

Министерство образования и науки Российской Федерации  
Технический институт (филиал) федерального государственного  
автономного образовательного учреждения высшего образования  
«Северо-Восточный федеральный университет  
имени М.К. Аммосова» в г. Нерюнгри

**ПРИКЛАДНЫЕ ИССЛЕДОВАНИЯ  
В ОБЛАСТИ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**Материалы региональной научно-практической конференции  
студентов и школьников  
по прикладной математике и информатике,  
посвящённой 225-летию со дня рождения Н.И. Лобачевского  
(25.03.2017 г.)**

Нерюнгри  
Издательство ТИ (ф) СВФУ  
2017

УДК 51:002  
ББК 22.1+32.81я43  
М 34

Утверждено научно-техническим советом ТИ (ф) ФГАОУ ВО «СВФУ»

**Редактор**

доцент кафедры математики и информатики ТИ (ф) СВФУ

***М.Ю. Похорукова***

*В представленном сборнике материалов научно-практической конференции, приуроченной к 225-летию со дня рождения Н.И. Лобачевского, рассмотрены актуальные вопросы прикладной математики и информатики, посвященные решению задач, связанных с программными средствами и математическими основами информатики.*

*Сборник предназначен для студентов и школьников, занимающихся научно-исследовательской работой.*

УДК 51:002  
ББК 22.1+32.81я43

© Технический институт (ф) СВФУ, 2017

## СОДЕРЖАНИЕ

<b>Абрамов Д.А.</b> Автоматизация процесса обработки данных транспортной компании...4	4
<b>Антонюк В.Л.</b> Решение задач по нахождению количества информации.....5	5
<b>Базылев В.И.</b> Разработка сайта предприятия для сдачи автомобилей в аренду.....8	8
<b>Батян А.А.</b> Проектирование автоматизированной системы для регистратуры «Горизонт».....12	12
<b>Голоков Н.А.</b> Разработка информационной системы для оптимизации деятельности аэропорта в г. Якутск.....15	15
<b>Иванов В.Д.</b> Автоматизированная система корпоративного негосударственного пенсионного обеспечения в АО ХК «Якутуголь».....19	19
<b>Королева В.В.</b> Обзор информационных систем для ведения бухгалтерского учета....22	22
<b>Кондратьев Н.А.</b> Создание имитационной модели для расчета необходимого количества зарядных станций в торговом центре.....26	26
<b>Кондратьев Н.А.</b> Зарядные станции для мобильных устройств.....30	30
<b>Морозов Д.Е.</b> Сравнительная характеристика кроссплатформенных фреймворков для мобильной разработки.....33	33
<b>Попова М.Н.</b> Проектирование информационной системы для автоматизированного расчета платежных поручений в АО «Газпромбанк» г. Нерюнгри.....36	36
<b>Протопопов В.А.</b> Мониторинг горных подстанций.....38	38
<b>Простакишин С.В.</b> Автоматизированная информационная система для учета выплат студентам СВФУ. Выбор средств разработки.....41	41
<b>Простакишин С.В.</b> Разработка информационной системы для оптимизации деятельности отдела по внеучебной работе ТИ (ф) СВФУ.....42	42
<b>Родионов Д.Н.</b> Проектирование структуры базы данных для автоматизированной информационной системы гостиницы.....44	44
<b>Рожкова А.Б.</b> Оптимизация количества касс в розничной торговой точке с помощью имитационной модели магазина.....47	47
<b>Семашка Е.Э.</b> Анализ программных средств автоматизации процессов торговли.....50	50
<b>Чертов Д.А.</b> Python.....52	52
<b>Великий русский математик Николай Иванович Лобачевский</b> .....55	55

## Автоматизация процесса обработки данных транспортной компании

*Абрамов Д.А., студент гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри*

*E-mail: ma3x793921@gmail.com*

*Научный руководитель:*

*старший преподаватель кафедры МИИ*

*Соболева Н.И.*

Автоматизация офисной деятельности и сокращение затрат человеческих ресурсов на ее ведение является основополагающим критерием при уменьшении временных затрат персонала для выполнения учетных операций и сокращения влияния человеческого фактора на работу с документацией, что в конечном итоге ведет к повышению качества работы и, следовательно, к улучшению эффективности деятельности предприятия. В частном случае, внедрение автоматизированных программных средств учета решает целый ряд проблем, которые до этого снижали скорость и эффективность документооборота предприятия. Уровень и качество внедрения систем автоматизации учета на предприятиях чаще всего не соответствуют требованиям персонала и руководства. Сложные в освоении крупные программные продукты имеют избыточный функционал, не учитывают тонкую специфику учетной работы конкретного предприятия или отдела, а также имеют высокую стоимость внедрения и сопровождения, что не всегда экономически целесообразно. Работнику в сфере учета необходимо программное обеспечение, обладающее минимально необходимым и достаточно для выполнения работы функционалом, являющимся легким в освоении и использовании. Руководству же необходимо минимизировать затраты на внедрение, обучение персонала и сопровождение конкретного программного продукта. Для достижения этих целей был разработан программный продукт для автоматизации некоторых аспектов учета автотранспортной компании в лице ИП Сокольников А. П.

Программа разработана на языке Visual C# средствами среды программирования Microsoft Lightswitch. В качестве сервера базы данных используется Microsoft SQL Server редакции Express.

