTRIE 4.0» немецкой корпорации HOMAG, основанная на таком концептуальном подходе изготовления позаказной мебели. Заключение: ее эффективность очевидна – быстрое выполнение заказов, оперативные расчеты за продукцию, отсутствие необходимости аренды или покупки торговых площадей, быстрый возврат оборотных средств, широкий ассортимент, объем производства, высокое качество и производительность. Привлекая в проект организации такого производства инвесторов можно и на территории Беларуси построить современное высокотехнологичное производство.

ЛИТЕРАТУРА

1. Стариков, А.В. Реинжиниринг бизнес — процессов мебельного предприятия при переходе к позаказному производству/ А.В.Стариков, П.А.Петров// Математическое моделирование, компьютерная оптимизация технологий, параметров оборудования и систем управления лесного комплекса.

2. Богатин, Ю.В. Экономическое управление бизнесом «Инжиниринг и реинжиниринг»/ Ю.В. Богатин, В.А.Швандар.

3. Бунаков, П.Ю. Статья «Многоаспектное представление проекта в концепции безошибочного проектирования и производства в корпусной мебели»/ П.Ю. Бунаков, А.В.Стариков, И.А.Бакулин, А.А.Старикова, В.Н.Харин// Журнал «Лесной вестник 4/2008».

4. Печатное издание «Новости деревообработки» – № 4–5 (172–173)/2015г.

УДК 338

Учащ. Е. А. Тедер
Науч. рук. преп. М.А.Голуб
(филиал БГТУ «Бобруйский государственный лесотехнический колледж»)

КАК СОХРАНИТЬ ДЕНЬГИ ВО ВРЕМЯ КРИЗИСА

Цель работы: проанализировать способы сохранения денег, исследовать их особенности и определить наиболее выгодные и эффективные.

Намеченная цель предусматривает необходимость решения следующих **задач:**

- изучить научную литературу по теме исследования;
- определить способы сохранения денег;

– сравнить наиболее распространенные способы сохранения денег и проанализировать полученные результаты.

Актуальность работы заключается в том, что вопрос сохранения и приумножения денег интересует каждого и вполне естественно, что в кризис человек особенно сильно переживает за свой капитал. Поэтому исследование способов сохранения и приумножения накопленных средств актуально во все времена.

Предмет исследования – накопленный капитал.

Объект исследования – способы сохранения денег и их особенности.

Гипотеза: предполагаю, что есть реальная возможность определить наиболее выгодные и эффективные способы сохранения денег и сократить (или уменьшить) риски их потерять.

При работе над исследованием были использованы следующие **методы:** сбор, систематизация, сопоставление, обработка, материала; метод количественного анализа, метод прогнозирования; исследовательский и пояснительный метод.

Практическая ценность заключается в возможности оказания помощи в выборе способа сохранения денег.

Существуют различные способы сохранения сбережений во время кризиса. Перечислим некоторые из них.

- франчайзинг;
- банковские депозиты;
- инвестиции в недвижимость;
- покупка ценных бумаг;
- инвестиции в золото.

При исследовании были использованы материалы официальных сайтов банков Республики Беларусь.

Проведено исследование и выявлен какой из рассмотренных способов является наиболее эффективным.

Рассмотрим на примере куда наиболее эффективно инвестировать капитал в размере 40 000р.

Эти сбережения я могу:

- разместить на банковский депозит на 1 год по наилучшим ставкам в иностранной и национальной валюте;
- инвестировать в золотые мерные слитки;
- инвестировать в недвижимость;
- инвестировать в покупку франчайзи.

Таблица 1 — Расчет сроков окупаемости

Способ инвестирования	Вложенные деньги, р.	Сумма дохода за год	Срок окупаемости, год
Банковский депозит (более 1 года) национальная валюта (15%)	40000	6430	6,7
иностранная валюта (3%)	20000\$	610	33
Облигации (7%)	20000\$	1400	14,3
Покупка золотых слитков (40г. по цене 860р. за 10 гр. и НДС)	34400 +6880 =41280	6192 около 15%	6,7
Инвестиции в недвижимость (покупка квартиры 1 комнатной)	20000\$	100\$ *12=1200 (арендная плата)	16,6
Франчайзинг	40000	8000-10000 около 20-25%	4-5 лет

Средние сроки окупаемости франчайзинговых проектов: до года – для бизнесов с инвестициями до \$15 тыс.; 1,5 года - при инвестициях до \$50 тыс.; до 2 лет – для бизнесов с инвестициями до \$100 тыс.; до 3 лет-производственные и крупные проекты в сфере питания. (other-news/2016/10/25/1254666.html)

Как видно из исследуемых способов наилучший это вложение средств в покупку франчайзи.

Для тех же, кто не привык тратить все до последней копейки, наилучший путь — создать, как любят говорить экономисты, диверсифицированный портфель, то есть такой, в котором будут представлены несколько способов сбережений. Это позволит значительно сократить риски потерять все свои накопления при любом развитии событий.

ЛИТЕРАТУРА

- 1 Иванов, В.В. Деньги, кредит, банки: учебное пособие / В. В. Иванов. – М.: Финансы и статистика, 2011. – 592 с.
- 2 <http://myfin.by>; <http://myfin.by/bank/metally/gold>;
- 3 <http://kodeksy.by/bankovskiy-kodeks>

УДК 338

Учащ. А. О. Округ

Науч. рук. преп. Т. Г. Сенцова

(филиал БГТУ «Бобруйский государственный лесотехнический колледж»)

ВЫГОДНОСТЬ ПРИМЕНЕНИЯ СКИДОК ДЛЯ ПРОИЗВОДИТЕЛЯ И ПОТРЕБИТЕЛЯ

Актуальность выбранной темы заключается в том, что потребители хотят купить такой товар, который бы полностью удовлетворил

его потребность и не сильно ударил по кошельку. Большинство покупателей, при условии, что товар обладает одинаковыми свойствами и качеством выберут тот, который стоит меньше. А если товар можно купить дешевле, чем он стоил ранее, то это воспринимается как подарок.

Так все-таки скидки — это выгодно? Кто выигрывает: производитель или потребитель?

Цель работы — определение выгоды применения скидок для производителя и потребителя.

Исходя из цели, были поставлены следующие задачи:

- 1 рассмотреть историю возникновения скидок;
- 2 провести анализ эффективности применения скидок;
- 3 оценить выгоду применения скидок для производителя и потребителя.

В работе рассмотрена история возникновения скидок. Приводились примеры, кому могли предоставить скидку. Рассматривались торговцы, которые используя такой метод эффективной продажи, как скидки, могли составить достойную конкуренцию крупным торговцам, которые использовали традиционную торговлю.

Для того чтобы провести анализ эффективности применения скидок, была собрана вторичная информация (газеты, журналы, интернет). Также проводилось наблюдение за покупателями в магазине. Опрашивались некоторые производители товаров, предлагающие товары со скидкой.

На основе полученных данных была выявлена выгода применения скидок для производителя и потребителя.

ЛИТЕРАТУРА

- 1 Отчеты ОАО «ФанДок»
- 2 marketch.ru
- 3 peomag.by
- 4 infourok.ru

СПОСОБЫ ЗАЩИТЫ ОТ НЕПРАВОМЕРНОГО ДОСТУПА К КОМПЬЮТЕРНОЙ ИНФОРМАЦИИ В СЕТИ ИНТЕРНЕТ

Цель исследования — определить уровень защиты сайтов от неправомерного доступа в сети интернет и обозначить меры и способы защиты от разного рода взломов.

Результаты исследования.

Мы должны беспокоиться о безопасности информации и знать риски, связанные с предоставлением доступа к конфиденциальным, персональным данным. Электронные средства хранения более уязвимы, чем бумажные: размещаемые на них данные можно и уничтожить, и скопировать, и незаметно видоизменить.

С каждым годом растет число взломов и пострадавших от них владельцев сайтов. Чтобы грамотно защитить свой ресурс, владелец веб-ресурса должен понимать, какие есть варианты взлома, как действует злоумышленник, какие векторы атак наиболее критичны и как их закрыть.

Варианты взлома сайтов:

1. С использованием веб-приложений:
черезCMS;
через плагины;
через подрядчика.
2. С использованием сетевых технологий (беспроводные подключения).

Вариант взлома черезCMS (система управления контентом) заключается в следующем: разработчики допускают ошибки, недостаточно фильтруются входные параметры, не проверяются размещаемые в базе данные, информация выводится на странице «как есть», без безопасных преобразований.

Вариант взлома через плагины: чем больше плагинов, тем больше вероятность наличия уязвимостей. В идеале, использовать решение «из коробки», с примененными патчами и критическими обновлениями, которые закрывают все известные публичные уязвимости в ядре CMS. Но, поскольку, функциональности не всегда хватает, то перед установкой плагинов нужно обязательно проверять, насколько они защищены и безопасны.

Вариант взлома через подрядчика: бывает, что подрядчик устанавливает плагины, которые содержат уязвимости. Например, владелец сайта находит красивый плагин галереи для сайта и просит фрилансера купить и настроить этот модуль. Фрилансер находит такой же плагин на «левом» сайте, берет деньги с заказчика на покупку, но на самом деле скачивает бесплатно.

