

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Лесосибирский педагогический институт –
филиал Сибирского федерального университета

Кафедра высшей математики, информатики и естествознания

УТВЕРЖДАЮ

Заведующий кафедрой

Н.Ф. Романцова

подпись

« 8 » июня 2018 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 Информационные системы и технологии

Разработка клиент-сервера электронного расписания

Руководитель *М 07.06.18* доцент, канд. филос. наук М.В. Румянцев
подпись, дата

Выпускник *Васильев 07.06.18* В.В. Васильев
подпись, дата

Лесосибирск 2018

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка клиент-сервера электронного расписания» содержит 41 страницу текстового документа, 11 рисунков, список использованных источников, включающий 40 наименований.

РАСПИСАНИЕ, ПРОБЛЕМЫ РАЗРАБОТКИ ЭЛЕКТРОННОГО РАСПИСАНИЯ, MICROSOFT SQL SERVER 2017, MICROSOFT VISUAL STUDIO 2017, C#, МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, ANDROID.

Каждый из нас сталкивался с планированием каких-либо действий. Расписание – это график, содержащий сведения о времени, месте и последовательности действий. Составление расписания для одного человека (распорядок дня) не является сложной задачей, так как на составление данного расписания влияет наименьшее количество условий, которые надо учесть. В организации же совсем иная ситуация. Здесь уже надо учитывать рабочий график множества людей и их обязанности. Но, наиболее сложный процесс составления расписания падает на долю организаций, связанных с образованием. В данной ситуации на помощь приходят информационные технологии.

Цель данной работы – разработка комплекса программ для составления и запроса электронного расписания.

Задачи исследования:

- проанализировать учебную и научную литературу по теме выпускной квалификационной работы;
- рассмотреть современные технологии программирования для разработки электронного расписания;
- разработать информационную модель клиент-сервера «электронное расписание»;
- разработать комплекс программ «Электронное расписание».

В ходе проведения работы был проведен анализ научной и учебно-методической литературы, проанализированы примерные программы по теме «электронное расписание».

В итоге было разработан комплекс программ «электронное расписание»:

- настольное приложение для ввода данных;
- настольное приложение для создания, редактирования, сохранения, загрузки и импорта электронного расписания;
- мобильное приложение для вывода электронного расписания на мобильные устройства с операционной системой Android.

СОДЕРЖАНИЕ

Введение.....	5
1 Теоретические основы разработки электронного расписания.....	7
1.1 Проблемы создания электронного расписания.....	7
1.2 Рынок программного обеспечения для составления учебного расписания.....	11
1.3 Модель программного обеспечения.....	14
1.4 Технологии программирования для разработки приложений.....	17
1.4.1 Delphi.....	17
1.4.2 Java.....	18
1.4.3 Swift.....	19
1.4.4 C#.....	20
1.4.5 Python.....	21
1.5 Мобильные операционные системы.....	22
1.5.1 Операционная система Windows phone.....	22
1.5.2 Операционная система iOS.....	23
1.5.3 Операционная система Android.....	24
2 Практическая реализация электронного расписания.....	31
2.1 Информационная модель клиент-сервера «электронное расписание».....	31
2.2 Разработка программного продукта.....	32
2.2.1 Среда разработки.....	32
2.2.2 Программа №1 Ввод данных.....	33
2.2.3 Программа №2 создания расписания.....	34
2.2.4 Программа №3 мобильное приложение.....	36
Заключение.....	37
Список использованных источников.....	38

ВВЕДЕНИЕ

Каждый из нас сталкивался с планированием каких-либо действий. Расписание – это график, содержащий сведения о времени, месте и последовательности действий. Составление расписания для одного человека (распорядок дня) не является сложной задачей, так как на составление данного расписания влияет наименьшее количество условий, которые надо учесть. В организации совсем иная ситуация, так как здесь необходимо учитывать рабочий график множества людей и их обязанности. Но наиболее сложный процесс составления расписания, приходится на долю образовательных организаций. При составлении расписания в образовательных организациях необходимо учитывать следующие условия:

- ненормированный рабочий день преподавателей;
- преподаватели могут работать в нескольких учебных заведениях;
- наличие необходимого оборудования в кабинетах для занятий;
- наполняемость кабинетов;
- грамотное распределение занятий в течение учебного года;
- соответствие учебному плану.

Данные требования могут меняться в зависимости от организации. Представьте что сотрудник, отвечающий за расписание, пытается это все учесть и записать, а по мере составления расписания условия изо дня в день могут меняться.

В данной ситуации на помощь приходят информационные технологии.

Информационные технологии (ИТ) – это совокупность методов и средств, используемых для сбора, хранения, обработки и распространения информации [19].

На сегодняшний день существует немало специалистов, активно использующих языки программирования. Поэтому, когда возникает вопрос, на чем писать программу можно встретить много мнений. Ситуация очень схожая с вопросами о том какую музыку послушать или какой фильм посмотреть – сколько людей столько и мнений. Проанализировав программное обеспечение

(ПО), мы выяснили, что большинство приложений написано на языке C#. Это повлияло на выбор языка программирования в программной среде Microsoft Visual studio 2017 на языке C# с использованием сервера базы данных Microsoft SQL server 2017 для разработки комплекса программ «Электронное расписание».

Решение этой проблемы имеет теоретическое и практическое значение.

Объект исследования – технологии программирования.

Предмет исследования – среда программирования Microsoft Visual Studio 2017 для разработки приложения на операционных системах Windows и Android.

Цель исследования – разработка комплекса программ, для составления и запроса электронного расписания.

Задачи:

- проанализировать учебную и научную литературу по теме выпускной квалификационной работы;
- рассмотреть современные технологии программирования для разработки электронного расписания;
- разработать информационную модель клиент-сервера «электронное расписание»;
- разработать комплекс программ «электронное расписание».

1 Теоретические основы разработки электронного расписания

1.1 Проблемы создания электронного расписания

Первые записи о применении техники для упрощения или автоматизации процесса составления расписания появились еще в 60-х годах. Таким образом, история данного вопроса имеет большую историю. За более чем 55 лет исследования в данной сфере была проделано огромное количество интеллектуальных процессов огромного количества специалистов со всех уголков планеты.

С каждым годом вычислительная техника совершенствуется, поэтому в этом нет ничего удивительного, что и процесс создания электронного расписания совершенствуется.

Поначалу из-за труднорешаемости задачи и не доступности вычислительной техники, слабый интерес к данному вопросу если и был, то только у представителей высших учебных заведений. Хотя и они впоследствии ушли в сторону разработки программного обеспечения для учета и контроля успеваемости учащихся.

Бурный рост производительности и падение цен в 90-х на персональные компьютеры (ПК) позволили оснастить огромное количество не только институтов, но и школьных заведений вычислительной техникой. Именно в эти года снова заговорили о уже почти забытом на тот момент вопросе об электронном расписании. Стоит так же отметить, что именно в этот период стало появляться огромное количество начинающих разработчиков программного обеспечения, которые брались за любой вопрос автоматизации процесса. Кто-то это делал для саморазвития, а кто-то в поисках легкой наживой обещая автоматизацию абсолютно всех процессов.

Вначале XXI века индустрия ПК прекратило бурно развиваться, и перешла в стадию «стабильности». Окончательно занятии лидирующую позицию операционные системы (ОС) с графическим интерфейсом вытеснив

устаревший MS-DOS. Вместе с уходом MS-DOS «умерли» и его программы для составления электронного расписания. Появилась необходимость новых программ, которые будут активно использовать графический интерфейс для упрощения процесса.

И действительно, в самом конце прошлого столетия огромное количество производителей, ещё раз попробовало, но уже, как им казалось, совершенно в новом техническом и технологическом уровне взялись за разработку программного обеспечения для составления электронного расписания. На фоне прекращения заметного роста производительности персональных компьютеров, стабилизации идей в области программного обеспечения переросли в программы нового поколения. Основной особенностью этих программ является то, что они учитывали как ошибки, так и оригинальные идеи предыдущих. Здесь в первую очередь имеются в виду разработчики девяностых. С математическими результатами шестидесятых, семидесятых и восьмидесятых все просто – если ты о них знаешь, то используешь, если нет, то «открываешь Америку заново». Не для кого не секрет, что графический интерфейс предоставляет разработчикам принципиально намного больше возможностей по сравнению с псевдографическим (текстовым). Если мы начнем анализировать на рынке программы для составления электронного расписания и попытаемся их сравнить, то к своему удивлению встретим совершенно огромное и потрясающее разнообразие способов формирования и ввода исходных данных, необходимых для расчета и составления расписания, хотя если взглянуть на данный вопрос с математической точки зрения все программы делают одно и то же. Таким образом, одно из главных качеств для программ составления электронного расписания стало логичность и удобство пользовательского интерфейса.

На сегодняшний 2018 год стоит отметить, что по сравнению с программами девяностых, программы нынешнего поколения значительно «поумнели». Оптимизм у разработчиков заметно угас. Обещать полную автоматизацию всего, что попало под руку, уже почти никто не берется.

Почти все проекты, начатые в конце девяностых к настоящему времени уже прекратили свое существование в силу их не актуальности. Другие же продолжают развиваться и совершенствоваться. Третьи на протяжении более чем десяти лет застыли и прекратили своё развитие. Но об окончательном и бесповоротном решении задачи составления электронного расписания сегодня говорить все еще рано.