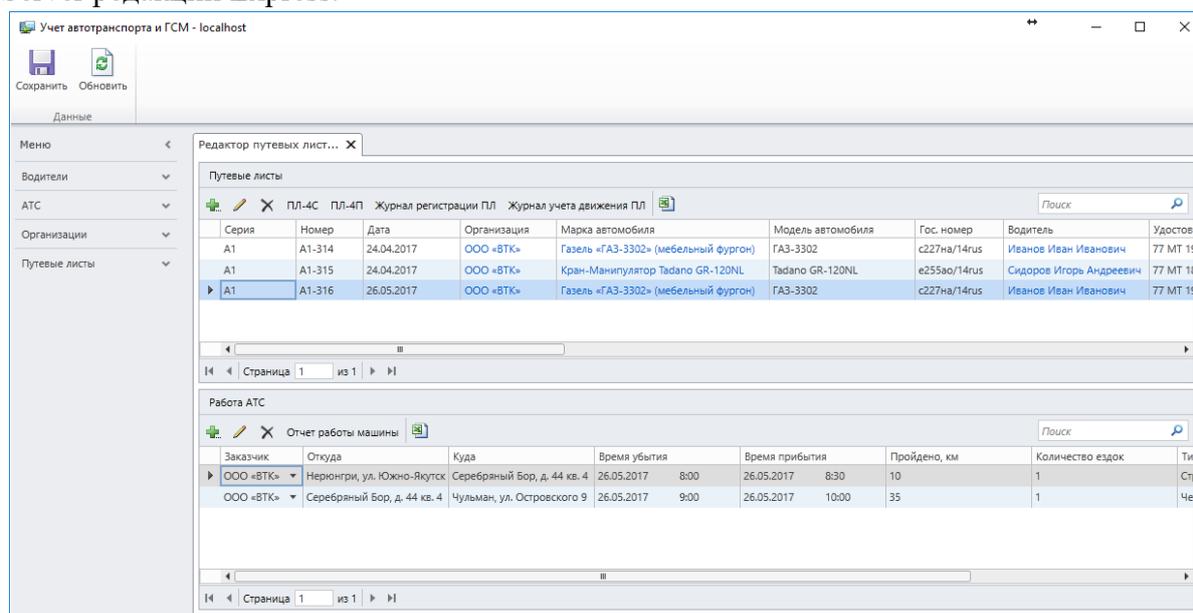


Рис. 1. Главное окно программы

Принцип работы данной программы аналогичен другим подобным решениям на рынке: пользователь ведет учет организаций-заказчиков, организаций, парка автотранспортных средств и путевых листов путем работы с таблицами базы данных. Для того чтобы оптимизировать эту работу, в системе предусмотрена возможность автозаполнения некоторых полей. Например, при создании путевого листа, пользователю не обязательно вручную записывать водителя, номер автомобиля или норму расхода топлива. Достаточно лишь выбрать автомобиль или водителя, и поля заполняются автоматически.

Также в системе предусмотрены проверки введенных данных. Это нужно для того, чтобы сократить число ошибок, которые может допустить пользователь. Например, пользователь не может добавить двух водителей с одинаковыми номерами водительских удостоверений или два автомобиля с одинаковым государственным номером. Если пользователь допустит такую ошибку, программа подсветит поле и не даст записать данные в базу, пока ошибка не будет исправлена.

Проблем при проверке данных на этом экране: 1. Щелкните здесь для получения дополнительных сведений.

✖ Гос. номер: Автомобиль с данным госномером уже существует в базе

Марка:

Модель:

Базовый расход ГСМ на 100 км, л: 0

Гос. номер: с227на/14rus

Гаражный номер:

Инвентарный номер:

Рис. 2. Ошибка ввода данных

Основной функцией программы является возможность вывода типовой документации в виде документов Microsoft Word и Excel. Такими документами являются путевые листы формата 4-П и 4-С, журналы регистрации и движения путевых листов, отчеты о работе автотранспортного средства. Пользователь, после добавления нового путевого листа в программу может вывести на печать необходимый документ. В котором уже автоматически будут записаны необходимые данные.

Подводя итоги, можно сказать, что данная система позволяет значительно сократить затраты временных ресурсов на ведение учета автотранспорта и путевых листов на предприятии а также сокращает число ошибок, вызванных влиянием человеческого фактора. Что и является главной целью автоматизации в целом.

### **Решение задач по нахождению количества информации**

*Антонюк В.Л., студент 1 курса электроэнергетического факультета,  
Дальневосточный государственный аграрный университет, г. Благовещенск  
E-mail: paparogue13@gmail.com  
Научный руководитель:  
к.т.н., доцент Коваль А.А.*

Вопрос об измерении количества информации является очень важным как для науки, так и для практики. В самом деле, если информация является предметом нашей деятельности, мы ее храним, передаем, принимаем, обрабатываем. Поэтому важно

договориться о способе ее измерения, позволяющем, например, ответить на вопросы: достаточно ли места на носителе, чтобы разместить нужную нам информацию, или сколько времени потребуется, чтобы передать ее по имеющемуся каналу связи. Величина, которая нас в этих ситуациях интересует, называется **объемом информации**. В таком случае говорят об алфавитном, или объемном, подходе к измерению информации.

Алфавитный подход к измерению информации применяется в цифровых (компьютерных) системах хранения и передачи информации. В этих системах используется двоичный способ кодирования информации. При алфавитном подходе для определения количества информации имеет значение лишь размер (объем) хранимого и передаваемого кода.

$$2^i = N.$$

Число  $N$  называется мощностью алфавита. Число  $i$  называется информационным весом символа алфавита. В двоичном коде каждая двоичная цифра несет одну единицу информации, которая называется 1 бит. Бит является основной единицей измерения количества информации. Длина двоичного кода, с помощью которого кодируется символ алфавита, называется **информационным весом символа**.

Пример, алфавит мощностью в 32 символа будет иметь вес 5 бит ( $32 = 2^5$ ).

Тогда текст из 1000 символов будет иметь информационный объем 5000 битов.