С использованием сетевых технологий: с помощью лишь одного телефона. (Да, именно поэтому однопользователи больше не просят, чтобы я раздал им интернет). Более серьезные вещи делаются на компьютере и чаще всего на операционной системе Linux.

Sql-инъекции.

Очевидно, что если у сайта есть уязвимости, то его можно взломать с помощью веб-атак. Но даже если сайт защищен техническими средствами, работает на надежной CMS, его все равно могут скомпрометировать. Наиболее популярный взлом через Sql-инъекции.

Sql-инъекция – это атака на базу данных, которая позволит выполнить некоторое действие, которое не планировалось создателем скрипта.

Когда пользователь авторизуется на сайте и передает свой пароль и логин, затем эти данные посылаются в базу данных и сверяются с записями в базе данных, если данные совпадают, то база данных отвечает что пользователь верно ввел данные и сайт разрешает доступ пользователю, если же данные не верны, то база данных возвращает ответ «нет» и сайт не пускает пользователя дальше.

Чтобы было более понятно приведу, пример из жизни.

Отец, написал в записке маме, чтобы она дала Васе 100 рублей и положил ее на стол. Переработав это в шуточный SQL язык, мы получим:

ДОСТАНЬ ИЗ кошелька 100 РУБЛЕЙ И ДАЙ ИХ Васе.

Так как отец плохо написал записку (корявый почерк), и оставил ее на столе, ее увидел брат Васи – Петя. Петя, будучи хакером, дописал там «ИЛИ Пете» и получился такой запрос:

ДОСТАНЬ ИЗ кошелька 100 РУБЛЕЙ И ДАЙ ИХ Васе ИЛИ Пете.

Мама, прочитав записку, решила, что Васе она давала деньги вчера и дала 100 рублей Пете. Вот простой пример SQL инъекции из жизни. Не фильтруя данные (мама еле разобрала почерк), Петя добился успеха.

Как взламывают большинство сайтов: примеры нецелевого взлома.

Чтобы осуществить сам взлом 10 минут не достаточно. Но, буквально вчера был найден уязвимый сайт и принято решение привести его как пример для презентации доклада.

Путем командной строки и программ была получена база данных, таблицы, затем поля из этих таблиц. И путем нехитрых действий просто выбраны поля, которые вызывают особый интерес, а именно пароли и e-mail пользователей. В данном докладе не демонстрируются способы взлома, основная цель – показать, что существует такая возможность.

Программист, владеющий методами технической защиты, несомненно, должен знать технологии взлома защит для того, чтобы, во-первых, не повторять ошибки существующих систем и, во-вторых, создавать более эффективные и надежные механизмы.

Пострадать в результате нецелевой атаки довольно просто: для этого достаточно не заметить или проигнорировать критическую уязвимость в CMS, шаблонах или плагинах своего сайта. Любая незакрытая брешь – отличный шанс очутиться среди «товарищей по несчастью» и прочно закрепиться в хакерской выборке кандидатов для взлома. А в случае «успешной» атаки приготовьтесь к тому, что ваш сайт начнут активно использовать для спам-рассылки, заражения пользователей, хостинга фишинговых страниц, атак на другие сайты или заработка на рекламе. Если на вашем сайте есть слабое звено и веб-проект уязвим к определенному виду атак, то рано или поздно вас взломают. Это нужно понимать каждому веб-мастеру и владельцу сайта.

Как же защититься от взлома?

Защитой сайта и его безопасностью должны заниматься две стороны – это владелец сайта и хостинг провайдер, на котором сайт находится.

Если хакеры взломают защиту хостинга, то они без труда могут получить доступ к файлам любого сайта, расположенного на нем и внедрить вирус или скрытую ссылку.

Не скачивайте плагины, различные шаблоны и модули с сомнительных ресурсов.

Не доверяйте создание и доработку сайта сомнительным людям. Это то, о чем говорилось в начале презентации «про подрядчиков». Отдали сайт человеку на доработку и он вставил туда какой-то новый плагин, это стало уязвимостью сайта.

Вовремя обновляйте ваши CMS, плагины и модули. Потому что зачастую разработчики в обновлениях закрывают дыры в обнаруженных уязвимых местах кода.

Способы защиты:

1. Установите антивирусную защиту.

2. Регулярно обновляйте операционную систему и программы.
3. Скачивайте файлы с надежных ресурсов.
4. Защитите Ваши персональные данные.
5. Используйте надежные пароли.

Конечно нельзя забывать про социальную инженерию. В наше время это популярно. У всех были ситуации, похожие на: «Зайди туда-то, нажми туда-то и получи 1000 \$». Не доверяйте никому и даже самому себе.

Заключение.

Мотив большинства хакеров – это любопытство, вызов в жизни или ложное чувство силы, в то время как мотивом других являются только деньги. Однако, каким бы ни был мотив хакера, взлом вашего компьютера является незаконной деятельностью, а хакеры – это реальные люди, которые могут ворваться в ваш компьютер точно также, как воры могут проникнуть в ваш дом. Поэтому, вы должны знать, как остановить хакеров (воров) и как защитить компьютер от взломщиков, которые могут украсть ваши деньги или повредить вашей репутации.

ЛИТЕРАТУРА

1. SQL-injection в деталях [Электронный ресурс]. – Режим доступа : URL : <http://haknotdie.org/22h/12/10.html>.

2. Методы и способы взломов сайта. Что такое SQL инъекции и что делать если сайт взломали? [Электронный ресурс]. – Режим доступа : URL : <http://semantica.in/blog/vidy-vzlovov-sajtov-chto-delat-esli-sajt-vzlovomali.html>.

3. Атаки sql-injection (mysql) [Электронный ресурс]. – Режим доступа : URL : <http://injection.rulezz.ru/sql-inj.html>.

УДК 674.815

Учащ. В. В.Трутнёв, А. В. Гирдюк,
П. А. Корягин, В. В. Прудников
Науч.рук.преп. С.А. Остапчик, М.М. Шнитко
(филиал БГТУ «Витебский государственный технологический колледж)

ИССЛЕДОВАНИЕ СВОЙСТВ ПЛИТ МДФ

Введение. Тема исследовательской работы основана на том, что в последние два десятилетия плиты МДФ вытесняют из мебельного и столярно-строительного производства такие древесные материалы как древесно-стружечные плиты (ДСП), ламинированные древесно-стружечные плиты (ЛДСП) и фанеру.

В данной работе исследования направлены на изучение свойств плит МДФ и сравнение их со свойствами других плитных материалов. Исследования проводились методами изучения литературных источников, материалов сети интернет, а также проведением практических опытов.

Из истории производства МДФ. Первая плита, состоящая не из измельченной стружки, а из древесных волокон, была случайно создана в 1924 году американцем Уильямом Мейсоном. Древесноволокнистая плита средней плотности, более известная нам по буквенной комбинации МДФ, происходящей от английского названия этого материала — medium density fiberboard — запущена в массовое производство около 1966 года в США. Разработки этой плиты велись одновременно многими производителями, поэтому точно назвать изобретателя, создавшего первую плиту МДФ, невозможно. В России плиты МДФ не производились, первая линия по производству этого материала появилась лишь в 1997 году в Вологодской области. На сегодня крупнейшим мировым производителе плит МДФ и изделий из них является Китай. В Республике Беларусь первый цех по изготовлению плит МДФ был введен в эксплуатацию в 2013 году

Область применения плит МДФ. Древесноволокнистые панели популярны у дизайнеров — из этого материала выполняется декоративная обшивка потолков, создаются воздуховоды, подоконники, эстетически привлекательные и прочные створки межкомнатных дверей.

Но более всего МДФ плиты применимы в мебельной промышленности. Разработанная техника гнутья конструкционных элементов из склеенного древесного волокна позволила создавать из этого материала кресла и стулья, изящные спинки кроватей. Как правило, вся мебель для помещений с высокой атмосферной влажностью (санузлов, кухонь) выполняется из шпонированного или ламинированного МДФ.

Свойства МДФ. Госстандарта России и Беларуси на МДФ не существует, производители МДФ либо разрабатывают собственные технические условия, либо выпускают плиты по европейскому стандарту ANSI A208.2.

В процессе производства плитам МДФ могут быть приданы более высокие качественные свойства по водостойкости, огнестойкости и биостойкости, чем у серийных плит.

Плюсы плит МДФ:

- ✓ прочностные характеристики древесноволокнистых плит лишь чуть ниже, чем у естественной древесины;

- ✓ влагостойкость;
- ✓ длительный срок службы;
- ✓ широкая палитра поверхностных материалов отделки;
- ✓ монтажные работы проводятся в короткий срок;
- ✓ благодаря плотности МДФ плит, их можно обрабатывать фрезами, создавая объемный рельеф;
- ✓ стоимость 16 мм ламинированного МДФ примерно равна 8.6 руб. за квадратный метр.

Минусы плиты МДФ:

- ✓ тяжёлая;
- ✓ плохое удержание фурнитуры;
- ✓ работы по фрезерованию сопровождаются обильным пылеобразованием;
- ✓ несущие способности значительно уступают древесине;
- ✓ в состав входят фенолформальдегидные смолы.

Сравнение свойств МДФ и ЛДСП

В процессе работы были проанализированы свойства МДФ и ЛДСП, некоторые из них представлены в таблице.