Обычно когда заходит речь о пользе использования программного обеспечения по составлению расписания, указывают такой фактор, как – на сокращение времени сотрудника при составлении учебного расписания. Ещё один часто указываемый фактор, что расписание с помощью данной программы может быть лучшего качества. Хотя лично я с этим фактором не согласен, так как неважно за какой программой будет работать сотрудник – голова у него от этого мыслить, по-другому не станет. Данный фактор работает только в случае полной автоматизации процесса. При автоматизации процесса исключаются субъективные оценки и личная заинтересованность сотрудника по отношению к преподавателю (учителю), при составлении расписания, в том числе и при распределении педагогической нагрузки. С другой стороны, это позволит исключить неприязнь в адрес сотрудника со стороны преподавателей, поскольку компьютер «лицо не заинтересованное». Тем самым, расчет распределения педагогической нагрузки и расписания на компьютере, может соблюсти принципы справедливости и равноправия. Но принять полную автоматизацию готовы далеко не все учебные заведения.

Если говорить о сложности решения задачи составления электронного расписания то квалифицированные пользователи персонального компьютера, кажется, что задача составления электронного расписания ничуть не сложнее создания, например, качественного видео редактора или аудио редактора. Однако число исследователей, так или иначе изучавших эту задачу, трудно поддается подсчету. Среди них десятки докторов технических и физико-математических наук, сотни кандидатов наук, не только технических, не говоря уже о тысячах любителей решить математические задачки, а так же не надо

забывать огромную армию студентов технического и физико-математического профиля обучения. Чаще всего среди исследователей задачи составления электронного расписания для учебных заведений упоминают двух академиков – Танаева В. С. (белорусский математик, директор НИО «Кибернетика» НАН РБ) и Михалевича В. С. (украинский математик и кибернетик, академик АН Украины, академик РАН).

И, тем не менее, не смотря на колоссальные усилия исследователей, говорить о полном и удовлетворительном решении задачи составления учебного расписания – не приходится

Зададим себе вопрос:

Имеющиеся в настоящее время на рынке программы являются беспомощными для упрощения или автоматизации процесса составления электронного расписания для учебных заведений?

Касательно упрощения процесса можно смело ответить, что можно найти на рынке программу, которая позволит решить данную проблему. Но тут есть маленькая оговорка. До сих пор есть (и не мало) людей которые составляют расписание «в ручную» на бумажке, для таких людей тот же EXCEL будет упрощением процесса.

По поводу автоматизации процесса можно ответить только при условии, полного анализа программы которое будет занимать не менее 15 – 30 часов. При этом количество проверенных программ должно быть более трех, и они должны быть успешные и представительные. Суммировав это мы должны посвятить более 100 часов своего личного времени изучению этих ПО. После всех этих вычислений возникает другой вопрос: – «Где взять столько свободного времени или человека, который согласится заниматься тяжелым (в каком-то смысле бессмысленным) трудом?»

1.2 Рынок программного обеспечения для составления учебного расписания

Рынок ПО составления учебных расписаний, который складывается вместе с рынком любого ПО для персональных компьютеров, представляется просто уникальным, или же по крайней мере удивительным, на худой конец весьма странным. Так в чем же его уникальность или странность?

Когда-нибудь приходилось видеть такую рекламу: – «Купите наш пылесос, который не может всасывать пыль». Или такую: – «Все кастрюли которые мы вам можем предложить – дырявые». Или такую: – «Купите нашу программу для составления расписания, которая не может его составить, но зато может составлять».

Так сколько же стоит пылесос, который не всасывает пыль, дырявая кастрюля или телевизор, который никогда, нечего не показывает? Прежде чем ответить на этот нелегкий вопрос попробуем оценить количество потенциальных покупателей и сравнить его с количеством учебных заведений, которые уже совершили свою покупку.

Демографами установлено, что около 16% населения развитых стран являются школьниками. Именно эта цифра используется при строительстве новых школ в новых районах застройки [27]. Далее будем производить арифметические вычисления на примере Российской Федерации. Население составляет примерно 140 млн. человек. Таким образом, школьников примерно 22 млн. Школ около 50 тыс. Значит среднее количество учеников в школе 440 человек. Но это среднее количество. Известно (!!!16), что за последние 60 – 70 лет типовыми проектами школ, считались школы на 1000 – 1400 учеников. Отсюда вывод – имеется огромное количество школ с количеством учеников на много меньшим, чем наша средняя цифра – 440 человек. Очевидно, что это школы в сельской местности или в очень не больших городах. Отсюда же, более сильный вывод – огромному количеству школ, программы для составления расписания учебных занятий не нужны в принципе.

Оценить количество школ, которым такие программы не нужны в принципе, конечно же, весьма трудно. Тем не менее, внимательно посмотрев на потолок, увидим там цифру – 70%. Из чего следует, что 30% школ имеют количество учеников от 500 и более, а таким школам программа, которая, не может составлять школьное расписание, а может его составить, не помешала бы. Получаем окончательную цифру – 15 тыс. школ. Вот это, пожалуй, потенциальная емкость рынка для Российской Федерации.

Поставим такой вопрос. Имеются ли на рынке программы, которые могут оказать хоть какую-то помощь при составлении расписания? Например, многие в ручную составляют расписание в два этапа. На первом этапе «Разбираются с иностранцами». Другими словами, составляют расписание для преподавателей и классов при изучении иностранного языка. Вторым этапом идет все остальное. По крайней мере, две программы, представленные на рынке, с этим, первым этапом, справляются на зависть прекрасно. Здесь же можно спланировать время проведения элективных курсов. При этом расставляется от 10 до 40 процентов занятий. Так что, кое-какая польза от применения компьютера, оснащенного этими программами, конечно же, есть. Причем одна из этих программ, весьма агрессивно и настойчиво пытается достроить расписание. В некоторых случаях, пускай и редких, это ей удается. Другая же, при достраивании расписания, абсолютно беспомощна.

Так сколько же народу на сегодня пользуется ПО для составления расписания учебных занятий в Российской Федерации? Некоторые производители такого ПО, на своих сайтах публикуют информацию о своих клиентах. Правда, к этой информации стоит относиться весьма осторожно. Некоторые производители в «припадках маркетинга» идут на весьма бесхитростный обман потенциальных клиентов. И тем не менее, отделив зерна от плевел получим цифру – порядка 1500 школ. Что составляет около 10% емкости потенциального рынка. Следовательно, 90% - это потенциальные клиенты.

Теперь обратим свой взор на мировой рынок. Как следует из предыдущих вычислений, весьма удобным способом расчета количества потенциальных клиентов, является такой способ. Берем население страны, отбрасываем четыре нуля, получаем количество потенциальных клиентов. Так и поступим:

Европа – 500 млн. человек.

США – 300 млн. человек.

Канада – 30 млн. человек.

Япония – 125 млн. человек.

Австралия – 20 млн. человек.

Другие развитые страны – 25 млн. человек.

Отбрасываем четыре нуля. Получаем – 100 тыс. потенциальных клиентов.

Теперь вопрос: – «Сколько же учебных заведений из этого списка используют ПО для составления расписания?». Применяем ту же методику, отделяя зерна от плевел, что и для Российской Федерации. Получаем цифру – порядка 30-ти тыс. Что составляет 30% рынка. При этом 70% открыты для агрессивного маркетинга.

Теперь остается перевести количество в качество. То есть, умножить количество потенциальных клиентов на цену одной лицензии ПО. Другими словами, оценить емкость мирового рынка в американских рублях. Но для этого нужно знать цену такой лицензии.

На самом деле, формула очень проста. Программное обеспечение, какой бы сложности и объема оно ни было, стоит ровно столько, сколько за него платит клиент (пользователь). Ярчайшим примером тому, является операционная система Windows от Microsoft. Наверное, мало кто задумывался о том, что по количеству труда, таланта, знаний и т.п., высадка человека на Луну, по сравнению с этой операционной системой, является детскими шалостями. И тем не менее, заплатите сто пятьдесят долларов США на бочку, и вы легальный пользователь.

Проблема только в том, что количество потенциальных клиентов – пользователей операционной системы и программы для составления школьного

расписания не сопоставимо, ни в первом, ни во втором приближении. Отсюда вывод: – «Не смотря на то, что некоторые просят по 15 долларов за дырявые кастрюли, программа, которая действительно могла бы решить большинство проблем, должна стоить дорого». Остается только ответить на вопрос: – «Что такое дорого?»

Конечно же, у каждого представления о «Дорого» свои. Но наверное, для сотрудника составляющего расписание (или аналогичной должности, если мы говорим о мировом рынке), дорого это его месячная зарплата. То есть, от 1000 до 5000 долларов США. Что собственно, мы и наблюдаем, или по крайней мере ранее наблюдали, в реальности. Первое время, на мировом рынке эти программы именно столько и стоили. Падение цен, как нам представляется, произошло именно из-за того, что вдруг обнаружилось – за 5000 долларов была куплена дырявая кастрюля. Теперь, те же дырявые кастрюли, предлагают по 500 долларов США. И наконец, перемножив количество на цену, получим примерную емкость мирового рынка ПО для составления школьного расписания – от 100 до 500 млн. долларов США. То есть, рынок не менее денежно-емкий, чем, скажем, рынок различных систем автоматизированного проектирования в промышленности и строительстве. И кстати говоря, не менее наукоемкий.

