

РАЗРАБОТКА ПРОГРАММЫ ДЛЯ АНАЛИЗА УСПЕВАЕМОСТИ СРЕДСТВАМИ РАЗЛИЧНЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Долгое время в нашем учреждении педагоги использовали для расчета параметров успеваемости простейшие средства: ручку, листок бумаги или калькулятор. Поэтому мы решили разработать простое приложение для автоматизации таких расчетов.

Объект моего исследования: Процесс создания программы для анализа успеваемости

В современном программировании существует много средств и языков подходящих для решения этой задачи. Какой из них выбрать? Каким должно быть мое приложение, чтобы соответствовать требованиям? Я решил, что стоит разработать несколько вариантов программы, проверить их на практике и, сравнив результаты, сделать выводы.

Цель работы: Разработать приложения для анализа успеваемости для различных платформ средствами нескольких языков программирования.

В первой очередь я проанализировал технологии и инструментов разработки приложений от истоков до наших дней. Сделал краткий обзор основных проблем разработки приложений.

Всего технологий у меня получилось 4:

- неструктурированное программирование
- процедурное и модульное программирование
- объектно-ориентированное программирование
- компонентные технологии

Среди современных средств разработки клиентских приложений выделил два класса:

- нативные инструменты (предоставляются фирмами разработчиками операционных систем)
- кроссплатформенные (позволяют создавать приложения легко перемещаемые между различными устройствами)

Работу над программой я решил начать с изучения запросов и возможностей предполагаемых пользователей. Пользователями приложения стали учителя разного возраста, с разным уровнем

компьютерной грамотности. В ходе опроса были выделены **обязательные требования** к программе:

- простота установки, запуска и использования программы;
- после запуска программа должна запрашивать количество учащихся конкретного класса и количество выставленных отметок по баллам;
- программа должна контролировать правильность ввода;
- программа должна работать корректно.

И дополнительные пожелания:

- программа должна быть доступна на любом домашнем, в том числе мобильном, устройстве;
- возможность сохранения полученных результатов;
- возможность дальнейшей доработки программы.

Мы изучили, какие похожие программы имеются в сети. Оказалось, что в Интернете есть готовые приложения такого рода, однако многие из них либо ведут расчеты в пятибалльной системе, либо являются платными, либо требуют предварительного ввода отметок всех учащихся за весь период обучения. Изучались возможности платформ Znaj.by и Schools.by. Однако они требуют выхода в Интернет, а это для большинства наших учителей не всегда возможно. Поэтому, сначала мы решили создать простое приложение, которое сможет работать без подключения к сети, а позже заняться его доработкой для других платформ

Первой была создана версия программы на языке PascalABC. Ввод исходных данных в ней осуществлялся с клавиатуры, а вывод на экран. Для пользователей был разработан простейший текстовый интерфейс. Программа осуществляет проверку корректности ввода. Программа использовалась для анализа успеваемости более года. Однако ее недостатки вынудили перейти к разработке десктопного приложения для ОС Windows.

В качестве средства для создания второй версии программы, был выбран язык Python. Дистрибутив программы реализован в виде exe-файла. Само приложение выполнено в виде отдельного программного окна с фиксированными размерами и строкой заголовка. Визуальный интерфейс программы представляет собой отдельно расположенные зоны, где в соответствующих формах вводятся с клавиатуры общее количество учащихся и отметки по баллам. В случае отсутствия отметок какого-либо вида вместо них автоматически устанавливается значение 0. Переход к вычислению показателей успеваемости осуществляется кнопкой «Рассчитать» в нижней части окна программы.

Кроссплатформенный вариант приложения, который можно использовать не только на компьютере, но и на любом мобильном устройстве, поскольку для его работы необходим лишь браузер, было решено создать в виде веб-сайта, размещенного на GitHub – крупнейшем бесплатном веб-сервисе для хостинга IT-проектов и их совместной разработки. Использовался язык HTML5 с элементами CSS 3.0 – каскадной таблицы стилей. Для реализации самой программы расчета и переходов к сторонним ресурсам использовался язык JavaScript 1.8.5 и фреймворк jQuery 3.5.1. Отладка велась вручную через консоль браузера (F12). Приложение размещено по адресу <http://teachelper.ml>

На заключительном этапе для всех версий программы было проверено соответствие исходному заданию, а также выполнена проверка на отсутствие ошибок и проведено тестирование на реальных входных данных.