Практический опыт

Учитывая возможности учебных мастерских был проведён следующий практический опыт. На двух образцах плит МДФ и ЛДСП концевой фрезой были выбраны отверстия под петли. После обработки даже визуально видно, что шероховатость поверхности после обработки ЛДСП гораздо больше, чем у МДФ. Из этого следует: качество поверхности после механической обработки у плит МДФ лучше, что дает возможность нанесения защитно-декоративного покрытия методом крашения, а также упрощает процесс подготовки плит к облицовыванию.

Выводы

Широкое применение плит МДФ обусловлено:

1. достаточно высокой водостойкостью;
2. механической прочностью;
3. высокой плотностью;
4. способностью к гнущю и принятию криволинейных форм;
5. возможностью механической обработки под крашение и облицовывание тонкими пленками;
6. достаточно высокой биостойкостью;

Качественные показатели плит МДФ оправдывают ее более высокую стоимость по сравнению с ЛДСП.

УДК81

Учащ. В. А. Попова, В. А. Литвинюк
Науч. рук. преп. А. В. Цвирко

(филиал БГТУ «Витебский государственный технологический колледж»)

**ЭПОХА ПЕТРОВСКИХ РЕФОРМ. ГЕРМАНИЗМЫ
(ШТРАФ, АБЗАЦ ИЛИ КРАСНАЯ СТРОКА)**

Зачастую при изучении иностранного языка мы сталкиваемся с такой острой проблемой как отсутствие мотивации к его изучению. В своей исследовательской работе мы попытались продемонстрировать положительное влияние немецкого языка на русский, с целью формирования интереса к изучению иностранного языка, к иноязычной культуре.

Предметом нашего исследования являются заимствования, проникшие в русский язык в эпоху Петра I, их влияние на структуру и развитие русского языка в то время и на современном этапе.

Актуальность нашей темы обусловлена тем, что в последнее время немецкий язык незаслуженно вытесняется английским, несмотря на то, что культурные, политические и экономические контакты с Германией становятся все тесней.

Методы исследования: изучение и анализ специальной и справочной литературы по данной теме, сравнительно-исторический метод, метод анализа языка на этимологическом уровне, описательный метод.

В начале исследования нами были поставлены следующие цели:

1. Изучить понятие «германизмы» и историю их проникновения в русский язык

2. Ознакомиться с личностью Петра Великого и оценить его вклад в развитие государства Российского

3. Произвести компаративный анализ лексических единиц

4. Собрать и проанализировать имеющуюся информацию по теме исследования

5. Воспитывать потребность в практическом использовании немецкого языка

6. Формировать уважение к иноязычной культуре

7. Расширять общий и филологический кругозор

В заимствовании русским языком иноязычных слов в разные эпохи отразилась история нашего народа.

Самые первые заимствования из германских языков проникали в русский еще в VIII-XII вв. Например, меч – dasSchwert, панцирь (броня) – derPanzer, котел – derKessel, холм – derHügel.

Политические и культурные связи двух народов усилились в XV веке при Иване III и в XVII веке при царе Алексее Михайловиче.

Лексическое влияние европейских языков на русский особенно активизировалось в Петровскую эпоху. Преобразование всех сторон русской жизни при Петре I, его административные, военные реформы, успехи в просвещении, развитие науки - все это способствовало обогащению русской лексики иноязычными словами.

Времена правления Петра Первого характеризуются проникновением большого потока «германизмов». Германизмы — это слова или выражения, заимствованные каким-либо языком из германских языков, или оборот речи, построенный по образцу этих языков.

18 августа 1682 года на престол взошел Пётр I. Великий русский царь, а впоследствии Император, правил страной на протяжении 43 лет. Великий государь отличался необычайной способностью к преобразованиям. Пожалуй, невозможно отыскать такую сферу, которая не претерпела бы изменений, вызванных энергией деятельного царя. Большое влияние на характер реформ оказала симпатия Петра к западному образу жизни и быту, зародившаяся в ранней юности, во время частых его визитов в Немецкую слободу в Москве. Этим объясняется то, что реформам быта придавалось особенное значение.

Одной из важнейших задач, поставленных царем, было воспитание образованных людей, необходимых для построения нового государства. Начало Светскому образованию в России было положено открытием указа 14 января 1701 г. В петровское время устойчиво развивалась отечественная наука. В 1725 г. в Петербурге была открыта первая Академия наук. Сегодняшнее профессиональное образование также во многом обязано Петру I. Появились первые навигационные, морские и инженерные школы.

При Петре I церковно-славянский шрифт был заменен новым более простым гражданским, которым мы пользуемся до сих пор.

Огромное влияние на язык того времени оказало проникновение целого ряда иностранных слов и новых понятий в науку, технику и искусство: dieAktie - акция, derDrillbohrer – дрель, derWerkbank – верстак, dasLager – лагерь.

Из Германии тогда были приглашены на работу в молодую столицу империи – Санкт-Петербург – большое количество ученых и специалистов. В 1716 году вышел указ Петра об обязательном обучении административных служащих немецкому языку.

Заимствование из языка в язык может идти двумя путями: устным и письменным, через посредство книг. При письменном заим-

ствовании слово изменяется сравнительно мало. При устном же облик слова часто изменяется сильнее.

Основные причины заимствования:

– экстралингвистические (исторические контакты народов, экономика языковых средств, авторитетность языка-источника)

–внутрилингвистические:

1) отсутствие в родном языке эквивалентного слова для нового предмета или понятия. Приведем несколько примеров германизмов, которые дали жизнь новым понятиям: *derPreiskurant* – прейскурант, *derAgent* – агент, *dieMarschrout* – маршрут.

2) тенденция к использованию одного заимствованного слова вместо описательного оборота, т.е., появление слов, которые упрощают состав языка, в который проникают, например абзац (вместо «красная строка»), айсберг (вместо «ледяная гора»), кастрюля (вместо «круглый котел»), курорт (вместо «место лечения»).

После Петровских реформ немецкий язык на протяжении нескольких десятков лет оказывал сильное влияние на лексику русского языка. В конце XVIII века знание иностранных языков использовалось для лучшего овладения специальностью и являлось показателем образованности и зрелости дворянина.

Петербург тяготел к французскому языку, Москва – к немецкому. В 1724 году был открыт Петербургский университет. В первые годы преподаватели приезжали из Германии и до начала сороковых годов преподавание велось на немецком языке.

В ходе нашей работы мы изучили историю, проанализировали и выявили причины и пути проникновения германизмов в русский язык. Также мы ознакомились с личностью Петра Великого и оценили его вклад в развитие государства Российского. Петровская эпоха в истории нашего народа характеризуется существенными реформами. Эти преобразования совершили переворот в сознании и в привычках русских дворян.

Сегодня сложно себе представить русский язык без внедренных в него иностранных слов. В повседневной жизни мы часто используем заимствованную из немецкого языка лексику. Такие слова как акция, пакет, картофель появляются в нашей речи часто, без них трудно себе представить речь русскоязычной части населения. Такая замена была жизненно необходима для упрощения и обогащения состава русского языка, для выражения новых понятий в области техники, культуры и военного дела.

Изучая германизмы, мы пришли к выводу, что заимствование слов из других языков — это естественный и необратимый процесс,

который обусловлен международными связями. В ходе данного исследования менялось и наше представление о многих словах русского языка, что заставляет нас вспомнить слова великого немецкого поэта Иоганна Вольфганга фон Гёте, который говорил: «Wer eine Fremdsprache nicht kennt, weiß nichts von seiner eigenen». — «Кто не знает иностранного языка, тот не знает ничего о своём собственном».

Таким образом, сильное влияние немецкого языка на русский началось еще со времени Петра Великого, когда иностранные слова буквально «пропитывали» русскую речь. Но это был не единственный период проникновения германизмов в речь русскоязычной части населения. Заимствования из немецкого языка активно проникали в русский и в революционный период (1917–1926 гг.), и во время Великой Отечественной войны (1941–1945 гг.). В дальнейшем мы планируем изучить эти заимствования и сделать их темой нашей будущей научно-исследовательской работы.

УДК 315.5

Учащ. Н. В. Бабуль

Науч. рук. преп. И. В. Ковалькова

(филиал БГТУ «Витебский государственный технологический колледж»)

РАННИЙ БРАК – ПРОБЛЕМА И АКТУАЛЬНОСТЬ СОВРЕМЕННОГО ОБЩЕСТВА

Семья – это основанное на браке или кровном родстве объединение людей, связанных общностью быта, взаимной моральной ответственностью и взаимопомощью. Как необходимый компонент социальной структуры любого общества, выполняющий многие социальные функции, семья играет важную роль в общественном развитии. Через семью сменяются поколения людей, в ней осуществляется продолжение рода, происходит первичная социализация и воспитание детей вплоть до достижения ими гражданской зрелости.

Основу семьи составляет брачный союз между мужчиной и женщиной, санкционированный обществом. Своим возникновением, существованием и развитием семья обязана, прежде всего, общественным потребностям, нормам и санкциям, предписывающим супругам заботиться о своих детях. Как социальное явление семья изменяется в процессе развития экономического базиса общества; в то же время развитие семьи обладает относительной самостоятельностью.