1.3 Модель программного обеспечения

Прежде всего, нужно понять, что консольное приложение, никогда не сможет убедить в том, что действительно решается задача составления расписания. Разве что, самого программиста, который написал это приложение. Создать черное окошко, с время от времени появляющимися там строчками типа: – «количество решенных уравнений = 1458» под силу даже школьнику. Таким образом, нормальный человек, просто не поверит такой программе. Обойтись без полноценного win-приложения, никак не получится. Но, сначала

неплохо было бы, определиться с целями создания такого приложения. По крайней мере, на виду две таких цели. Это – создание полноценного ПО со всеми вытекающими из этого последствиями, и – создание приложения демонстрирующего работу алгоритма, которое хуже или лучше сможет убедить человека в том, что его не обманывают. Понятно, что по трудоемкости эти две цели просто не сопоставимы. Вполне естественно, что было принято решение пойти по легкому пути. Что требуется от такого win-приложения? Прежде даже можно поставить другой вопрос: – «Какое оно должно быть?».

Во-первых. Сразу же снимается головная боль об удобном, понятном, практичном и красивом пользовательском интерфейсе. Для такой демонстрации вполне достаточно примитивного интерфейса. Важно лишь то, что бы пользователю были видны те исходные данные, которые предлагаются программе для расчета и результаты этого расчета. По крайней мере, теоретически у пользователя будет возможность проверить соответствие исходных данных и полученного с помощью программы результата. Сложна ли такая проверка? Естественно не просто. Особенно если знать какое количество проверок содержится в консольном приложении для постоянной верификации получаемых результатов, а так же размер кода этих проверок.

Во-вторых. Снимается проблема файла помощи, руководства пользователя, необходимых для полноценного программного обеспечения. На главную форму приложения налепили двадцать с лишним кнопок, из которых на каждом этапе расчета активна только одна, не считая кнопок типа – «О программе», «Начать новую задачу», «Заккрыть». Нажимаешь на такую кнопку, появляется окошко с кнопкой “Генерация данных”. Нажимаешь «Генерация данных», в окошке на белом фоне появляются построенные данные. Закрываем окошко. Кнопка, на которую только что нажали, перестает быть активной, следующая на которую следует нажать, становится активной. Жмем. Открывается следующее окошко. А там кнопка «Построить расписание». Нажимаем на «Построить расписание», появляется построенное расписание. Проверяем – правильно построено расписание или нет. И так до тех пор, пока

не будут пройдены все шаги алгоритма. А дальше можно нажать на большую кнопку «Начать новую задачу». И так по кругу. Или нажать кнопку «Заккрыть». На первый взгляд может показаться: – «Вся эта демонстрационная программа – мартышкин труд». Но это не так. По крайней мере, по трем причинам.

Во-первых. В ходе разработки демонстрации была решена довольно важная задача разработки будущей архитектуры полноценного ПО. А именно. Требовалось жесточайшим образом отделить «мозги» от «туловища». Если выражаться более понятно, отделить код алгоритма расчета расписания от кода генератора исходных данных и кода пользовательского интерфейса. Весь код алгоритма расчета расписания сосредоточен в динамически подключаемой библиотеке, таким образом, пользовательский интерфейс, как клиент, может обращаться с заданиями к динамической библиотеке, которая выступает в качестве сервера, на построение различных расписаний, составляемых на различных шагах алгоритма. Это позволяет, не трогая кода алгоритма расчета расписания, проводить эксперименты с различными вариантами интерфейса до полного и окончательного удовлетворения пользователей.

Во-вторых. Не смотря на свою примитивность, демонстрационный пользовательский интерфейс представляет из себя логическую модель удобного, понятного, практичного и красивого интерфейса пользователя. Так, например, в нем реализована возможность возврата к предыдущему шагу алгоритма, а эта возможность, в свою очередь повлияла на структуру данных программы. Кроме того, демонстрационный интерфейс поддерживает такую особенность алгоритма, как продвижение от шага к шагу в строгой последовательности, что обеспечивает целостность данных и их защиту от неверных изменений.

В-третьих. Пользовательский интерфейс пригоден для анализа математической модели практических ситуаций, которые возникают при составлении расписания, принятой в программе. Такой анализ или экспертизу могли бы осуществить специалисты хорошо знакомые с темой, например, завучи с достаточным опытом работы, преподающие в школе математику.

Разобраться в деталях расчета, конечно же их квалификации не достаточно, зато в силу полученной общей математической культуры, разглядеть явные упущения в постановке задачи они могут гораздо лучше любого профессионального математика, который знаком с работой школы только понаслышке или по различного рода публикациям.

«И что же дальше?». А дальше разработка полноценного ПО по всем законам и правилам программной инженерии, которое теперь уже, по сложности, не превосходит обычное ПО для ERP-систем.

1.4 Технологии программирования для разработки приложений

Рассмотрим наиболее популярные технологии программирования для разработки современных мобильных приложений.

1.4.1 Delphi

Delphi – объектно-ориентированный язык программирования со строгой статистической типизацией переменных. На сегодняшний день на языке Delphi можно писать приложения как для настольных ОС (Windows, Linux, macOS), так и для мобильных (Android, iOS, Windows Phone).

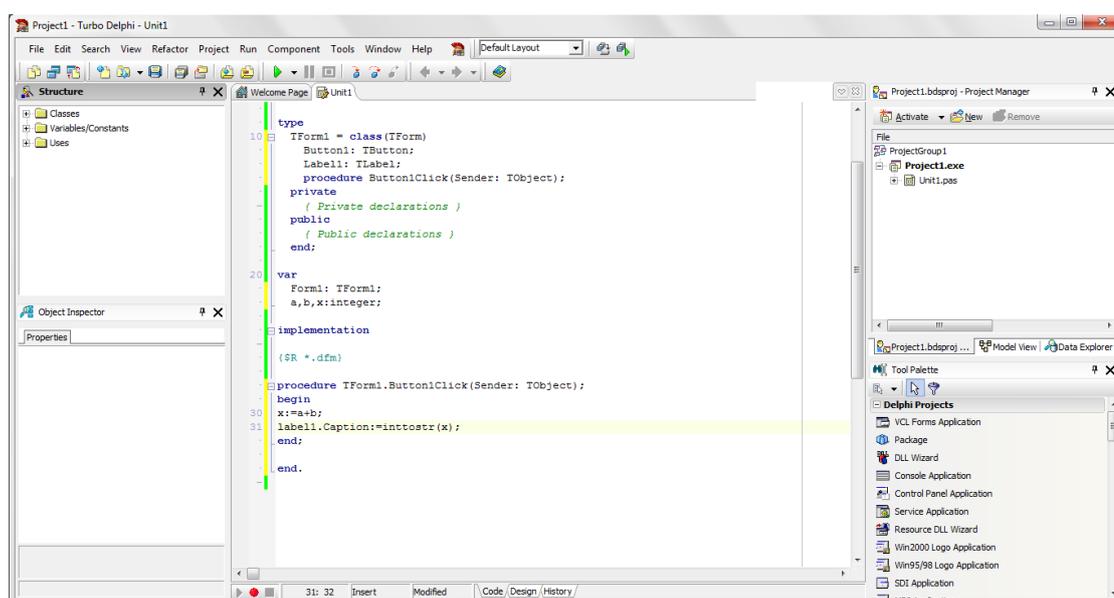


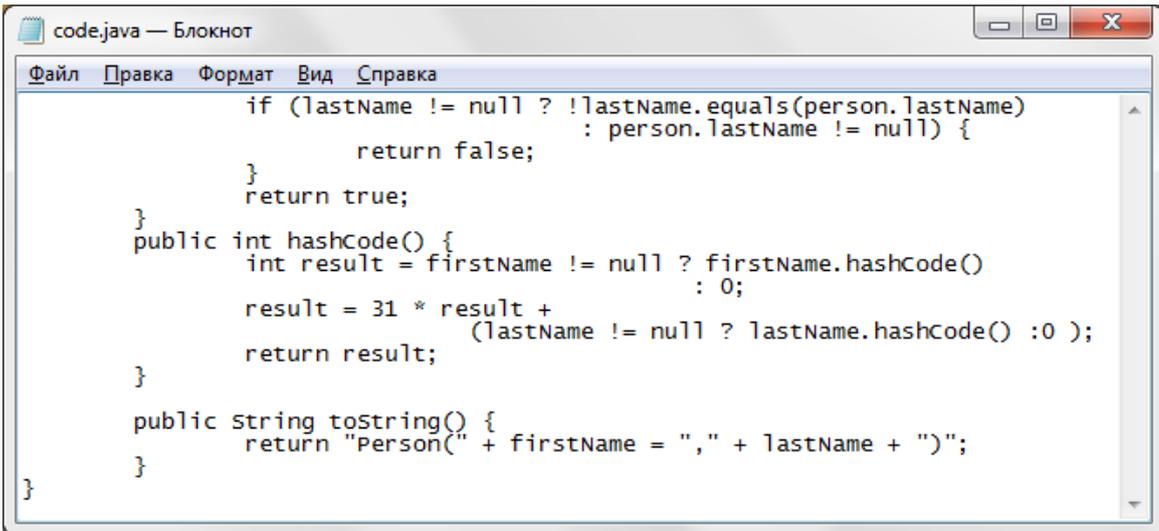
Рисунок 1 – Код Delphi в программе Turbo Delphi

Мобильная среда Delphi подходит для решения простых задач внутри своей среды. Если программист решит начать взаимодействовать с мобильной ОС, он столкнется либо со сложной задачей, либо с невыполнимой задачей. Обходят эту проблему написания функций, которые должны взаимодействовать с интерфейсом или внутренней средой ОС, на другом языке с которыми могут сосуществовать язык программирования Delphi [34].