Итак, если  $i$  — информационный вес символа алфавита, а  $K$  — количество символов в тексте, записанном с помощью этого алфавита, то информационный объем  $I$  текста выражается формулой:

$$I = K \times i \text{ (битов).}$$

Для автоматизации процесса вычисления количества и объема информации можно воспользоваться программой для работы с электронными таблицами, например Microsoft Excel. Для примера, разберем две задачи с событийным подходом и способ их решения через таблицы Excel.

В озере обитает 12500 окуней, 25000 пескарей, а карасей и щук по 6250. Какое количество информации несет сообщение о ловле рыбы каждого вида. Сколько информации мы получим, когда поймем какую-нибудь рыбу?

**Дано:**  $K_o=12500$ ,  $K_n=25000$ ,  $K_k=K_{щ}=6250$ .

**Найти:**  $I_o, I_n, I_k, I_{щ}, I$ .

**Решение:**

1. Найдем общее количество рыбы:  $N = K_o + K_n + K_k + K_{щ}$ .

События	Число событий
окунь	12500
пескарь	25000
карась	6250
щука	6250
Всего (N)	50000

2. Подсчет вероятности каждого события (формула  $p_i = K_i/N$ ).

События	Число событий	Вероятность (P=K/N)
окунь	12500	0,25
пескарь	25000	0,5
карась	6250	0,125
щука	6250	0,125
Всего (N)	50000	1

3. Подсчет количества информации о каждом происходящем событии (формула  $I_i = \log_2(1/p_i)$ ).

События	Число событий	Вероятность (P=K/N)	Количество информации в происходящем событии $I = \log_2(1/P)$
окунь	12500	0,25	2
пескарь	25000	0,5	1
карась	6250	0,125	3
щука	6250	0,125	3
Всего (N)	50000	1	0

4. Подсчет количества информации для событий с различными вероятностями (формула Шеннона).

События	Число событий	Вероятность (P=K/N)	Количество информации в происходящем событии $I = \log_2(1/P)$	$P * \log_2(P)$
окунь	12500	0,25	2	-0,5
пескарь	25000	0,5	1	-0,5
карась	6250	0,125	3	-0,375
щука	6250	0,125	3	-0,375
Всего (N)	50000	1	0	0

События	Число событий	Вероятность (P=K/N)	Количество информации в происходящем событии $I = \log_2(1/P)$	$P * \log_2(P)$	$I = \sum P_i * \log_2 P_i$
окунь	12500	0,25	2	-0,5	-1,75
пескарь	25000	0,5	1	-0,5	
карась	6250	0,125	3	-0,375	
щука	6250	0,125	3	-0,375	
Всего (N)	50000	1	0	0	

Ответ: для ловли одной рыбы мы получим 1,75 бит информации.

Имея данную таблицу, мы можем изменять значения переменных и получать ответы на поставленные задачи.

В непрозрачном мешочке хранятся 10 белых, 20 красных, 30 синих и 40 зелёных шариков. Какое количество информации будет содержать зрительное сообщение о цвете вынутого шарика?

События	Число событий	Вероятность (P=K/N)	Количество информации в происходящем событии $I = \text{LOG}_2(1/P)$	$P * \text{LOG}_2(P)$	$I = \sum P_i * \text{LOG}_2 P_i$
белые шары	10	0,1	3,321928095	0,33219281	1,846439345
красные шары	20	0,2	2,321928095	0,46438562	
синие шары	30	0,3	1,736965594	0,52108968	
зеленые шары	40	0,4	1,321928095	0,52877124	
всего (N)	100				

Получаем, что количество информации о цвете вынутого шара будет равно приблизительно 1,85 бит.

### Разработка сайта предприятия для сдачи автомобилей в аренду

*Базылев В.И., студент гр. ПМ-13, ТИ (ф) СВФУ, г. Нерюнгри*

*E-mail: [yovaviper@mail.ru](mailto:yovaviper@mail.ru)*

*Научный руководитель:*

*к.п.н., зав. кафедрой МиИ Самохина В.М.*

Ресурс бронирования автомобилей представляет собой стол заказов или интернет магазин, поэтому в качестве шаблона было принято решение создать небольшой функционирующий интернет-магазин с возможностью последующего изменения.

Для корректной работы интернет-ресурса необходима база данных, которая содержит в себе 3 таблицы:

1. Game – каталог в котором хранятся все данные об автомобилях, такие как: Название автомобиля, государственный регистрационный знак, 3 поля для хранения фамилии имени и отчества водителей, обслуживающих данный автомобиль, топливо, на котором работает автомобиль, вместимость, категория автомобиля, статус, цена за час обслуживания.

2. Orders – таблица хранящая данные о поступившем заказе. Здесь будут сохранены следующие данные: имя заказчика, номер телефона, адрес электронной почты, 10 полей для заполнения маршрута движения, поле для комментария, 2 поля для дат, поле проверки правильности и полноты заполнения, поля для хранения расчёта суммы оплаты клиентом.

3. OrderLine – для хранения обработанных заказов. Содержит в себе те же поля, что и Orders, а также новое поле для хранения общей суммы оплаты клиентом.

Чтобы элементы отображались на странице, нужно их вызвать. Так же будет использоваться специфичный тег с использованием языка C# `<%# %>`. В этом теге прописываем элементы, которые будут показываться на странице при помощи свойства Item. В конечном итоге тег вызова элемента выглядит следующим образом: `<%#`

Item.Name%> - вызов отображения элемента, хранящего название автомобиля. Аналогично вызываются и остальные элементы. Имена были заданы в классе Game.cs.

После заполнения всех полей, нужно создать кнопку для возможности бронирования автомобиля. Здесь используется тег кнопки <button>, а для передачи значений о бронируемом автомобиле используется свойство type="submit" (рис. 1).

Но помимо наполнения страницы, необходимо настроить навигацию. В этом поможет файл с расширением .Master. Он создает разметку страницы. В нашем случае страницы будет размечена на 3 области.