Законодательством Республики Беларусь брачный возраст устанавливается с 18 лет, но, как и во многих странах предусмотрено

снижение брачного возраста определенных ситуациях. Такие положения законодательно закреплены в статье 18 Кодекса Республики Беларусь о браке и семье. А именно:

- ✓ беременность, постановка на учет в женскую консультацию;
- ✓ рождение ребенка (причем, не имеет значения факт родства ребенка по отношению к будущему супругу);
- ✓ приобретение несовершеннолетним полной дееспособности до достижения совершеннолетия.

При наличии таких обстоятельств лицам, желающим вступить в брак, всего лишь необходимо обратиться в орган, регистрирующий акты гражданского состояния, с заявлением **оснижении брачного возраста**.

Как считают многие, причины ранних браков следующие:

- ✓ Сильное чувство «любви». Такие аргументы, как отсутствие жилья и финансовой защиты на влюбленных просто не действуют.
- ✓ Стремление узаконить свои отношения.
- ✓ Статистика ранних браков называет беременность самой распространенной причиной.
- ✓ Желание освободиться от чрезмерной родительской опеки.

Нельзя с уверенностью сказать, что ранние браки имеют только негативные последствия, они имеют как свои «плюсы», так и «минусы».

Плюсы ранних браков:

1. возможность рождения или уже воспитания ребенка в полной семье, когда у него уже будет папа и мама;
2. возможность преодолеть психологический барьер девушке, но уже ставшей женщиной, будучи в статусе не матери одиночки, а жены.

Минусы ранних браков (непосредственно связаны с негативными последствиями):

1. незрелость личности – один из основных факторов, влияющих на количество разводов. О сформированности личности в 16–18 лет речь идти не может. В таком возрасте человек еще фактически является ребенком, пусть и выросшим, но все же инфантильным;
2. невозможность самореализации в профессиональном.

Не все готовы к таким переменам и реально оценивают плюсы и минусы ранних браков, а в результате, возникают конфликты, непонимания, обиды и прочие негативные вещи, которые, если не разрешаются, то неизбежно ведут к развалу отношений.

Поэтому основной задачей государства является создание благоприятных для рождения желанных и здоровых детей социально-экономических условий, которые в себя включают:

- ✓ систему мер и законодательных актов по поддержке молодой семьи, особенно при рождении ребенка;
- ✓ создание условий, позволяющих гражданам реализовывать свои репродуктивные права, таких, как центры планирования семьи, молодежные центры.

Для чего нужно планирование семьи:

- ✓ для рождения желанных здоровых детей;
- ✓ для достижения гармонии в психосексуальных отношениях в семье;
- ✓ для того, чтобы родительство было ответственным, чтобы на свет появлялись желанные и здоровые дети.

Целями программы являются:

- ✓ формирование у подростков позитивного отношения к здоровому образу жизни, планированию семьи и ответственному родительству;
- ✓ помощь в осознании подростками своих обязанностей и ответственности по отношению к себе и окружающим.

В программу включены разделы, посвященные взаимосвязи семьи и общества, а также юридические и законодательные вопросы в этой области.

Законодательство Республики Беларусь, регулирующее брачно-семейные отношения, в полной мере учитывает современные позиции мирового сообщества в зарождении и становлении семьи. Изучив положительный опыт Финляндии, актуальным решением проблемы раннего брака является разработка программы, направленной на становление и обеспечение интересов семьи.

Вашему вниманию предлагается проект программы курса планирования семьи «Молодая семья».

Задачами данной программы будут являться:

- ✓ защита прав и законных интересов несовершеннолетних детей и их несовершеннолетних родителей;
- ✓ оказание помощи государственными органами, учреждениями, организациями в области: здравоохранения, образования, оказание правовой помощи при разрешении спорных вопросов.

Завершая исследовательскую работу, мы пришли к выводу, что порядок снижения брачного возраста в Республике Беларусь, действующий в настоящее время, необходимо урегулировать нормами процессуального характера. Для достижения данной цели предлагаем

учредить на базе органов местного управления специализированные органы по работе с молодежью, которые будут работать в тесном контакте с учреждениями образования, здравоохранения, социальными и правовыми службами.

УДК 620.9

Учащ. К. В. Ковалёв

Науч. рук. преп. Л. П. Кудрина

(филиал БГТУ «Белорусский государственный колледж промышленности строительных материалов»)

ЭНЕРГОСБЕРЕЖЕНИЕ В БЫТУ ДЛЯ ВСЕХ И КАЖДОГО

Современная Беларусь обеспечивает себя своим сырьем только на 20–25%. Задача каждого гражданина Республики Беларусь – это экономия электроэнергии. Наша страна большинство энергоресурсов закупает у других государств, из-за чего повышение потребления электричества повлечет резкое повышение тарифов, что и происходит на данный момент.

Коммунально-бытовое хозяйство является на сегодня крупным потребителем топлива и энергии: на его долю приходится около 20% топливно-энергетических ресурсов. Потребление электроэнергии в жилом секторе достигает сейчас более 600 миллионов кВт*ч. Из них около 40% расхода электроэнергии приходится на электробытовые приборы, 30% расходуется на освещение и более 12% — на приготовление пищи. Самыми крупными потребителями электроэнергии в коммунально-бытовом хозяйстве являются жилые дома. В них ежегодно расходуется в среднем 400 кВт*ч на человека, из которых примерно 280 кВт*ч потребляется внутриквартиры на освещение и бытовые приборы различного назначения и 120 кВт*ч – в установках инженерного оборудования и освещения общедомовых помещений. Внутриквартирное потребление электроэнергии составляет примерно 900 кВт*ч в год в расчёте на «усреднённую» городскую квартиру с газовой плитой и 2000 кВт*ч – с электрической плитой.

Люминесцентные (энергосберегающие) лампочки. Экономия электроэнергии. Коэффициент полезного действия у энергосберегающей лампы очень высокий и световая отдача примерно в 5 раз больше чем у традиционной лампочки накаливания. Например, энергосберегающая лампочка мощностью 20 Вт создает световой поток равный световому потоку обычной лампы накаливания 100 Вт. Благодаря такому соотношению, энергосберегающие лампы позволяют экономить

экономии на 80% при этом без потерь освещенности комнаты привычного для вас. По сравнению с традиционными лампами накаливания, энергосберегающие лампы служат в несколько раз дольше. Энергосберегающие лампы, имея другую конструкцию и принципиально иной принцип работы, служат дольше ламп накаливания в среднем в 8 раз. Это примерно от 5 до 12 тысяч часов работы лампы (обычно ресурс работы лампы определяется производителем и указывается на упаковке). Опасны они же тем, что в колбе лампы содержатся пары ртути, и при разгерметизации сосуда может быть нанесен вред здоровью человека.

Светодиодные лампы. Преимущества светодиодных ламп.

Во-первых, электропотребление в 10 раз меньше, чем у ламп накаливания и в 3 раза меньше, чем у люминесцентных ламп;

во-вторых, срок службы около 3–5 лет непрерывной работы.

Есть еще достоинства – это их относительная безвредность, т.е. отсутствие ртути и простая утилизация. Это не влияет на снижение электропотребления, а кроме ртути есть другие металлы, о них молчат производители. Однако есть весомые недостатки, в следствие которых эти лампы не получили широкое применение: Высокая цена единичной лампы (окупаемость такой лампы составляет 5 лет, что равно ее сроку службы), достаточной быстрая деградация кристаллов (в следствии свет становится тусклее), очень маленькая площадь освещения (вам понадобится больше таких ламп для привычного освещения) и сложное устройство лампы, которое довольно быстро выходит из строя из-за перепадов напряжения в сети питания.

Также, с целью экономии, можно произвести следующие операции: замена простых ламп накаливания на криптонные лампы накаливания, замена двух ламп меньшей мощности на одну несколько большей мощности (использование одной лампы мощностью 100 Вт вместо двух ламп по 60 Вт каждая экономит при той же освещенности потребление энергии на 12%), поддержание допустимого напряжения, периодическая замена ламп к концу срока службы (около 1000 ч), периодическая чистка от пыли и грязи ламп, плафонов и осветительной арматуры, снижение уровня освещенности в подсобных помещениях, коридорах, туалетах и т.п., широкое применение светорегуляторов, позволяющих в широких пределах изменять уровень освещенности и т. д.

Также имеются и другие способы экономии электроэнергии. Например с помощью установки элементов автоматики (таймеры отключения, датчики света и движения, звуковые датчики), располагать лампы светящейся частью вверх так, что бы они светили на потолок

(лучше всего что бы потолок не поглощал свет и был белого цвета, это позволяет установить меньшее количество источников света на большой площади для достижения необходимого равномерного освещения), довольно простой способ поставить полупроводниковый диод (300В, 3А) в разрыв провода, включающего освещение в подъезде или подвале.

Солнечные батареи. В частных домах целесообразно применять солнечные батареи, которые, благодаря прогрессу, может вырабатывать достаточное количество электроэнергии для полной автономии жилого частного дома (дачи).