Данная ОС больше всего подходит для начинающих программистов мобильного приложения, для того что бы они могли посмотреть разницу разработки мобильных приложений от обычных для ПК. Но так как активное использования данного языка в индустрии ПО давно прекратилось в дальнейшем рекомендуется перейти на более другие языки.

1.4.2 Java

Java – строго типизированный объектно-ориентированный язык программирования, принадлежащий компании Oracle.



```
code.java — Блокнот
Файл  Правка  Формат  Вид  Справка
    if (lastName != null ? !lastName.equals(person.lastName)
        : person.lastName != null) {
        return false;
    }
    return true;
}
public int hashCode() {
    int result = firstName != null ? firstName.hashCode()
        : 0;
    result = 31 * result +
        (lastName != null ? lastName.hashCode() : 0);
    return result;
}
public String toString() {
    return "Person(" + firstName + "," + lastName + ")";
}
}
```

Рисунок 2 – Код Java в программе Блокнот

Одной из главной особенности Java является его универсальность. Java может работать как и обычное приложение, так и непосредственно в окне

браузера. Данная особенность подходит, когда код программы нужно адаптировать к разным версиям ОС. Особенностью так же является тем что для написания когда на языке Java не нужны особые программы, можно просто записать код в блокнот и сохранить в формате *.java, а затем в программе сделать ссылку на этот файл кода и он будет выполняться [25].

У Java есть отдельный сценарный язык программирования под названием JavaScript. Данный сценарный язык программирования не является самостоятельным, а идет дополнением для решение определенных задач в других объектно-ориентированных языках программирования [25].

1.4.3 Swift

Swift – это мультипарадигмальный компилируемый язык программирования общего назначения. Создан компанией Apple для разработки приложений на операционных системах macOS и iOS.

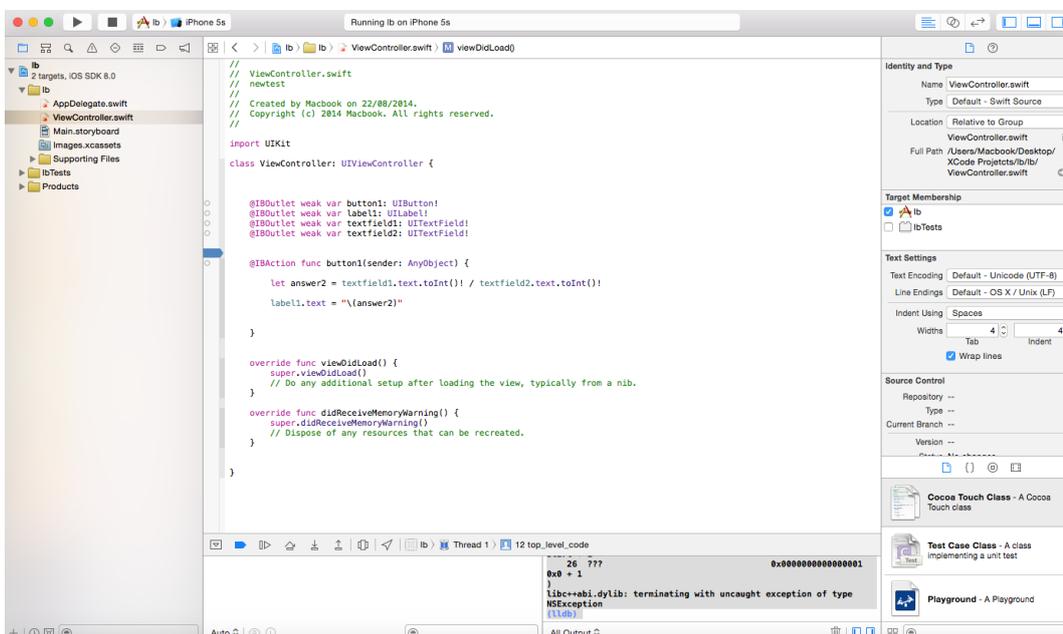


Рисунок 3 – Код Swift в программе Xcode 6

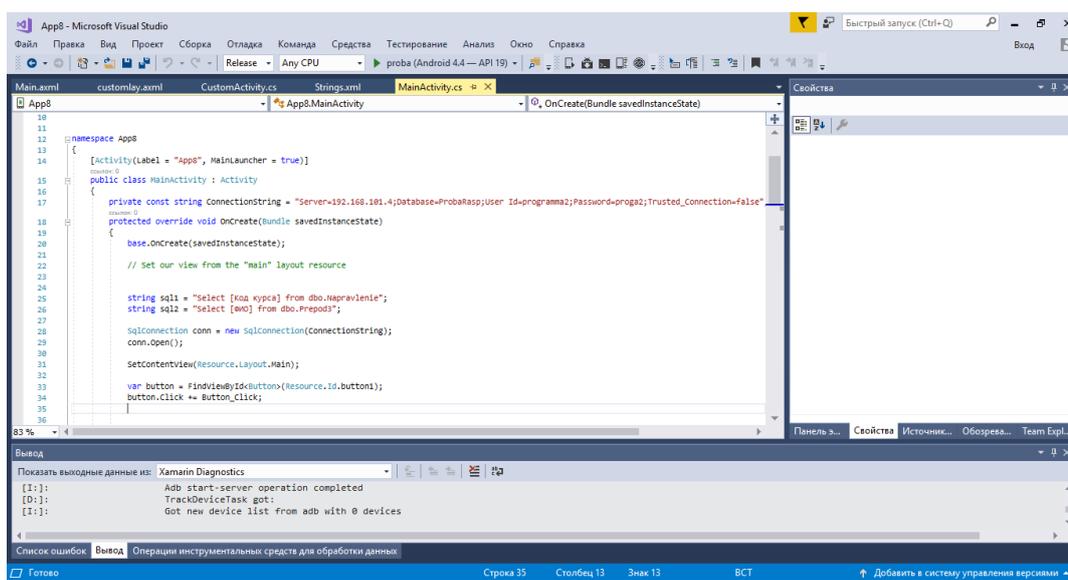
Данный язык сочетает в себе функции C и Objective-C. Swift рассчитан на использование паттернов безопасного программирования, что превращает создание приложений в гибкий и увлекательный процесс. В данном языке

используется технология Automatic Reference Counting (ARC) для упрощения процесса работы с памятью [1].

Swift – подходит для разработчиков которые готовы в первую очередь разрабатывать для ОС фирмы Apple так как не получится компилировать код программы для других ОС в следствии чего некоторые функции придется переработать для корректной работы на других языках [1].

1.4.4 C#

C# - это простой, современный объектно-ориентированный и типобезопасный язык программирования. C# относится к широко известному семейству языков C. C# является объектно-ориентированным языком, но поддерживает также и **компонентно-ориентированное** программирование[38].



```
App8 - Microsoft Visual Studio
Файл Правка Вид Проект Сборка Отладка Команда Средства Тестирование Анализ Окно Справка
Release Any CPU proba (Android 4.4 - API 19)
Main.xml customlay.xml CustomActivity.cs Strings.xml MainActivity.cs
App8.MainActivity
OnCreate(Bundle savedInstanceState)
10
11
12 namespace App8
13 {
14     [Activity(Label = "App8", MainLauncher = true)]
15     public class MainActivity : Activity
16     {
17         private const string ConnectionString = "Server=192.168.181.4;Database=ProbaRasp;user Id=programa2;Password=progaz;Trusted_Connection=false";
18         protected override void OnCreate(Bundle savedInstanceState)
19         {
20             base.OnCreate(savedInstanceState);
21             // Set our view from the "main" layout resource
22
23
24
25             string sql1 = "Select [код купца] from dbo.Napravlenie";
26             string sql2 = "Select [ово] from dbo.Preprod";
27
28             SqlConnection conn = new SqlConnection(ConnectionString);
29             conn.Open();
30
31             SetContentView(Resource.Layout.Main);
32
33             var button = FindViewById<Button>(Resource.Id.button1);
34             button.Click += Button_Click;
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
```

которой будет зависеть от выбранных компонентов. Так же Microsoft предоставляет лицензию разработчика для выставления своего продукта в магазинах данной фирмы.

1.4.5 Python

Python – это высоко уровеньный язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемость кода[20].

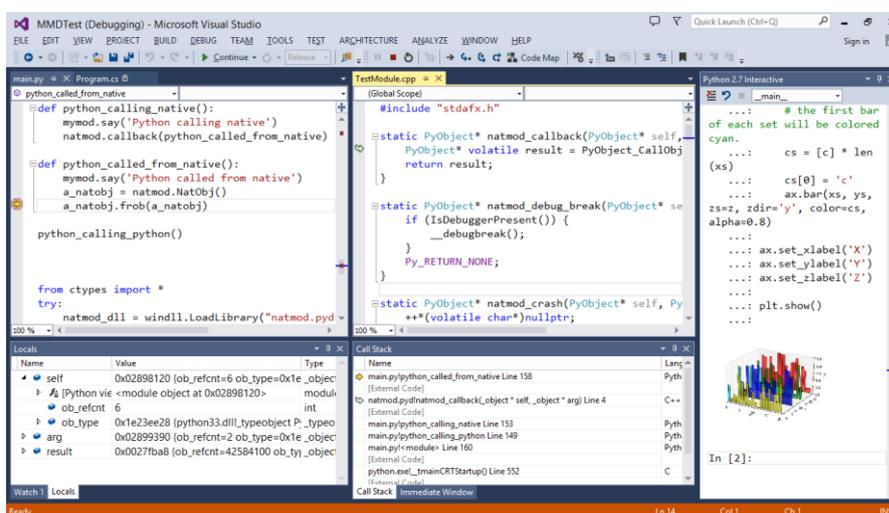


Рисунок 5 – Код Python в программе Visual Studio 2017

Python подходит для решения львиной доли повседневных задач, будь то разработка компьютерных игр, работа с электронной почты или же создание веб-сценариев. Язык python практически ничем не ограничен по этому подходит для решения как мелких проектов так и крупных. Python активно используется IT-гигантами, например Yandex и Google.