1. Сверху - область с названием предприятия и цветовой гаммой основного портала.
2. Слева – меню категорий автомобиля, нажав на которые нам будут показаны автомобили только определенного класса, например, «Легковой автомобиль».
3. В центре – поле, которое меняет наполнение. В качестве главной страницы – Listing.aspx, при нажатии на кнопку «Забронировать» - форма заполнения данных о клиенте.

При нажатии на кнопку «Забронировать» клиент должен попадать на страницу заполнения данных о себе.

В случае если все поля были заполнены верно, клиенту выводится сообщение с благодарностью за сделанный заказ. Это сообщение будет появляться в случае если сработает метод checkoutMessage.

```
<asp:Content ContentPlaceHolderID="bodyContent" runat="server" >
  <div id="content">
    <asp:Repeater ItemType="GameStore.Models.Game"
      SelectMethod="GetGames" runat="server">
      <ItemTemplate>
        <style>
          .status {
            color: red;
          }
        </style>
        <div class="item" id="ItemDB">
          <h3><%= Item.Name %></h3>
          <h4><b>Описание:</b></h4>
          <b>Водители:</b><br>
          <%= Item.Fam1 %><br>
          <%= Item.Fam2 %><br>
          <%= Item.Fam3 %><br>
          <b>Топливо:</b> <%= Item.Gas %><br>
          <b>Вместимость:</b> <%= Item.Capacity %><br>
          <b>Государственный номер:</b> <%= Item.Num %><br>
          <b>Статус автомобиля:</b> <span class="status"><%= Item.Status %></span>
          <h4><%= Item.Price.ToString("c") %></h4>

          *Цена указана за час использования
          <button name="add" type="submit" value="<%= Item.GameId %>">
            Забронировать
          </button>
        </div>
      </ItemTemplate>
    </asp:Repeater>
  </div>
</asp:Content>
```

Рис. 1. Код заполнения элементами страницы Listing.aspx

Теперь нужно предоставить диспетчеру возможность отвечать на полученные заказы, а также, не входя в сторонние программы редактировать базу данных автомобилей.

Условно страницу можно разделить на 4 слоя:

1. Слой подписи таблицы;
2. Слой вставки данных из базы данных;

3. Слой редактирования значений базы данных;
4. Слой вставки нового значения.

Первый слой представляет собой обычные подписи навигации по таблице. Вторым слоем вносятся в таблицу страницы данные из базы данных с уже знакомыми полями, а также добавляются две кнопки справа от заполненной таблицы: «Изменить» и «Удалить», которые ссылаются на созданные методы. В случае пока не нажата ни одна из кнопок все поля разрешены только для чтения. Если же нажать на кнопку «Изменить», все поля предоставят возможность изменить их значения, а кнопки поменяют значения на «Обновить» и «Отмена».

Внизу же страницы диспетчеру будет предоставлена возможность вставки нового элемента. Если основная таблица доступна для чтения до тех пор, пока не нажата кнопка, то поля для ввода нового элемента доступны сразу и нажав на кнопку «Вставить» элемент будет добавлен в базу данных.

На странице Orders должны отображаться поступившие заказы, с отображением введенных клиентом данных.

При запуске проекта на стороне клиента будет отображен каталог автомобилей (рис.2).

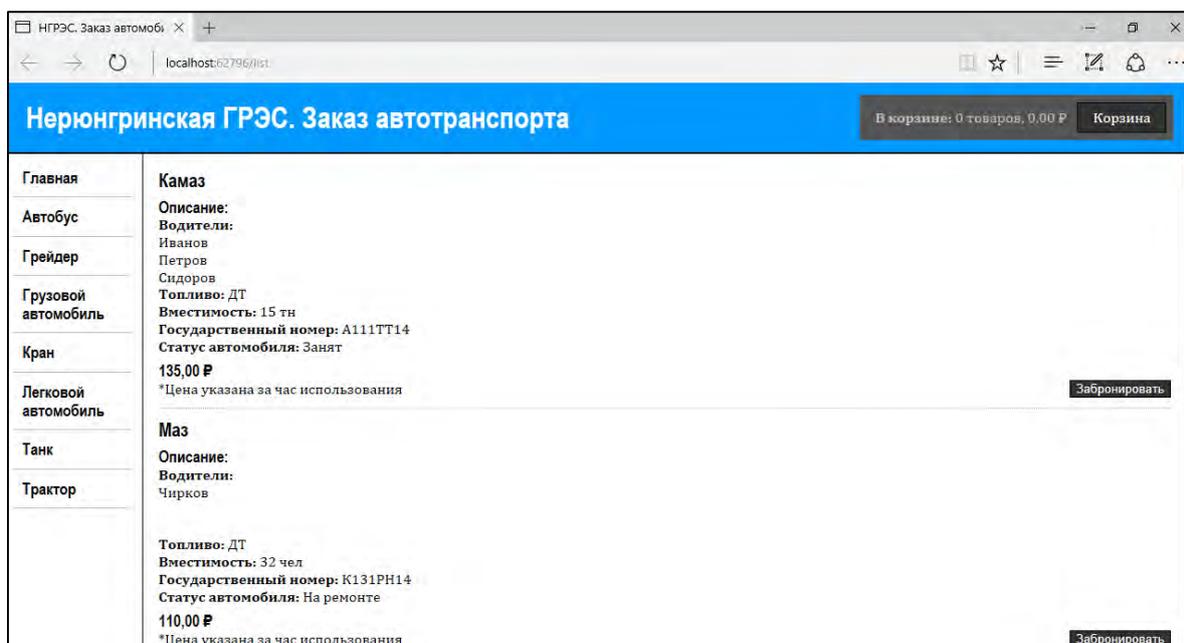


Рис. 2. Главная страница созданного ресурса

При нажатии кнопки «Забронировать» у свободного автомобиля сайт должен перейти на страницу checkout.aspx и показать поля с возможностью заполнения информации о клиенте (рис. 3).