Экономия электроэнергии на кухне. Если вы пользуетесь электрочайником, то налейте столько, сколько нужно вам сейчас. Если вы пользуетесь электроплитой, то вам следует знать, что при выборе посуды, которая не соответствует размерам плиты, теряется 5–10% энергии, посуда с искривленным дном «ворует» до 40–60%. Итак, дно посуды должно быть ровным и с размером, соответствующим диаметру конфорки. Так же стоит заменить обыкновенные конфорки электроплиты на конфорки технологии “HiLight”, или же индукционной плитой. Конфорки “HiLight” нагреваются всего за 5–7 секунд, однако куда более практичны индукционные плиты. Индукционные плиты нагревают продукты в 1,5–2 раза быстрее, чем любые другие, включая газовые. По точности температурного режима и эффективности работы им нет равных. Такой способ позволяет наиболее эффективно эксплуатировать энергоресурсы. Однако у такой плиты есть и свой недостаток: для индукционной плиты требуется специальная индукционная посуда. Помните, что быстрое испарение воды при кипении увеличивает время приготовления пищи на 30%. После закипания жидкости нужно уменьшить мощность, подаваемую на конфорку.

Так же следует выбирать электроприборы с классами энергоэффективности А и А+. Определенный класс присваивается прибору в зависимости от количества киловатт, которое он потребляет.

Кондиционеры. В жаркое время года, если в квартире установлен кондиционер, то для экономии электроэнергии целесообразно установить на окна металлические шторы или любые другие, которые отражают инфракрасное излучение от солнца, которое и нагревает помещение, вследствие чего кондиционеру приходится не так часто охлаждать помещение, что бы поддерживать необходимую температуру, и вследствие чего и экономится электроэнергия. Так же стоит выбирать кондиционеры с классом энергоэффективности А и А+.

УДК 62-1

Учащ. А. М. Лашко

Науч. рук. преп. Н.М. Атысова

(филиал БГТУ «Белорусский государственный колледж промышленности строительных материалов»)

МОДЕРНИЗАЦИЯ ЦЕМЕНТНОЙ МЕЛЬНИЦЫ 3.2 X 15М С ЦЕЛЮ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ

Цемент является одним из главных видов строительных материалов. Процесс помола цементного клинкера с добавками является важной стадией производства, предшествующий получению готового продукта — цемента необходимого качества.

Многокамерная трубная шаровая мельница $\text{Ø}3,2 \times 15$ м предназначена для тонкого измельчения известняка, мергеля, клинкера, угля, химического и керамического сырья, а также других материалов, как для сухого, так и для мокрого способа измельчения.

Производительность мельницы до модернизации составляет 270 т\ч.

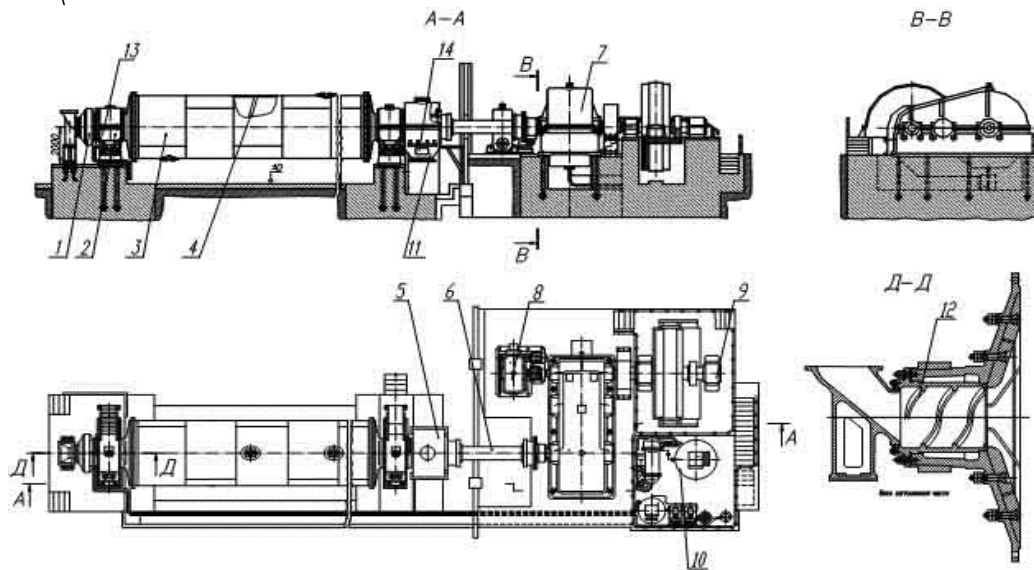


Рисунок 1 – Двухкамерная трубная мельница $\text{Ø}3,2 \times 15$ м.

Трубная мельница представляет собой цилиндрический горизонтально расположенный вращающийся барабан, частично заполненный мелющими телами (стальными шарами и цельпесом). К барабану на болтах присоединяются торцевые крышки загрузочная и разгрузочная, выполненные заодно с пустотелыми цапфами, опирающимися на два подшипника. В первой камере мельницы происходит измельчение материала шарами размером $\text{Ø}100$, $\text{Ø}90$, $\text{Ø}60$. Измельченный материал проходит через отверстие межкамерной перегородки и поступает во вторую камеру мельницы, где измельчение произ-

водится цельпесом. Готовый материал, прошедший через выходную решетку, захватывается лопастями и поднимается вверх, сползая по внутренней стенке днища и внешней стенки разгрузочного конуса. Далее он попадает в разгрузочный патрубок, через отверстие в нем проваливается на вращающееся вместе с мельницей сито и выходит в разгрузочный кожух (аспирационная коробка).

Барaban и днище мельницы изнашиваются вследствие истирания измельченным материалом и ударами мелющих тел. Для предохранения от преждевременного износа их облицовывают стальными плитами.

Основными достоинствами этой мельницы является простота конструкции, надежность эксплуатации, высокая степень измельчения. Однако наряду со всеми достоинствами у нее есть существенные недостатки: высокая удельная металлоемкость, низкий КПД (электроэнергия, расходуемая непосредственно на измельчение, составляет всего лишь около 1% всей потребляемой электроэнергии); невысокая производительность при её габаритах.

Для увеличения производительности данной мельницы внутри загрузочной и разгрузочной части необходимо установить лопастные эллипсные сегменты, которые наклонены к соответствующим торцевым днищам.

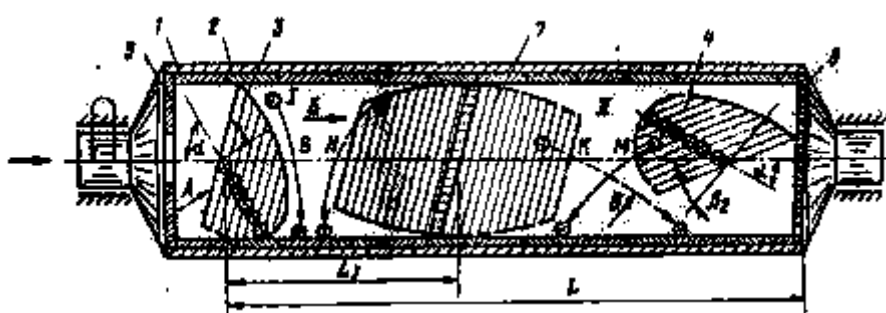


Рисунок 2 – Модернизация цементной мельницы

Между сегментами установлена межкамерная перегородка 7, выполненная в виде лопасти двухстороннего действия со срезанными образующими, под углом $2-30^\circ$ в направлении вращения барабана мельницы. Площадь срезанных частей не должна быть меньше площади ее живого сечения. Срезанные части лопасти обеспечивают беспрепятственный проход измельчаемого материала во вторую камеру, кроме того выравнивают уровни мелющих тел в камерах. Приведенный угол среза предотвращает перемещение мелющих тел и материала обратно из второй камеры в первую.

Применение энергообменных устройств обеспечивает разрушение застойных зон мелющей загрузки и включение в работу дополни-

тельного количества (до 40%) мелющих тел. Это позволяет уменьшить общую массу мелющих тел на 10–15%, повысить производительность мельницы и снизить удельный расход электроэнергии на 15–20%.

Перегородка наклонена в сторону разгрузочной части и повернута относительно большой оси первого со стороны загрузки сегмента на угол $\varphi = 70-120^\circ$. Второй эллипсный сегмент, наклонен в ту же сторону, что и лопасть и повернут относительно ее большей оси на угол $\varphi = 20-40^\circ$.

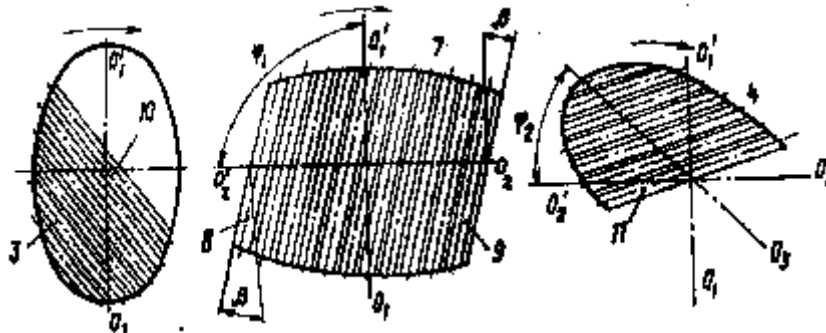


Рисунок 3 – Модернизация цементной мельницы

Материал поступает в загрузочную часть мельницы, где захватывается остроугольным краем эллипсного сегмента и направляется в камеру. Камера ограничена первым эллипсным сегментом и лопастью двухстороннего действия, которые образуют трапециевидальный профиль, что обеспечивает подъем мелющих тел на значительную высоту и интенсивное ударно-истирающее воздействие, необходимое для грубого помола материала, находящегося в первой камере. Угол поворота лопасти обеспечивает предотвращение выбивания высокоподвижного шлама через уплотнение загрузочной цапфы. Затем материал поступает во вторую камеру, ограниченную лопастью и эллипсным сегментом, имеющих одинаковый наклон и образующих камеру с параллельными плоскостями. Установленные таким образом устройства обеспечивают интенсивное раздавливающее воздействие мелющих тел на измельчаемый материал. Параллельное расположение рабочих плоскостей лопасти и сегмента исключает подъем мелющих тел на большую высоту. Поворот выходного эллипсного сегмента исключает выбивание шлама через разгрузочную решетку. Таким образом, данная схема обеспечивает интенсивный внутренний цикл измельчения материала. Недостатком данной конструкции является повышенная металлоемкость, и область применения данной машины является возможность применения только при мокром измельчении сырьевых материалов.