Таким образом, рассмотрев наиболее популярные технологии программирования было решено остановиться на технологии C# по причине доступности ПО и учебных материалов. В технологии Java чаще всего визуальная часть программируется непосредственно в коде, что в свою очередь может быть неудобно. Технология Swift требует больших материальных затрат,

что не каждый может себе позволить. Python не был взят за основу по причине полного отсутствия знаний операторов и опыта работы в данном языке.

1.5 Мобильные операционные системы

1.5.1 Операционная система Windows phone

Windows Phone – операционная система Microsoft являющаяся приемником Windows Mobile. Данная ОС присутствует на рынке с 2010 года и входит в тройку популярных мобильных операционных систем и в некоторых странах обходит по популярности даже знаменитую iOS. Windows Phone отлично взаимодействует с настольными версиями ОС и разработчик который разрабатывал приложения для Windows достаточно пройти небольшой курс по разработки мобильного приложения для данной мобильной ОС [25].

Первая версия данной операционной системы носила версию 7 и имела уникальный пользовательский интерфейс под название Metro. Данная ОС менее распространена чем другие но, тем не менее, она постоянно развивается и улучшается под удобство пользователя. Данная черта очень сильно мешает разработчикам приложений, так как для облегчения и быстрого действия Microsoft полностью меняют структуру ОС из за чего приложения, разработанные для старой версии приходится «пересобрать» для более новой версии .

Windows Phone является закрытой ОС и поэтому для тестирования приложений на каком-либо устройстве нужно его «разлочить» путем регистрации в программе Windows Phone Developer Registration Tool которая устанавливается вместе с пакетом для разработки Windows Phone SDK [25].

Данная ОС очень интересная начинающим разработчикам именно из за более низкой популярности на мировом рынке, так как конкуренция гораздо меньше и даже не проводя рекламу своего продукта очень большой шанс, что приложения будет скачано и установлено пользователями.

С экономической точки зрения Windows Market Place не самая удачная идея так минимальная цена и % который берёт Microsoft с продажи отбивает всякое желание делать своё приложения дешевым из за этого большинство начинающих разработчиков на Windows Phone в первую очередь копируют практически полностью уже существующие программы который доступны только на ОС конкурентов.

1.5.2 Операционная система iOS

iOS – мобильная операционная система фирмы Apple которая вышла в 2007 году эксклюзивно для iPhone и iPod touch, а позже для iPad и Apple TV.

Операционная система iOS является самой закрытой ОС среди своих конкурентов. Она помимо сильного ограничения возможностей разработчика приложений по доступности к внутренней оболочке системы ещё и является эксклюзивной для устройств фирмы Apple. Данная закрытость очень идеально подходит простым пользователем, так как все приложения до попадания в Apple Store проходят строгую сертификацию, тем самым пользователь может не бояться, что установит вредоносное приложения, а невозможность установки приложений в обход Apple Store пресекает появление любых вирусов на данных устройствах [11].

К сожалению, среда разработки Apple SDK эксклюзивна для операционной системы macOS из за чего разработчику приходится либо приобретать ноутбуки фирмы Apple либо устанавливать на виртуальную машину PC версию macOS, в народе называемую «Хакинтош» [1].

Для разработки простых приложений на iOS нужны базовые знания программирования, но если разрабатываемое приложение должно взаимодействовать с интерфейсом ОС, нужны более углубленные и специализированные знания данной ОС, в отличие от конкурентов, где функции и способ взаимодействия схожий между собой [11].

iOS не самый лучший выбор для начинающего программиста из за своей строгости и закрытости, но она может похвастаться самой лучшей совместимостью между своими устройствами, другими словами – разрабатываешь программу для одно устройства – разрабатываешь для всех устройств фирмы Apple.

1.5.3 Операционная система Android

Android – это операционная система на Linux которая используется не только на мобильных устройствах, но и на современной домашней техники (холодильниках, телевизоров и т.д.). Разработана она группой Open Handset Alliance (ОНА), при поддержке компании Google. Основной язык программирования для андроида среди разработчиков считается Java, но так же можно разрабатывать приложения на таких языка как Си, Delphi, Ruby и т.д.

Android был выпущен 23 сентября 2008 года, именно эта дата считается стартом жизни операционной системы так как в этот день вышло первое устройство которое поддерживала данную ОС. Вскоре последовали и другие анонсы производителей смартфонов на данной операционной системе [23].

Android является открытой ОС. Это означает, что любой желающий может написать программу, так как библиотеки для разработки (SDK) находятся в открытом доступе для любого желающего. Это, к слову, позволило так же разработчикам сред разработки сделать разнообразное ОП для разработки приложений. При сертификации своего приложения в магазине Google Store можно доступно и без каких-либо проблем продавать свой продукт.

Android поддерживает 2D и 3D графику позволяет создавать на данной операционной системе не только приложения контроля и отчетов но и развлекательного характера данную особенность подтверждает наличие огромной библиотеке игр в магазине приложений. Также Android позволяет

разработчикам использовать одновременно и 2D и 3D графику в своих приложениях.

Архитектура Android состоит из четырех уровней:

1) Ядро – это ключевой уровень ОС он отвечает за безопасность системы, управление памятью, процессами и энергоэффективностью системы. Ядро предоставляет работу сетевого протокола и драйвера. Кроме того она играет важную роль для связи аппаратным программным обеспечением и остальной частью системы.

2) Программное обеспечение промежуточного слоя – это набор библиотек предназначенных для работы функционала приложений:

– Surface Manager – менеджер окон. ОС Android графика не отрисовывается в буфер дисплея, вместо этого в систему поступают команды отрисовки в закадровый буфер дисплея, где они постепенно накапливаются и только потом выводятся пользователю на экран благодаря этому получаются бесшовные эффекты и прозрачные переходы между окон.

– Media Framework – библиотека, реализованная на базе Packet Video Open Core. Данная библиотека позволяет воспроизводить или записывать аудио видео контент.

– SQLite – библиотека СУБД необходима для работы с базами данных которые используются в приложениях.

– 3D библиотеки – используются для работы с 3D-графикой, так же при возможности используется аппаратное ускорение в системе. Реализовано на основе API OpenGL/

– Free Type – библиотека для работы со шрифтами и операциями над ними.

– Lib Web Core – браузерная библиотека Web Kit

– Skia Graphics Engine (SGL) – библиотека для работы с 2D-графикой.

– Libc – стандартная библиотека C языка, модифицированная для работе на устройствах на базе Linux.

Так же на этом уровне располагается среда Android Runtime. Она состоит из виртуальной Java машины Dalvik и библиотек ядра. Данная среда рассчитана для поддержке работы нескольких приложений. Каждое приложение в операционной системе Android запускается на виртуальной машины Dalvik, т. е. все приложения изолированы друг от друга. Данная технология позволяет предотвратить конфликт между приложениями, которые взаимодействуют с ядром ОС. Функция защиты является ключевой в этом уровне [6].

3) Application Framework (каркас приложений) – через данный уровень разработчики получают доступ к API. Благодаря архитектуре Framework, каждому приложению предоставляется одинаковый доступ.

В набор систем и сервисов Framework лежащие в основе каждого приложения входят:

- расширяемый набор Views, который используется для создания визуальных компонентов приложения (встроенного веб браузера, таблиц, списков);

- Content Providers – данные которые приложения «открывают» для других приложений, чтобы те могли их использовать в своей работе;

- Resource Manager – предоставляет доступ к ресурсам не несущие кода (строковые данные, файлы, грифика);

- Notification Manager, благодаря ему все приложения имеют возможность отображать собственные уведомления для пользователя в строке состояния;

- Activity Manager – данный менеджер управляет жизненным циклом приложений, а так же сохраняет данные об истории работы приложения и навигацию по ним;

- Location Manager – позволяет получать геолокационные данные для приложений.

Нужно так же понимать, что Framework отличается от обычных библиотек так как:

– выполняет код, написанный только для себя, обычные библиотеки исполняются самостоятельно;

– состоит из библиотек с разным функционалом, а обычные библиотеки содержат функционал только близкий к своей логике.

4) Уровень Applications. Здесь находятся базовые программы, установленные на Android. Такие как почта, карты, разные виды менеджеров, браузера, и т. д. Список базовых программ изменяется в зависимости от модели и версии ОС на устройстве. Как раз на этом уровне располагаются все сторонние приложения [7].