Главная	<h2>Оформить заказ</h2> <p>Пожалуйста, введите свои данные, и мы отправим Ваш товар прямо сейчас!</p> <h3>Заказчик</h3> <p>Имя*: <input type="text" value="Владимир"/></p> <p>Номер телефона*: <input type="text" value="+79142847905"/></p> <p>Адрес электронной почты*: <input type="text" value="vovavaper@mail.ru"/></p> <h3>Маршрут движения</h3> <p>Адрес 1*: <input type="text" value="ГУС"/></p> <p>Адрес 2: <input type="text" value="к/т Октябрь"/></p> <p>Адрес 3: <input type="text" value="МЖК"/></p> <p>Адрес 4: <input type="text" value="к. Юность"/></p> <p>Адрес 5: <input type="text" value="ЦКид Пушкина"/></p> <p>Адрес 6: <input type="text" value="Аммосова"/></p> <p>Адрес 7: <input type="text" value="НГРЭС"/></p> <p>Адрес 8: <input type="text"/></p> <p>Адрес 9: <input type="text"/></p> <p>Адрес 10: <input type="text"/></p> <h3>Временной промежуток</h3> <p>Дата с*: <input type="text" value="31.05.2017 8:20"/></p> <p>Дата по*: <input type="text" value="01.06.2017 8:20"/></p> <p>Дополнительная информация: <input type="text" value="Цель: Заполнение документов"/></p> <p>Поля помеченные * обязательны для заполнения</p> <p style="text-align: right;"><a href="#">Обработать заказ</a></p>
Автобус	
Грейдер	
Грузовой автомобиль	
Кран	
Легковой автомобиль	
Танк	
Трактор	

Рис. 3. Страница заполнения данных о клиенте

В случае ввода верных дат, сайт должен показать сообщение с благодарностью (рис. 4).

The screenshot shows a web browser window with the address bar displaying 'localhost:62796/checkout'. The page title is 'Нерюнгринская ГРЭС. Заказ автотранспорта'. The main content area features a blue header with the text 'Нерюнгринская ГРЭС. Заказ автотранспорта'. Below the header, there is a sidebar with navigation links: 'Главная', 'Автобус', and 'Грейдер'. The main content area displays a large 'Спасибо!' (Thank you!) message, followed by the text: 'Решение по вашему заказу придет на указанную вами электронную почту. Спасибо что выбрали Нерюнгринскую ГРЭС!'.

Рис. 4. Ответ на запрос

Запрос удачно отправлен. Сторона администратора должна отображать таблицу поступивших заказов и предоставлять возможность редактировать базу данных. При верном входе попадаем на страницу полученных заказов (рис. 5):

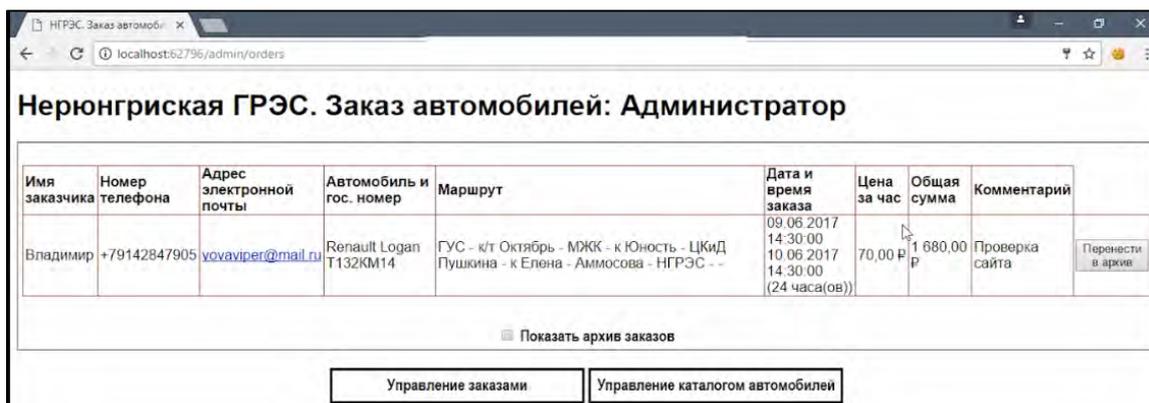


Рис. 5. Страница полученных заказов

Если нажать на кнопку «Управление каталогом автомобилей» мы должны попасть на страницу редактирования базы данных (рисунок 6).

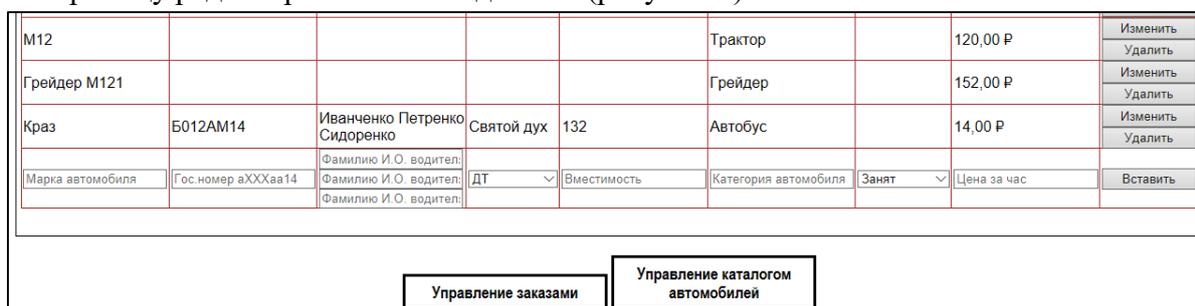


Рис. 6. Страница редактирования базы данных

Сайт протестирован, и весь описанный функционал работает.