Так как в последнее время широкое применение получил мокрый способ производства цемента, то целесообразно использовать данную модернизацию мельницы.

Углы наклона обеспечивают интенсивное ударное воздействие мелющих тел на материал в первой камере. Эллипсный сегмент выполняет не только энергообменную функцию, но и транспортирующую, обеспечивая импульс только по ходу движения. При этом необходимо принимать углы в строго определенном диапазоне, в противном случае происходит только ухудшение качества процессов. Проходя через перегородку и классифицируясь материал поступает во вторую камеру. Здесь происходит измельчение с постепенным изменением режима от ударно — истирающего со стороны перегородки до раздавливающее — истирающего со стороны сегмента.

Таким образом, предполагаемое сочетание внутримельничных устройств с их заданными геометрическими параметрами и взаимным расположением в мельнице обеспечивает наиболее рациональный интенсивный режим измельчения материала по длине барабана мельницы, позволяет варьировать режим измельчения в зависимости от размолоспособности материала и эффективно использовать полезный объем мельницы по всей ее длине.

Использование предлагаемой мельницы обеспечит в промышленных условиях повышение производительности мельницы на 10-15%. За счет интенсификации процесса измельчения материала по всей длине мельницы и повышение качества (тонкости помола) измельчаемого продукта за счет усиления раздавливающее - истирающего воздействия в поперечно - продольном направлении во второй камере.

Производительность трубной шаровой мельницы определяется по формуле:

$$Q = 6.45 \cdot V \cdot \sqrt{D_{CB}} \cdot \left(\frac{m}{V}\right)^{0.8} \cdot q \cdot K_{уд.}$$

где диаметр мельницы в свету, м:

$$D_{CB} = 2 \cdot R_{CB}$$

$$D_{CB} = 2 \cdot 1,5 = 3 \text{ м}$$

q – поправочный коэффициент, учитывающий тонкость помола, $q = 0,95$

$K_{уд}$ – удельная производительность, т/кВт.ч, $K_{уд} = 0,035$

m – масса мелющих тел, т;

V – полезный объем мельницы, м³.

$$V = \frac{\pi \cdot D_{св}^2 \cdot L}{4}$$

где L – длина барабана за вычетом толщины межкамерных перегородок, $L = 14,54 м$

$$V = \frac{3,14 \cdot 3^2 \cdot 14,54}{4} = 102,77 \text{ м}^3$$

$$Q = 6,45 \cdot 102,77 \cdot \sqrt{3} \cdot \left(\frac{141,3}{102,77} \right)^{0,8} \cdot 0,95 \cdot 0,035 = 49,25 \text{ т/ч}$$

Исходя из того, что при установке сегментных эллипсов, производительность мельницы увеличивается на 10-15%, то принимаем производительность мельницы равную

$$Q_{ДЕЙСТВ} = Q \cdot 1,125 = 55,4 \text{ т/ч}$$

Данная модернизация способствует интенсификации процесса измельчения материала по всей длине мельницы, повышению качества (тонкости помола) измельчаемого продукта за счет усиления раздавливающее - истирающего воздействия в поперечно - продольном направлении во второй камере.

УДК 62-1

Учащ. Е. С. Рыжкович, А. Е. Пузыревский
Науч. рук. преп. Н. А. Храповицкая
(филиал БГТУ «Полоцкий государственный лесной колледж»)

БИОТОПЛИВО. МИФ ИЛИ РЕАЛЬНОСТЬ? НА ПРИМЕРЕ УЧЕБНО-ОПЫТНОГО ЛЕСХОЗА ФИЛИАЛА БГТУ ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ ЛЕСНОЙ КОЛЛЕДЖ

Основное направление ресурсосбережения в лесной промышленности — рациональное использование древесного сырья, а также расширение использования и переработки древесных отходов в качестве заменителя деловой древесины, позволяющее достичь ощутимого экологического эффекта, состоящего в сокращении вырубаемых лесных площадей, сохранении природной среды и т.д.

Промышленно-хозяйственная деятельность лесного комплекса тесно связана с проблемами развития природоохранных и социальных функций лесов. Ограничение на дальнейшее увеличение объемов заготавливаемого древесного сырья вместе с требованиями сохранения и улучшения состояния лесной среды как части биосферы, с необходимостью повышения эффективности и использования всей биомассы, получаемой на лесосеках, требуют переориентации всего комплекса на ресурсосберегающий путь развития. Этот переход возможен только на основе использования новейших достижений науки и тех-

ники, внедрения безотходных технологий, расширения объёмов использования вторичных ресурсов и отходов производства.

Уже сегодня около 16% общего объема энергии в Республике производится из местных видов топлива (древесина, торф и т.д.). Для увеличения этой доли, в первую очередь, наряду с торфом Беларусь может рассчитывать на древесную массу. С одной стороны, древесные отходы в значительных количествах образуются при заготовке древесины и деревообработке, с другой – достаточно зайти в лес и увидеть, что такой древесины у нас хватает.

В Беларуси биоэнергетика начинает интенсивно развиваться в условиях необходимости достичь определенного уровня энергетической безопасности и в полном соответствии положениям Международного соглашения об изменении глобального климата, подписанного Республикой. Конечно, целью развития биоэнергетики является создание собственного топливно-энергетического цикла на возобновляемых видах биотоплива с учетом экологических и экономических преимуществ данного направления.

По оценке ИПЭ НАНБ совместно с Министерством лесного хозяйства технически доступен для биоэнергетики в настоящее время объем отходов, эквивалентный приблизительно 1,5 млн. т у.т./год. Согласно официальным данным Белорусского энергетического института только 25% этой величины используется в настоящее время.

К 2020 году потенциальные топливные ресурсы для биоэнергетики оцениваются в 3,7 млн. т у.т./год

Основными источниками топливной древесины или получаемого в лесу древесного топлива в учебно-опытном лесхозе являются:

- тонкомерная древесина, заготавливаемая при рубках ухода и прореживания молодняка;
- тонкомерная древесины, являющаяся побочным продуктом первых промышленных рубок прореживания;
- отходы лесосек, на которых проводятся рубки главного пользования;
- отходы деревообрабатывающей промышленности.

В настоящее время большинство лесосечных отходов (а именно верхушек и сучьев) рубок главного пользования и рубок промежуточного пользования просто остается в лесу. Это создает риск возникновения пожаров, а также приводит к потере теплотворной способности, которую можно было бы использовать для энергетических целей. В Беларуси в настоящее время растет спрос на древесную биомассу для комбинированного производства тепла и электроэнергии и систем централизованного теплоснабжения, а также со стороны промышленных и сельскохозяйственных предприятий по всей стране. Для удо-

влетворения этого Беларуси необходимо максимального задействовать все источники древесной биомассы.

Норматив образования отходов установлен по регионам с учетом естественного отпада и для учебно-опытного лесхоза составляет:

12,2% – сучья, ветви, вершинки на растущем дереве

9,6 – отпад при валке, трелевке, используемый на укрепление трелевочных волоков и далее в качестве удобрения

2,6% – сводный норматив лесосечных отходов, пригодных к использованию.

Суммарный объем отходов лесозаготовок по данным исследования составляет 3294 м³ в год. Чтобы определить объем условного топлива для производства топливной щепы из отходов лесозаготовок можно использовать коэффициент перевода м³ в т.у.т. Данный вид топлива берётся в использование, так как он наиболее приемлем для производства в учебно-опытном лесхозе. Для данного вида топлива коэффициент перевода составляет 0,22. Можно сделать вывод, что из годового запаса отходов лесозаготовок мы можем получить 724 т.у.т.

В ходе выполнения работы также были проведены расчёты по определению количества тепла, необходимого для нагрева воздуха в комнате на один градус (0,299 Ккал/м³*град). Скорректировав все расчёты по подаче и потере тепла, среднемесячное теплотребление моей комнаты на отопление и вентиляцию составит $24*180/6*0,0006732=0,443592$ Гкал/на комнату*месяц. На отопление всех комнат общежития (100 комнат) в месяц необходимо 44Гкал. Так как отопительный сезон составляет 6 месяцев, то $44*6=264$ Гкал в отопительный сезон. Используя расчеты, в которых рассчитан энергетический потенциал топливной щепы, потенциально производимой из отходов лесозаготовок в учебно-опытном лесхозе (он составляет 724 т.у.т. или 5062 Гкал), можно сделать вывод, что энергии, выделяемой при сгорании щепы, произведенной только лишь из отходов лесозаготовительного процесса, будет достаточно для отопления общежития №2 Полоцкого государственного лесного колледжа. Также следует отметить, что в расчет топливно-энергетического потенциала не включен запас низкокачественной древесины, запас дровяной древесины.