При разработке Android приложений нужно обязательно придерживаться основных принципов разработки:

а) Ресурсы нужно экономить. Несколько лет назад сотовые имели очень малую внутреннюю память и разработчикам приходилось учитывать сильную разницу доступной внутренней памяти у каждой модели устройства. Но, сейчас почти каждое устройства снабжено SD разъемом для SD карты размеры которого могут достигать вплоть до 125 гигабайт. Но это не повод не экономить место в памяти пользователя, так как в наше время пользователь пользуется одновременно множеством приложений. Представим, что установочный пакет нашего приложения в Google Store вешает 2 гигабайта и мы написали в системных требованиях точно такой же размер, мы очень быстро по обратной связи (на почту, в комментариях и т.п.) получим жалобу что ОС говорит что недостаточно места для установки приложения. Дело в том, что от выставления на сайт пакета до рабочей программы на устройстве пользователя программа проходит минимум 3 этапа. 1-й этап это скачивание с источника – пользователь непосредственно скачивает наше приложения в память устройства и это отнимет у него требуемые 2 гигабайта. 2-й этап это распаковка приложения в временную системную папку устройства а это ещё минус как минимум 2 гигабайта, данный размер может быть и больше это зависит от используемого компилятора. К слову так же стоит иногда учесть, что некоторые версии ОС не позволяют перемещать временную системную

папку из за чего она может находиться не на расширяемой SD карте устройства, а на сильно ограниченной внутренней памяти устройства. После проверки ОС на соответствие требованиям происходит 3-й этап где приложение уже устанавливается на SD карту устройства. В итоге мы получаем не 2 гигабайта в памяти устройства а как минимум 6 гигабайт занятой памяти, но после установки непосредственно сам запакованный пакет и распакованный в временной папки удаляются тем самым освобождая места [17].

б) Лучше несколько больших файлов, чем множество маленьких. Вы делаете приложение, которая выполняет свои команды со скоростью 5 миллисекунд, вы довольный, как и пользователь. Но спустя какое-то время скорость выполнение команды возрастает с 5 до 300 миллисекунд. Дело в том, что файловая система пытается сохранить жизнь своей флэшки, поэтому отслеживает интенсивные операции с файлами и часто востребованные файлы пытается перемещать с места на место. Чем больше файлов и чем меньше свободного места, тем больше ей на это нужно времени [6].

Исходя из всего изученного, можно выделить достоинства и недостатки операционной системы Android.

Достоинства:

- неограниченные возможности;
- возможность легко скачивать и устанавливать приложения из интернета;
- огромная библиотека приложений;
- свободный доступ к пользовательским файлам на внешней и внутренней памяти устройства;
- предоставляет доступ к сервисам Google;
- синхронизация сервисов Google между устройствами;
- поддержка многозадачности;
- расширения памяти;
- возможность установки других аналогов стандартных приложений;

- поддержка альтернативных версий операционных системы от независимых разработчиков;

- удобство в эксплуатации;

- разнообразные ценовые варианты устройств на Android;

- наличие Flash технологий;

- возможность полного изменения интерфейса.

Недостатки:

- высокое энергопотребление;

- приложения, которые не совместимы с разными версиями операционной системы;

- большое количество пунктов настроек в меню, которые обычному пользователю будут не понятны;

- сильная фрагментация системы;

- наличие вредоносных приложений;

- в стандартные приложения не входят программы для защиты устройства;

- система безопасности плохо защищена, так как является открытой;

- чувствительная аппаратная система;

- много однотипных программ в магазине приложений.

На сегодняшний день Android – лучшее решение для начинающих разработчиков, так как ПО для разработки приложений является бесплатным и работает на любом ПК с любой версией Windows начиная с версии Vista. В интернете есть много тематической литературы и кодов для примера, а так же специализированные сайты, где опытные разработчики обмениваются мнениями и оказывают помощь начинающим.

Таким образом, несмотря на большую конкуренцию мобильная ОС Android представляет наибольший интерес для начинающих разработчиков из за своей открытости и многочисленных площадок для продвижения своего приложения. На Windows Phone конкуренции меньше, но практически

отсутствует какая либо возможность продвижения своего продукта вне Windows Store. ОС iOS требует больших материальных затрат, что не каждый может себе позволить.

2 Практическая реализация электронного расписания

2.1 Информационная модель клиент-сервера «электронное расписание»

Проанализировав информацию, построим информационную модель комплекса программ электронное расписание

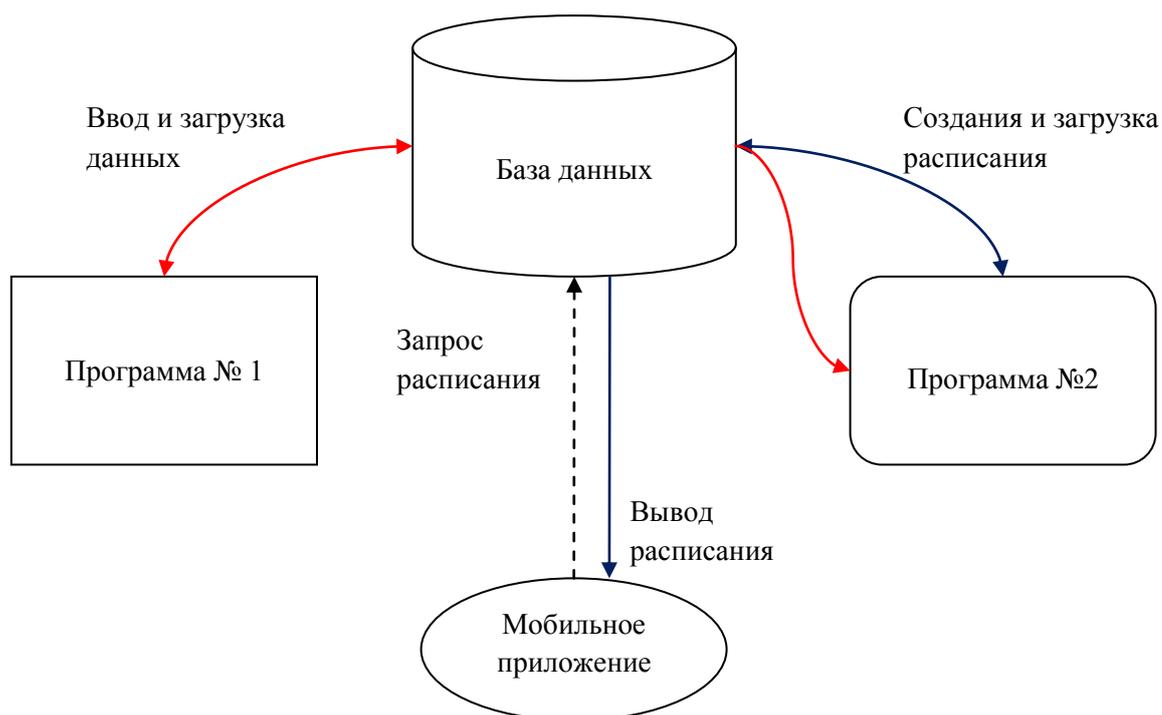


Рисунок 6 – Информационная модель

На данной информационной модели мы видим базовый функционал создания и загрузки электронного расписания. Благодаря этой модели прекрасно видно, насколько прост процесс создания электронного расписания. В центре находится база данных, которая будет хранить как данные нужные для создания расписания (информацию о кабинетах, преподавателей и т. д.), так и непосредственно само расписание. С ней работают 2 программы и мобильное приложение. Приложение № 1 вводит, редактирует необходимые данные для создания расписания, которые потом получает программа № 2. Программа № 2 после получения необходимых данных приступает к созданию расписания.

После создания расписания программа загружает ее в базу. Мобильное приложение делает запрос по дате для получения необходимого расписания. На данной модели видно, что программа № 2 не может изменять данные созданные программой № 1, так же как и программа № 1 не может создавать и изменять расписание. Мобильное приложение только получает данные – загружать что-либо в базу она не может. Информационная модель будет меняться после обновления функционала программа, который будет добавляться после отзывов и предложений пользователей

2.2 Разработка программного продукта.

2.2.1 Среда разработки

В качестве среды разработки было выбрана программа Visual Studio 2017 фирмы Microsoft. Существует множество сред разработки, но данный тип был выбран по причине схожего интерфейса с Turbo Delphi, а так же по причине того что данная программа может одновременно скомпилировать программу для всех требуемых ОС (Windows, Android, IOS). Visual так же предоставляет простую и удобную взаимосвязь продуктов Microsoft. На рисунке 7 представлен внешний вид программы.

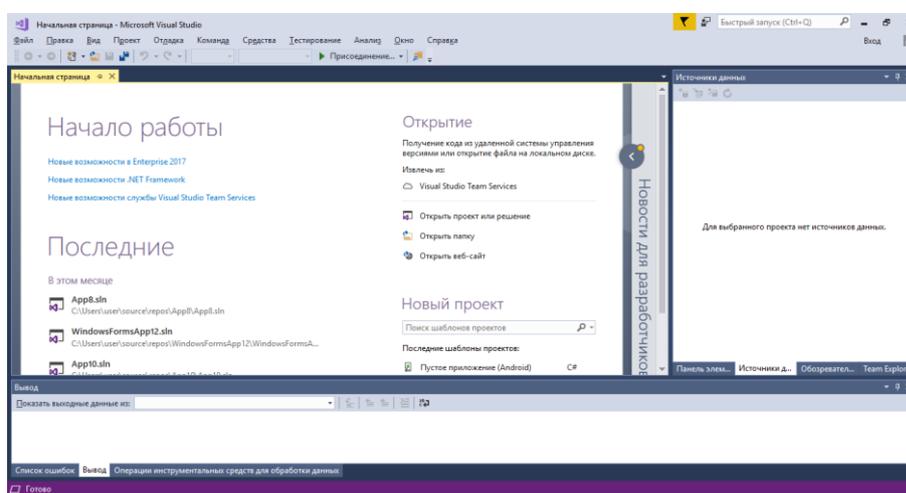


Рисунок 7 – Visual Studio 2007

Данная программа объединяет разные языки программирования (C++, C#) в единую интегрированную среду разработки, а также позволяет подключать язык “SQL” к проекту, и редактировать запросы, хранимые в редакторе SQL запросов. Аналогично можно редактировать HTML фрагменты.