В ходе тестирования выявлено, что все варианты приложения соответствуют минимальным требованиям и корректно выполняют расчеты на массивах реальных данных. Исключение составила программа на языке PascalABC, которая выдала сбой в работе на больших данных и не всегда выдавала оповещение о некорректном вводе в нужном формате. В ходе сравнительного анализа версий приложения установлено, что менее всего соответствовал запросам пользователей вариант программы на языке PascalABC. Среди серьезных недостатков программы – необходимость установки среды программирования Pascal ABC на компьютер пользователя и ввод нулевых значений. Более всего удовлетворяла требованиям пользователей веб-версия приложения, которая и была выбрана для дальнейшего развития.

Для дальнейшей разработки по результатам исследования выбрана веб-версия, созданная средствами HTML и JavaScript.

Практическая значимость: предложенные в работе приложения позволяют автоматизировать и упростить процесс расчета основных показателей успеваемости, они были внедрены в учебный процесс и помогли значительно сократить время подготовки аналитических отчетов.

ЛИТЕРАТУРА

1. Вирт Н. Алгоритмы и структуры данных / Пер. с англ. — М: ДМК Пресс, 2011. — 276 с.: ил.
2. Роджерс Д., Адамс Дж. Математические основы машинной графики / Роджерс Д., Адамс Дж.; Пер. с англ. - М.; Мир, 2001. - 604 с.

3. Хайлгесс А., Конвэй Д. Программирование под iOS. Для профессионалов / Пер. с англ. — СПб: Питер, 2013. — 608 с.: ил.

4. Клифтон Я. Проектирование пользовательского интерфейса в Android / Пер. с англ. — М: ДМК Пресс, 2017. — 452 с.: ил.

8. Мартин Р., Ньюкирк Д., Косс Р. Быстрая разработка программ: принципы, примеры, практика / Пер. с англ. — М.: Вильямс, 2004. — 752 с.: ил.

УДК 004.932.72'1

Учащ. А. В. Петровский

Науч. рук. С. С. Лещук, учитель информатики
(ГУО «Средняя школа №23 г.Гродно»)

РАСПОЗНАНИЕ ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ

Теория распознавания образа – раздел информатики и смежных дисциплин, развивающий основы и методы классификации и идентификации предметов, явлений, процессов, сигналов, ситуаций и т. п. объектов, которые характеризуются конечным набором некоторых свойств и признаков. Необходимость в таком распознавании возникает в самых разных областях – от военного дела и систем безопасности до оцифровки аналоговых сигналов.

Проблема моего исследования заключается в разработке программы для создания и разработки собственного приложения для распознавания объектов в изображениях.

Моя разработанная программа представляет из себя нейросеть, которая распознаёт на изображениях объекты (человек, машина и т.п.) и обводит их в квадрат, подписывая возле него название объекта. Программа написана на языке Python с использованием библиотеки TensorFlow (она создаёт каркас нейросети), Keras (она позволяет автоматизировать многие функции в TensorFlow) и OpenCV(она позволяет работать с изображениями в TensorFlow). Распознавание изображения относится к задаче ввода изображения в нейронную сеть и присвоения какой-либо метки для этого изображения. Метка, которую выводит сеть, будет соответствовать заранее определенному классу. Может быть присвоено как сразу несколько классов, так и только один. Если существует всего только один класс, обычно применяется термин «распознавание», тогда как задача распознавания нескольких классов часто называется «классификацией».

Классическая видео аналитика не справляется с поставленными задачами: число видеокамер постоянно растет, количество поступающих от камер событий тоже многократно увеличивается. Это приводит к большому количеству ложных срабатываний — оператор