В результате проведения исследования, можно сделать следующие выводы:

- в Беларуси биоэнергетика начинает интенсивно развиваться в условиях необходимости достичь определенного уровня энергетической безопасности и в полном соответствии положениям Международного соглашения об изменении глобального климата, подписанного Республикой;

- в настоящее время отходы лесозаготовительной, деревообрабатывающей промышленности являются наиболее перспективными для использования их в виде биотоплива;
- древесное биотопливо является самой дешевой и значительной по объему использования формой аккумулирования возобновляемой энергии. За счет широкомасштабного использования биомассы как энергетического топлива эти объемы можно увеличить в несколько раз;
- себестоимость производства древесного топлива из отходов лесозаготовок может колебаться в широких пределах, основным фактором себестоимости топлива являются – вид рубки, технология заготовки и расстояние до котельной. Для планирования использования отходов лесозаготовок в масштабных проектах, необходимо производить всестороннюю оценку с учетом реалий региона, технологических и экономических фактов, и только после этого целенаправленно формировать спрос на данный вид топлива.

ЛИТЕРАТУРА

1. Никишов В.Д. Комплексное использование древесины. – М.: Леса промышленность, 1985.- 246 с
2. Федоренчик А.С., Мохов С.П., Технология и оборудование комплексного использования древесного сырья. Минск: БГТУ, 2014. – 274 с.
3. Гоманай М.В. Технология переработки древесины: Учебно-справочное пособие. –М. : МГУЛ, 2001. – 232 с.

УДК 614.84

Учащ. Д. С. Гаврилова, М. С. Гаврилова

Науч. рук. преп. Е. В. Башлыкова

(филиал БГТУ Полоцкий государственный лесной колледж)

ПРИМЕНЕНИЕ СОВРЕМЕННЫХ СПОСОБОВ ОБНАРУЖЕНИЯ ЛЕСНОГО ПОЖАРА НА ПРИМЕРЕ ФИЛИАЛА БГТУ «ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ ЛЕСНОЙ КОЛЛЕДЖ» УЧЕБНО-ОПЫТНЫЙ ЛЕСХОЗ

Лесной пожар – самый опасный и коварный враг леса. Повреждение лесов пожарами приводит снижению их продуктивности, уменьшению эстетических и рекреационных свойств, и что самое страшное, к полной гибели насаждения. В периоды высокой пожарной опасности лесные пожары становятся стихийным бедствием, наносят огромный ущерб народному хозяйству. Охрана и защита лесов, сохранение их для потомков, рациональное использование и повышение

продуктивности являются важной задачей не только государственной лесной охраны. Эти вопросы затрагивают интересы всего государства. Не смотря на знания тактики и способов тушения лесного пожара, наличия техники — проблема охраны лесов от пожаров еще не решена. В последнее время в нашей стране успешно обновляются технические средства по своевременному обнаружению лесных пожаров.

Цель работы: определение эффективности использования современных способов обнаружения лесного пожара (использование квадрокоптера) на базе филиала БГТУ «Полоцкий государственный лесной колледж» учебно-опытный лесхоз.

В процессе работы было изучено противопожарное обустройство лесной территории лесхоза, а также технические характеристики квадрокоптера.

В результате исследования - выделены преимущества использования квадрокоптера для своевременного обнаружения лесного пожара

Степень внедрения – используется в учебных целях как методический материал при изучении раздела «лесная пирология».

Сегодня у нас есть несколько способов обнаружения лесного пожара это:

- **Наземное патрулирование**
- **Наблюдение с пожарно-наблюдательных вышек**
- **Авиапатулирование**

Однако у каждого из этих способов есть недостатки, которые не позволяют говорить о своевременности обнаружения лесного пожара.

Наземное патрулирование: это

- ограниченная видимость;
- трудность выявления очагов возгорания, особенно при наличии подроста, подлеска и сложной формы насаждения;

Наблюдение с вышек: это

- невозможность тушения лесного пожара;
- обслуживание требует квалифицированных специалистов и ремонтной базы (при оснащении вышек видеокамерами);
- усложняется охрана вышек и оборудования от повреждений;
- и как результат возрастает стоимость выявления пожаров;

Авиапатулирование:

- наличие периодичности наблюдения за лесным массивом;
- зависимость от погодных условий;
- стоимость авиапатулирования;

На наш взгляд, решить все эти проблемы, и устранить недостатки, позволит использование современных средств обнаружения лесного пожара. В частности использование квадрокоптера.

Преимущества:

- Позволяет своевременно выявлять лесные пожары
- Нарушение правил пожарной безопасности
- Может быть использован для классификации и инвентаризации лесов
- Позволяет выявлять реальные и потенциальные проблемы леса (ветровалы, очаги массового размножения вредителей и заболеваний древесно-кустарниковой растительности)
- Помогает при лесовосстановлении (разработка интерактивных карт, демонстрирующих текущее состояние лесов)
- Может быть использован в охотничьем хозяйстве (выявление браконьерства)
- Имеет хорошие технические характеристики и легок в управлении.

Недостатки: единственным недостатком является стоимость дрона, которая колеблется от одной до двух с половиной тысяч долларов. Новозможности его использования позволят окупить его стоимость.

Свои исследования мы проводили на базе учебно-опытного лесхоза, где из способов обнаружения лесного пожара оставлено только наземное патрулирование и использование квадрокоптера. Такое решение действительно дало свои результаты. Это и своевременность обнаружения пожара, подтверждение пожара и снижение стоимости обнаружения лесного пожара.

Основной задачей своевременного обнаружения лесных пожаров является сведение лесных площадей, отданных пожару, до минимума, что и прослеживается в учебно-опытном лесхозе. Огромную роль в выполнении данной задачи играет применение современных способов обнаружения лесного пожара, в частности применение квадрокоптера.

ЛИТЕРАТУРА

1. Рихтер. И.Э. Лесная пирология с основами радиоэкологии, Мн. 2006
2. Щетинский В.А Охрана лесов, Мн.1994
3. Червонный М.Г Охрана лесов, Мн.1984
4. Симский А.М Охрана леса от пожаров, Мн.1975
5. Тех. паспорт квадрокоптера
6. Интернет ресурсы
7. Документы учебно-опытного лесхоза(характеристика предприятия, маршрут наземного патрулирования).

СОДЕРЖАНИЕ

СЕКЦИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

1	<i>Кунец Д. С., Гордеев И. В.</i> Программирование приложений дополненной реальности в среде Unity.....	5
2	<i>Чевжик Е.А.</i> Использование Google Firebase в Android-приложении.....	8
3	<i>Юранов В.В.</i> Развертывание сайта на Heroku.....	11
4	<i>Етко А. А.</i> Сервисы распознавания и синтеза речи платформы IOS.....	14
5	<i>Суша Г.В.</i> Особенности разработки веб-системы для продаж строительных компаний.....	18
6	<i>Бриль Д.С., Кузьмина С.А, Самасюк Д.И.</i> Создание web-ресурса «Ways to help», рассчитанного на благотворительную деятельность.....	20
7	<i>Бугаенко В. В., Карнеева Д. Д.</i> Информационные технологии iBooks Author для визуализации контента.....	24
8	<i>Клыбик Д. Л.</i> Применение технологии виртуализации для создания отказоустойчивого кластера.....	27
9	<i>Лаптев Н. А., Дубовик М. В.</i> Новый язык программирования Jai.....	31
10	<i>Богатырев Е.А.</i> Особенности разработки клиентской части информационно-поискового сервиса в сфере услуг.....	33
11	<i>Сакольчик А.А.</i> Особенности разработки серверной части информационно-поискового сервиса в сфере услуг.....	37
12	<i>Бриль В. С., Кузьмина С. А.</i> Разработка латентных узоров средствами растровой и векторной графики	41
13	<i>Ковалевская А. А.</i> Разработка дизайна лонгрида информационного сайта.....	44
14	<i>Зябко А.С.</i> Визуальное представление мультимедийного репортажа в новостных интернет – СМИ.....	47
15	<i>Бурая Е.В.</i> Разработка навигации сайта «GameStop».....	50
16	<i>Бекиш В.С.</i> Принципы разработки дизайна интерфейсов мобильного устройства.....	54
17	<i>Кулешова Ю.С.</i> Журнальный стиль в веб-дизайне.....	58
18	<i>Лосич Д.И.</i> Художественные средства композиции - пропорции и золотое сечение, их применение в дизайне интерфейсов.....	61
19	<i>Мельникова Т. А.</i> Организация обратной связи с аудиторией сетевого издания.....	65
20	<i>Сугако А. А.</i> Влияние цветовой композиции на эффективность пользовательского интерфейса.....	69
21	<i>Соловьева А.В.</i> Управление вниманием пользователя на основе стратегии эргономики цвета и контрастности.....	73
22	<i>Новиков М. А.</i> Использование математических закономерностей в проектировании интерфейсов.....	76