2.2.2 Программа №1 «Ввод данных»

В данной программе пользователь вводит данные в базу данных. Эти данные в дальнейшем используются в других программах для составления расписания. На экране программы (рисунок 8) все вводимые данные после сохранения отображаются в виде таблицы.

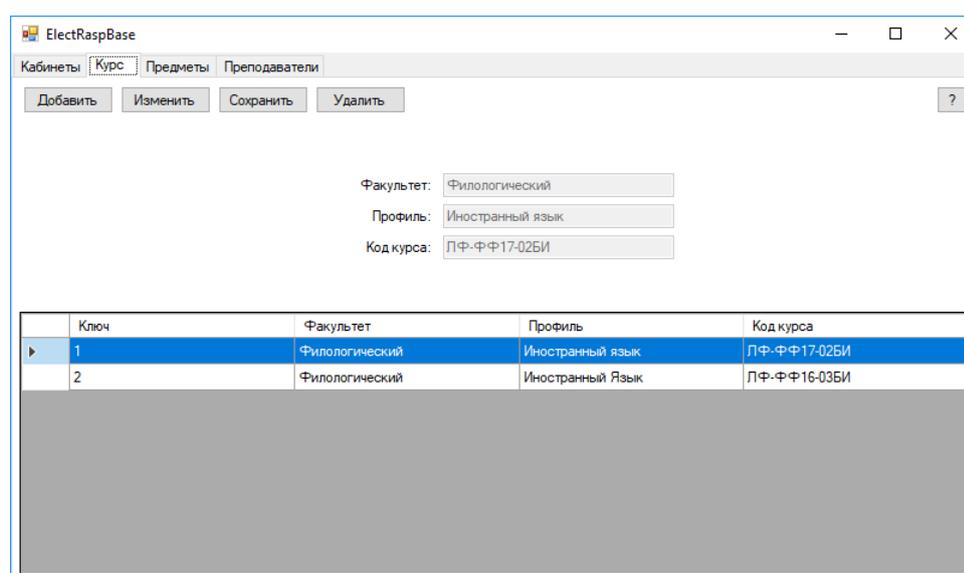


Рисунок 8 – Программа для ввода данных

Навигация по таблице осуществляется при помощи курсора мыши. Перед сохранением данных производится проверка вводимых данных на соответствие типа данных. В случае не соответствия программа сообщает пользователю о том, в каком поле ввода данных ошибка и просит её исправить. Дальнейшие операции в программе невозможны, пока пользователь не исправит ошибку. Для удобства ввода данных они распределены по вкладкам. Пользователь данной программы имеет возможность добавления,

удаление и редактирование введенных им данных. Данная программа не имеет доступа к таблице расписания.

2.2.3 Программа №2 «Создание расписания»

Данная программа выгружает данные из базы, которые были введены в программе №1, добавляет их в соответствующие списки (рисунок 9). В дальнейшем пользователь просто собирает расписание, выбирая значения из списка.

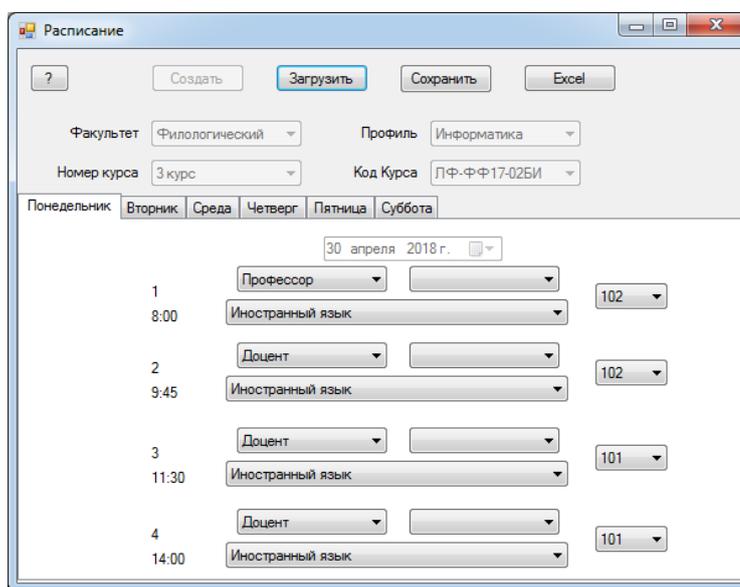


Рисунок 9 – Главное окно программы №2

При нажатии на кнопку «создать» на начальном экране открывается меню для выбора даты начала недели (рисунок 10). В случае выбора другой даты программа выдаст ошибку и позволит пользователю перевыбрать дату, либо отменить действие путем закрытия окна.

После выбора даты на главном окне поля выбора из списка (combobox) становятся доступны для дальнейшей работы с ними. При сохранении расписания происходит проверка на совпадения:

- а) существует ли в таблице данное расписание для текущего курса;
- б) занятость преподавателей;
- в) занятость кабинетов.

В случае совпадений программа, прежде чем сохранять, сообщит об этом пользователю. При возникновении 2 и 3 пункта программа даст возможность пользователю проигнорировать данные сообщения и процесс сохранения продолжится. Во время 1 пункта дальнейший процесс сохранения невозможен.

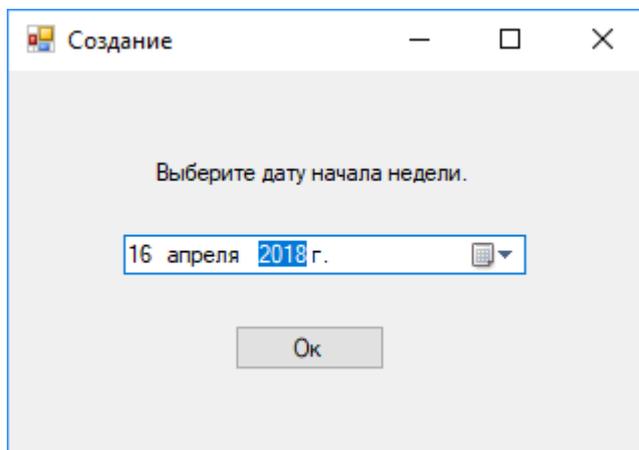
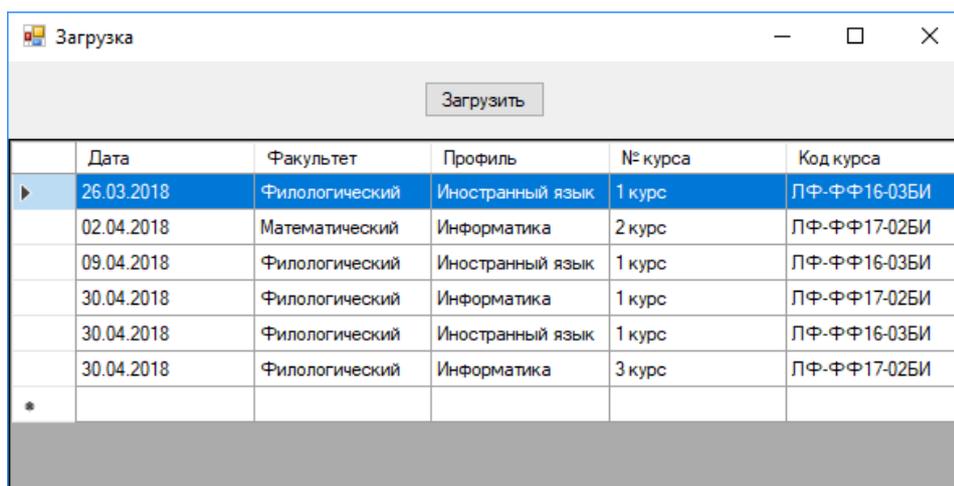


Рисунок 10 – Выбор даты

Для того что бы пользователь мог изменить уже существующее расписание в программе иметься функция загрузки уже созданного расписания. При нажатии кнопки «загрузить» появляется список созданных расписаний (рисунок 11).



	Дата	Факультет	Профиль	№ курса	Код курса
▶	26.03.2018	Филологический	Иностранный язык	1 курс	ЛФ-ФФ16-03БИ
	02.04.2018	Математический	Информатика	2 курс	ЛФ-ФФ17-02БИ
	09.04.2018	Филологический	Иностранный язык	1 курс	ЛФ-ФФ16-03БИ
	30.04.2018	Филологический	Информатика	1 курс	ЛФ-ФФ17-02БИ
	30.04.2018	Филологический	Иностранный язык	1 курс	ЛФ-ФФ16-03БИ
	30.04.2018	Филологический	Информатика	3 курс	ЛФ-ФФ17-02БИ
*					

Рисунок 11 – Окно загрузки

2.2.4 Программа №3 «Мобильное приложение»

Во время запуска приложения происходит первое подключение к базе данных. Во время этого подключения происходит проверка доступности сервера базы данных. Если сервер недоступен приложение сообщит об этом. Пройдя проверку доступности, происходит разрыв соединения от базы и программа загружает основное окно приложения. Дизайн мобильного приложения очень прост. Оно использует 1 окно, 2 поля списка и одну кнопку «поиск». Пользователь сначала выбирает код курса и дату и далее нажимает кнопку поиска, в этот момент происходит второе подключение к базе только уже с поиском нужных данных и загрузки их в мобильное приложение. Загруженное расписание отображается ниже кнопки «Поиска». После загрузки расписания, программа опять делает разрыв соединения. Данные разрывы необходимы в связи с тем, что не у каждой фирмы есть лицензия SQL сервера, а бесплатные версии сильно ограничены. Благодаря данным разрывом можно обойти ограничение по количеству активных подключений. Технически все равно можно «столкнуться» с данным ограничением, но в этом случае пользователь увидит ошибку о не доступности сервера, которая сама решится при повторном подключении, так как у других пользователей уже произойдет один из двух процессов разрыва соединения.