## Проектирование автоматизированной системы для регистратуры «Горизонт»

**Батян А.А., студент гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри**  
**E-mail: mineski1337@gmail.com**  
**Научный руководитель:**  
**к.т.н., доцент кафедры МиИ Похорукова М.Ю.**

Использование автоматизированной информационной системы в регистратуре комплекса «Горизонт» позволит не только вести базу пациентов и выписывать талоны на право очереди, но и автоматически рассчитывать счета, вести учет препаратов и базу сотрудников. Электронная регистратура позволяет сократить рабочее время сотрудников самой регистратуры и лечащих врачей, что, в конечном счете, ведет к повышению качества и доступности для пациентов врачебного обслуживания.

Основными целями внедрения автоматизированной регистратуры «Горизонт» являются: обеспечение руководства комплекса целостной и актуальной информацией, снижение временных затрат на составление документации, различных перечней и списков, используемых в регистратуре

Разрабатываемая система должна решать следующие задачи:

- регистрация пациентов как непосредственно пришедших в регистратуру, так и регистрирующихся по телефону;

- хранение перечней, списков и номенклатур в электронном виде;
- предоставление возможности одновременной работы с системой несколькими пользователями;
- разделение прав доступа различным группам пользователей;
- организация быстрого поиска и сортировки различных данных по различным критериям;
- интеграция с приложениями пакета Microsoft Office;
- формирование отчетности и типовых документов.

Разрабатываемая система должна иметь модульную систему. Каждый модуль классифицирован согласно типу сущностей и классу соответствующей документации.

- Модуль «Сотрудники и квалификации» - обеспечивает хранение, поиск и сортировку информации о сотрудниках комплекса и их квалификациях.

- Модуль «Пациенты и организации» - функциональная база данного модуля схожа с предыдущим, отличием является тип обрабатываемых сущностей. Задачи, которые решает данный модуль: ведение списка организаций и закрепление пациентов за организациями, ведение списка представителей пациентов, если таковые имеются, формирование договора на оказание медицинских услуг и бланка согласия на медицинское вмешательство, подсчет стоимости лечения, как по конкретному пациенту, так и по организации в сумме.

- Модуль «Процедуры и препараты» - хранит информацию о номенклатурах процедур и препаратов, предоставляемых центром, а также обеспечивает функции записи пациентов на прием к специалисту. Основной функцией данного модуля является выборка и последующий вывод отчета о проводимых процедурах по определенному пациенту или врачу за конкретный промежуток времени.

Одним из важнейших этапов разработки автоматизированных информационных систем является проектирование структуры данных [1].

Модель данных – это совокупность структур данных и операций их обработки. С помощью модели данных могут быть представлены информационные объекты и взаимосвязи между ними [2]. В качестве основного типа моделей данных в автоматизированной информационной системе для регистратуры комплекса «Горизонт» была выбрана реляционная модель данных.

Реляционная модель данных использует организацию данных в виде двумерных таблиц [3]. Каждая таблица, называемая реляционной таблицей, представляет собой двумерный массив и обладает следующими свойствами:

- все столбцы в таблице однородные, т.е. все элементы в одном столбце имеют одинаковый тип и максимально допустимый размер;
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов в таблице не имеет значения.

Основными структурными элементами реляционной таблицы являются поле и запись. Поле (столбец реляционной таблицы) – элементарная единица логической организации данных, которая соответствует конкретному атрибуту информационного объекта. Запись (строка реляционной таблицы) – совокупность логически связанных полей, соответствующая конкретному экземпляру информационного объекта.

При проектировании структуры данных для автоматизированной информационной системы регистратуры комплекса «Горизонт» были выделены следующие сущности: организации, пациенты, представители пациентов, сотрудники

комплекса, квалификации сотрудников, препараты, процедуры, препараты для процедур, назначенные процедуры. На рисунке 1 изображена логическая модель данных в нормальной форме Бойса-Кодда, которая пригодна для дальнейшей физической реализации.