23	<i>Котик Е.С.</i> Использование Flat-дизайна при разработке интерфейса пользователя.....	80
24	<i>Чобот Н. Н.</i> Использование принципов эмоционального дизайна при проектировании пользовательского интерфейса сайта «Рюкзак».....	83
25	<i>Щербакова А. Н.</i> Интерактивная инфографика в современной интернет сфере.....	88
26	<i>Тонкович В.Ю.</i> Разновидность блогов. Разработка дизайна блога.....	92
27	<i>Лозюк Д.С.</i> Анимация пользовательского интерфейса...	96
28	<i>Хворост В.С.</i> Психологические аспекты взаимодействия человека и компьютера.....	100
29	<i>Ярошук А.В.</i> Программный модуль веб-портала университета «Личный кабинет студента».....	104
30	<i>Берников В.О.</i> Определение деформаций дорожных одежд нежёсткого типа с учётом реологических свойств материалов	107
31	<i>Фёдоров А.В.</i> Использование теории графов при решении инженерных задач.....	110
32	<i>Бусько У.Л.</i> Полезные советы проектировщикам интерфейсов.....	113
33	<i>Ткаченко А.Д.</i> Онбординг – первое свидание с пользователем.....	116
34	<i>Бердник К.С.</i> Ментальные модели как опора для дизайнерских решений	119
35	<i>Ревинская Ю.В., Бердник К.С.</i> Роль криптографии в обеспечении безопасности облачных вычислений.....	124
36	<i>Козловский А.А.</i> Программное средство для сбора, анализа и использования данных в информационной системе на основе светодиодных дисплеев.....	128
37	<i>Котик Е. С., Хворост В. С.</i> Криптовалюта биткоин.....	131
38	<i>Чобот Н. Н.</i> Алан Тьюринг и его система взлома машины «Энигма»...	134
39	<i>Лесняк И.А.</i> Анализ целостности текстовых файлов-стеганоконтейнеров с осажденной информацией на основе применения кернинга после структурных изменений стеганоконтейнера.....	138
40	<i>Панченко О.Л.</i> Анализ устойчивости процесса синхронизации двух нейронных сетей в задачах генерации криптографического ключа.....	141
41	<i>Сущенко А. А.</i> Стеганографическое преобразование текстов-контейнеров на основе языков разметки.....	145
42	<i>Чопик А. А.</i> Стеганографическая стойкость изображения-контейнера к объему осаждаемого текстового сообщения.....	149
43	<i>Щербакова А. Н.</i> Клод Шеннон – основоположник цифровых технологий и теории безопасности информационных процессов.....	152
44	<i>Короткая П. А.</i> Программное средство для учета товаров на предприятии.....	155
45	<i>Рак Н.В.</i> Интернет - опиум XXI века.....	158
46	<i>Деликатная М.М.</i> База данных «Студенты»	161

47	<i>Бабако А. В.</i> Обучающая программа для изучения файлового ввода/вывода на языке C/C++	164
48	<i>Сафончик И.Д.</i> Программное обеспечение управления европейской системой перевода и накопления учебных кредитов (ECTS) для IT специальностей.....	167
49	<i>Мартынюк А.А.</i> Сверточные коды для исправления пакетов ошибок при помехоустойчивой передаче информации.....	169
50	<i>Боровик А.А.</i> Применение технологий дополненной реальности в образовании.....	171
51	<i>Скородумова А.В.</i> Возможности геолокации на примере мобильного приложения.....	175
52	<i>Журавлёв В.Е.</i> ANDROID-приложение для перевода текста на иностранный язык.....	179
53	<i>Радиванович Д.А.</i> Приложение для фильтрации электронной корреспонденции.....	182
54	<i>Бубель А.С.</i> Алгоритм для нахождения всех возможных решений в сетевом графике, заданном в декартовой системе координат.....	186
55	<i>Сосновский А.А.</i> Программное средство Logic Blocks для операционной системы ANDROID.....	190
56	<i>Сенюк В.К.</i> Программное средство для редактирования и поиска кратчайших путей обхода графа.....	194
57	<i>Белькевич Р.И.</i> Обзор готовых решений для эмуляции и замещения бэкенда веб-приложений.....	196
58	<i>Зайцев А.Н.</i> Выбор оптимального алгоритма поиска подстроки в строке для различных видов данных.....	200
59	<i>Зайцев А.Н.</i> Алгоритмы препроцессинга изображений для улучшения качества оптического распознавания символов.....	204
60	<i>Булова А. Ф., Акшевская Е. И.</i> Bull milkman или как создать хороший прототип игры за 48 часов?.....	208
61	<i>Шенец П. С.</i> Моделирование движения тел под действием фона гравитонов.....	212
62	<i>Шавейко А.А.</i> Программное приложение «Развитие памяти».....	215
63	<i>Богданович Е.А.</i> Локализация консольных приложений.....	218
64	<i>Худницкая А. А.</i> Интернет-приложение для учета личных финансовых расходов.....	222
65	<i>Жаворонок П.Е.</i> Сервис бронирования билетов маршрутных транспортных средств «Marshrutka.by».....	226
66	<i>Трестьян А. Ю., Каргин Н. С.</i> Выбор и тестирование NoSQL баз данных.....	229
67	<i>Луцевич О.В.</i> Веб-приложение для реализации алгоритмов поиска на	234

	примере шахматной игры.....	
68	<i>Бурмакова А.В.</i> 3d-модель прогнозирования процесса загрязнения геологической среды в результате пролива нефтепродукта.....	237
69	<i>Врублевский Е.С.</i> Реализация задачи об оптимальном пути в сетевом графике в случае нефиксированных весов дуг на языке C++.	241
70	<i>Ровба А.В.</i> Предупреждение чрезвычайных ситуаций на водоемах технического назначения.....	244
71	<i>Люлькович Е.С.</i> Спектральные проявления атропоизомерии порфиринов в растворах.....	246
72	<i>Шиканов С.С.</i> Проводимость пленок оксида индия в присутствии адсорбированного кислорода.....	249
73	<i>Корнева В.А.</i> Анализ сцепления ZrC-Ni-УДА покрытий с твердосплавной основой.....	253
74	<i>Улитёнок А. О.</i> Исследование содержания дейтерия в воде в процессе выпаривания.....	257
75	<i>Березовская В. С.</i> Фрактальность кристаллов, дендритов и дендримеров.....	259
76	<i>Пашковский П. В.</i> Графическое представление молекул для анализа свойств веществ на примере бутана, пентана и их изомеров.....	262
77	<i>Ярошук А.А.</i> Оптимальное расположение растений при посадке.....	265
78	<i>Чистяков Н.В.</i> Аддитивные цепочки.....	272
79	<i>Радченко М. Ю., Карпач А. М.</i> Что такое граф? применение графов в программировании.....	276
80	<i>Хорхалёв В. В.</i> Эллиптические кривые и их приложения в криптографии.....	278
81	<i>Андреюк В.А., Баран Е.А.</i> История математических обозначений.....	281
82	<i>Синюк Д.И.</i> Об опыте реализации игры «Гарвард» в БГТУ.....	286
83	<i>Савось Н.Ю.</i> Проблема классов P и NP.....	292
84	<i>Шенец П.С.</i> Игра «Ним».....	295
85	<i>Зайцев А.Н.</i> Методы сглаживания изображений.....	298
86	<i>Байгазин Д.А.</i> Замечательные кривые.....	301
87	<i>Тарасенко Д. Д.</i> Красота многогранников в математике, искусстве и архитектуре.....	305
88	<i>Мирончик А.В.</i> Геолокация объектов внутри помещений.....	312

СЕКЦИЯ «ПЕРВЫЙ ШАГ В НАУКУ»

1	<i>Волова К. П.</i> Защита окружающей среды от вредного воздействия химических отходов промышленных предприятий города Бобруйска...	316
2	<i>Казакова К.П.</i> Лингвистический анализ названий улиц города Бобруйска.....	317
3	<i>М. А.Хурсан</i> Сродкі прыцягнення ўвагі чытачоў ў загалюўках газетных публікацый.....	319
4	<i>Гавриленко Д.</i> Новая концепция построения линии для автоматического выполнения индивидуальных заказов.....	320
5	<i>Тедер Е. А.</i> Как сохранить деньги во время кризиса.....	322
6	<i>Окрут А. О.</i> Выгодность применения скидок для производителя и потребителя.....	324
7	<i>Быковский З. С.</i> Способы защиты от неправомерного доступа к компьютерной информации в сети Интернет.....	326
8	<i>Трутнёв В. В., Гирдюк А. В., Корягин П. А., Прудников В. В.</i> Исследование свойств плит МДФ.....	329
9	<i>Попова В. А., Литвинюк В. А.</i> Эпоха Петровских реформ. Германизмы (штраф, абзац или красная строка).....	332
10	<i>Бабуль Н. В.</i> Ранний брак – проблема и актуальность современного общества.....	335
11	<i>Ковалёв К. В.</i> Энергосбережение в быту для всех и каждого.....	338
12	<i>Лашко А. М.</i> Модернизация цементной мельницы 3.2 X 15М с целью повышения производительности.....	341
13	<i>Рыжкович Е. С., Пузыревский А. Е.</i> Биотопливо. Миф или реальность? На примере учебно-опытного лесхоза филиала БГТУ Полоцкий государственный лесной колледж.....	345
14	<i>Гаврилова Д. С., Гаврилова М. С.</i> Применение современных способов обнаружения лесного пожара на примере филиала БГТУ «Полоцкий государственный лесной колледж» учебно-опытный лесхоз.....	348