ЗАКЛЮЧЕНИЕ

Проанализировав учебную литературу по теме исследования и рынок приложений, связанных с электронным расписанием можно с уверенностью сказать, что на рынке немало программ для составления расписания, но с вязи с тем, что нет единого шаблона расписания у организаций, можно с легкостью найти пользователя для своей программы. Особенно, если она создавалась с «оглядкой» на конкретную организацию.

На сегодняшний день практически все пользователи мобильных устройств имеют доступ в интернет, поэтому создание мобильного доступа к конкретной информации крайне необходимо. Пользователи мобильного приложения освобождаются от необходимости лишних действий для получения доступа к электронному расписанию, которое может быть расположено на сайте организации, или от необходимости идти к информационной панели для получения сведения о расписании (что может быть критичным для обучающихся на заочном отделении). Это так же сводит практически к нулю вес лишних данных получаемым пользователем, а при некоторых мобильных тарифах у операторов это важно.

Разработанный комплекс программ клиент-сервер «электронное расписание» имеет свой потенциал среди других приложений благодаря своей простоте. Хоть она и уступает разнообразию функционала некоторым своим аналогам, но это не значит, что программа не будет обновляться. Все дальнейшие функции будут добавляться по мере поступления отзывов и предложений пользователей по улучшению программы.

Основные результаты исследования представлены на научно-практической конференции «Современное педагогическое образование: теоретический и прикладной аспекты» (секция «Информационно-управляющие системы») и XV Международной научно-практической конференции «Научный форум: технические и физико-математические науки».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аарон, Х. Objective-C. Программирование для iOS и MacOS / Х. Аарон. – Санкт-Петербург : Питер, 2012. – 254 с.
2. Албахари, Д. С# 6.0 Справочник. Полное описание языка / Д. Албахари, Б. Албахари. – Москва : Вильямс, 2017. – 1040 с.
3. Балтак, С. В. Построение расписания учебных занятий на основе раскраски вершин графа / С. В. Балтак, Ю. Н. Сотсков // Информатика. – 2006. – №3. – С. 58–69.
4. Вагнер, Б. Наиболее эффективное программирование на С#. 50 способов улучшения кода / Б. Вагнер; пер. с английского Ю. Н. Артеменко. – 3-е изд. – Москва : Диалектика, 2017. – 224 с.
5. Гафаров, Е. Р. Математические методы оптимизации при составлении учебного расписания / Е. Р. Гафаров, А. А. Лазарев // Новые информационные технологии в образовании. Сборник научных трудов. – Москва : 1С-Публишинг, 2013. – 237 с.
6. Голощاپов, А. Л. Google Android. Системные компоненты и сетевые коммуникации / А. Л. Голощاپов. – Санкт-Петербург : БХВ-Петербург, 2012. – 384 с.
7. Голощاپов, А. М. Google Android. Программирование для мобильных устройств / А. М. Голощاپов. – 2-е изд., перераб. и доп. – Санкт-Петербург : БХВ-Петербург, 2012. – 443 с.
8. Гэри, М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон. – Москва : Мир, 1982. – 416 с.
9. Дейтел, П. Android для программистов. Создаем приложения / П. Дейтел, Х. Дейтел, Э. Дейтел, М. Моргано. – Санкт-Петербург : Питер, 2013. – 560 с.
10. Дрейер, А. М. С# для школьников / А. М. Дрейер. – Москва : НОИ ИНТУИТ, 2016. – 164 с.

11. Дэйв, М. iOS 5 SDK. Разработка приложений для iPhone, iPad и iPod touch / М. Дейв, Н. Джек, Л. Джефф. – Москва : Вильямс, 2012. – 624 с.
12. Ичбана, Д. Бил Гейтс и сотворение Microsoft / Д. Ичбана, С. Кнеппер. – Ростов-на-Дону : Издательство «Феникс», 1997. – 353 с.
13. Ишкова, Э. А. Самоучитель C#. Начала программирования / Э. А. Ишкова. – 2-е изд. – Санкт-петербург : Наука и техника, 2013. – 496 с.
14. Казанский, А. А. Объектно-ориентированное программирование на Visual Basic 2010 и Visual C# 2010 в среде разработки Microsoft Visual Studio / А. А. Казанский. – Москва – 2012. – 424 с.
15. Кармен, Д. Создание приложений для Android за 24 часа / Д. Кармен, Д. Лорен, К. Шейн; пер. с англ. М. А. Райтмана. – Москва : Эксмо, 2015. – 526 с.
16. Культин, Н. Microsoft Visual C# в задачах и примерах / Н. Культин. – Санкт-Петербург : БХВ-Петербург, 2012. – 311 с.
17. Левин, А. Android на планшетах и смартфонах / А. Левин. – Санкт-Петербург : Питер, 2013. – 192 с.
18. Ликнесс, Д. Приложения для Windows 8 на C# и XAML / Д. Ликнесс. – Санкт-Петербург : Питер, 2013. – 368 с.
19. Что такое информационные технологии, и кто с ними работает? – МоеОбразование.ru – интернет-портал для старшеклассников, абитуриентов, студентов [Электронный ресурс] – Режим доступа: https://moeobrazovanie.ru/chto_takoe_informatsionnye_technologii.html
20. Общедоступная многоязычная универсальная интернет-энциклопедия со свободным контентом [Электронный ресурс]. – Режим доступа: <http://ru.wikipedia.org>
21. Петцольд, Ч. Создание мобильных приложений с помощью Xamarin.Forms [Электронный ресурс] / Ч. Петцольд. – Microsoft Press, 2016. – Режим доступа: <https://docs.microsoft.com/ru-ru/xamarin/xamarin-forms/creating-mobile-apps-xamarin-forms/>

22. Пугачев, С. В. Разработка приложений для Windows Phone 7.5 / С. В. Пугачев, С. И. Павлов, Д. В. Сошников. – Санкт-Петербург : БХВ-Петербург, 2012. – 384 с.

23. Рето, М. Android 2. Программирование приложений для планшетных компьютеров и смартфонов / М. Рето. – Москва : Эксмо, 2011. – 297 с.

24. Сайт о высоких технологиях, оперативные новости индустрии, тестовые испытания, обзоры оборудования и советов для разработчиков [Электронный ресурс]. – Режим доступа: <http://www.ixbt.com>

25. Сборник уроков, полезных статей, советов для разработчиков приложений на операционной системе Android [Электронный ресурс]. – Режим доступа: <http://www.developer.android.com>

26. Смирнов, В. В. Петербургские школы и школьные здания. История школьного строительства в Санкт-Петербурге – Петрограде – Ленинграде 1703 – 2003 гг. / В. В. Смирнов. – Санкт-Петербург : Издательство «Русско-Балтийский информационный центр «БЛИЦ», 2003. – 192 с.

27. Стиллмен, Э. Изучаем С# / Э. Стиллмен, Д. Грин. – 3-е изд. – Санкт-Петербург : Питер, 2014. – 816 с.

28. Урнов, В. А. Расписание – наиболее востребованный АРМ в образовании / В. А. Урнов // Информатика и образование. – 2001. – № 4. – С. 47–52.

29. Усов, В. А. Swift. Основы разработки приложений под iOS / В. А. Усов. – Санкт-Петербург : Питер, 2016. – 304 с.

30. Фелкер, Д. Android: разработка приложений для чайников / Д. Фелкер. – Москва : Диалектика, 2012. – 236 с.

31. Фленов, М. Библия С# / М. Фленов. – Санкт-Петербург : БХВ-Петербург, 2016. – 544 с.

32. Форум начинающих и профессиональных программистов, системных администраторов, администраторов баз данных [Электронный ресурс]. – Режим доступа: <http://CyberForum.ru>

33. Хабрахабр – крупнейший в Европе ресурс для IT-специалистов, созданный компанией «ТМ» в 2006-м году. [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru>

34. Хакер – одно из крупнейших российских медиа об it и it-безопасности, посвященное вопросам компьютерных трюков, информационной защиты, программирования и администрирования компьютерных сетей. [Электронный ресурс]. – Режим доступа: <https://хакер.ru/>

35. Харди, Б. Программирование под Android. Для профессионалов / Б. Харди, Б. Филлипс. – Санкт-Петербург : Питер, 2014. – 592 с.

36. Хашими, С. Разработка приложений для Android / С. Хашими, С. Коматинени, Д. Маклин. – Санкт-Петербург : Питер, 2011. – 736 с.

37. Хейлсберг, А. Язык программирования С#. Классика Computer Science / А. Хейлсберг, М. Торгесен, С. Вилтамут, П. Голд. – 4-е изд. – Санкт-Петербург : Питер, 2012. – 784 с.

38. Шарп, Д. Microsoft Visual С#. Подробное руководство / Д. Шарп. – 8-е изд. – Санкт-Петербург : Питер, 2017. – 848 с.

39. Шилд, Г. С# 4.0. Полное руководство / Г. Шилд; пер. с англ. И. В. Берштейна. – Москва : ООО «И.Д. Вильямс», 2011. – 1056 с.

40. Docs.microsoft.com – хранилище технической документации, справочных материалов по API, примеров кода, кратких инструкций и руководств для разработчиков и IT-профессионалов. [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com>.