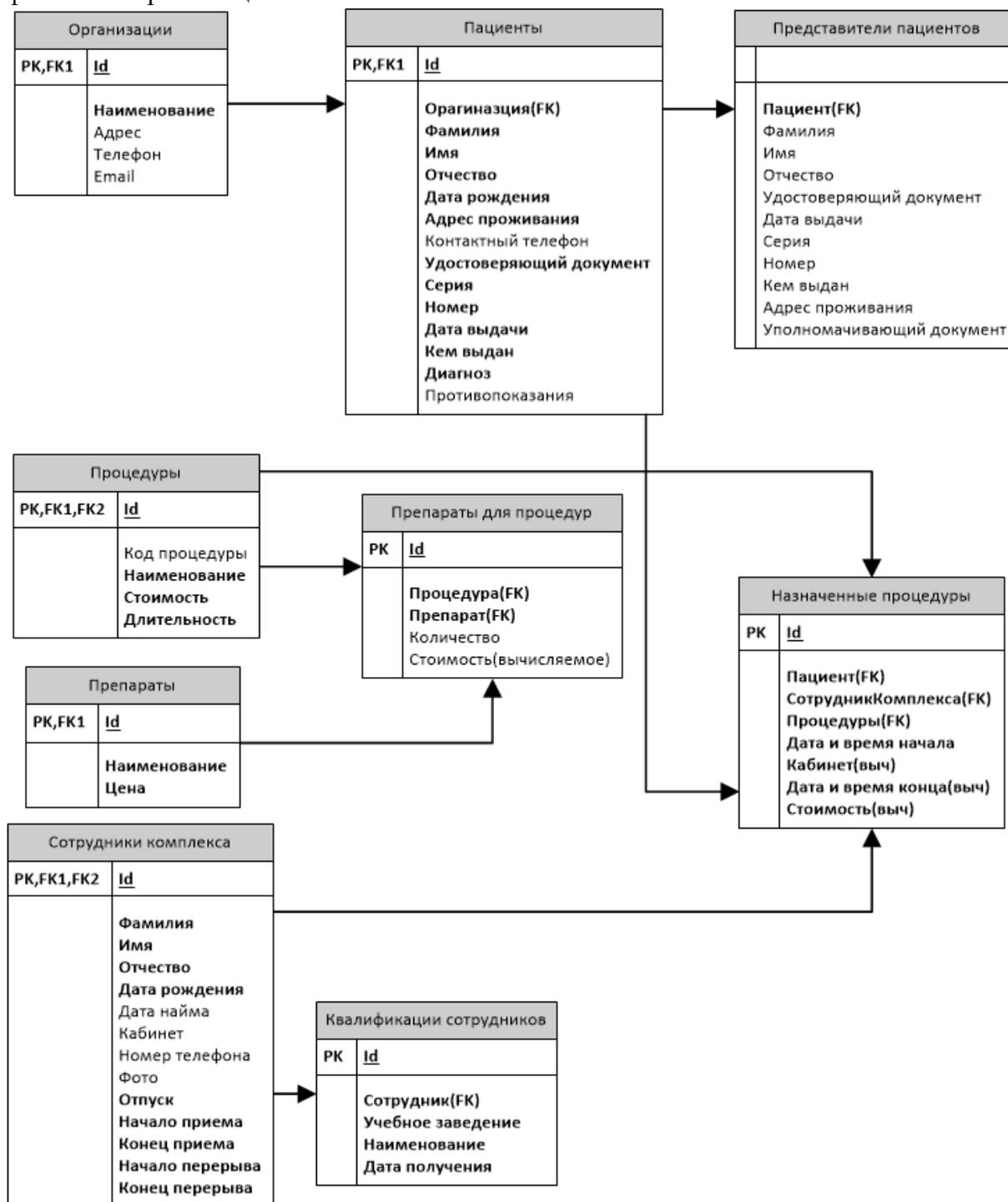


Рис. 1. Логическая структура базы данных регистратуры «Горизонт»

Список литературы:

1. Советов Б.Я., Водяхо А.И., Дубенецкий В.А., Цехановский В.В. Архитектура информационных систем: учебник для студ. Учреждений высш. проф. Образования.– М. : Издательский центр «Академия», 2012. – 288 с.
2. Дейт Крис Дж. Введение в системы баз данных. М.: Вильямс, 2006. 1328 с.
3. Кузнецов С.Д. Введение в реляционные базы данных. - 2-е изд. - М.:

## **Разработка информационной системы для оптимизации деятельности аэропорта в г. Якутск**

*Голоков Н.А., студент гр. ПМ-13, ТИ (ф) СВФУ, г. Нерюнгри*

*E-mail: chernots@list.ru*

*Научный руководитель:  
старший преподаватель кафедры МиИ Соболева Н.И.*

На данный момент СИиС является техническим центром всего АО «Аэропорт Якутск». Под ведомством службы находится более 300 персональных компьютеров, десятки принтеров и МФУ. Служба Информатики и Связи решает проблемы связанные с установкой, заменой ПК и комплектующих, внедрением и сопровождением лицензионных программных продуктов.

Учет заявок на устранение неисправностей компьютерной техники производился вручную, информация необходимая для учета заявки не была систематизирована.

Возникла необходимость разработки информационной системы для учета заявок по ремонту и обслуживанию компьютерной техники в СИиС, которая позволит повысить производительность труда сотрудников службы, качество и скорость выполнения заявки.

Служба Информатики и Связи (СИиС) отвечает за обеспечение работоспособности и своевременное решение всех проблем, связанных с информационными технологиями и связью, а именно:

- установка, замена ПК и комплектующих;
- ремонт и замена периферийных устройств (в том числе расходные материалы);
- обеспечение доступа к сети Интернет и работоспособности локальной сети;
- ответственность за исправную работу телефонии, видеонаблюдения и IP-TV;
- внедрение и сопровождение лицензионных программных продуктов.

Выделяя требования к создаваемой информационной системе, в первую очередь, данное программное обеспечение должно отличаться простотой использования.

Информационная система должна иметь интуитивно понятный интерфейс, позволяющий пользователю выполнять только те задачи, в которых он имеет достаточную компетенцию.

Требования к функциональным характеристикам:

- хранить и предоставлять доступ к базе сотрудников и компьютерной техники предприятия;
- внесение и учет поступающих заявок;
- поиск по введенным данным.

Создадим базу данных для учета заявок в Службе Информатики и Связи. Прежде всего понадобятся данные: фамилия, имя и отчество, затем указывается отдел и контактный телефон заявителя. В следующей таблице указываются подробности поступившей заявки.

Оптимизируем данные, исходя из правила трех нормальных норм. В результате получает три таблицы:

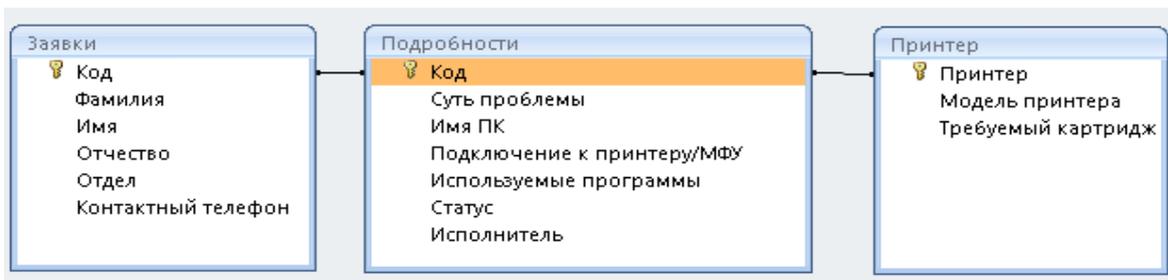


Рис. 1. Структура базы данных

Главной здесь будет таблица Заявки, которая содержит основные данные о заявителе. Она имеет релятивные связи с другими таблицами. Поле «Код» является автоинкрементным, то есть автоматически будет увеличиваться на единицу, тем самым обеспечивается уникальность ключа. В подчиненных таблицах имеются поля «Код» и «Принтер» целого типа, по ним осуществляется связь.

Главная таблица имеет связь один-к-одному с таблицей «Подробности», которая связана с таблицей «Принтер». Главная форма программы выглядит следующим образом (рис. 2):

Фамилия	Имя	Отчество	Отдел	Контактный телефон
Анисимов	Петр	Петрович	КПП №1	455559
Никифитин	Алексей	Филиппович	Международный терминал	497834
Сидоренкова	Анастасия	Васильевна	Аналитический отдел	455671
Павленко	Юлия	Александровна	Бухгалтерия	499787
Федоров	Михаил	Игнатьевич	Аэродромная служба	496545
Ростовцев	Юрий	Максимович	Бюро пропусков	497218

Суть проблемы	Имя ПК	Подключение к принтеру/МФУ	Используемые программы	Статус	Исполнитель
* замена картриджа	-	erson 4880	-	Выполняется	Истокин Илья Викторович

Рис. 2. Главная форма информационной системы

Здесь расположены три обычных панели. Свойству Align верхней панели присвоено значение alTop, нижней панели присвоено значение alBottom, средней панели – alClient, чтобы она заняла все оставшееся место на форме. Не забудьте очистить свойство Caption всех трех панелей.

Далее на верхней панели были расположены две кнопки для добавления новых и редактирования текущих заявок. И также Edit для поиска по фамилии. Свойству Name формы присвоено значение fMain, свойство Caption формы имеет текст «Аэропорт Якутск» модуль сохранен под именем Unit1.pas.

Далее в проект добавлен модуль данных (File -> New -> Data Module). Модуль данных - это невидимая форма для размещения на нем невидимых компонентов. Также модуль данных часто используют и для хранения глобальных переменных, общих функций и процедур, которые должны быть видны по всей программе.

Свойству Name модуля данных мы присвоим имя fDM, а модуль сохраним как DM.pas. Далее добавляем в модуль компонент ADOConnection с вкладки ADO палитры компонентов. Этот компонент обеспечит связь других компонентов с базой данных при помощи механизма ADO. Связь обеспечивается свойством компонента

ConnectionString Далее в модуль данных устанавливаются 2 компонента ADOTable, по одному на каждую таблицу базы данных.

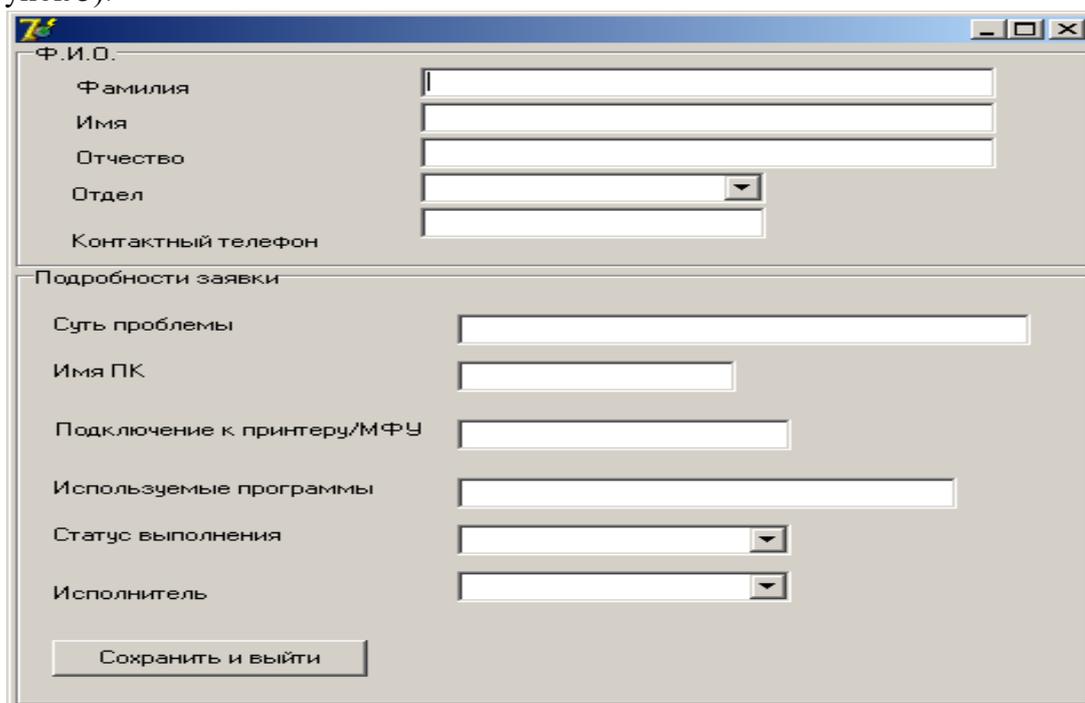
Рядом с компонентами ADOTable устанавливаем компоненты DataSource из вкладки Data Access палитры компонентов. Компонент DataSource предназначен для организации связи с НД.

К главной форме требуется подключить созданный компонент. Для этого переходим на главную форму и с помощью меню выбираем: File-Use Unit. Выделяем верхнюю сетку DBGrid, в ее свойстве DataSource выбираем fDM.DSz. В таком же свойстве нижней сетки выберите fDM.DSp. Сетки среагировали, и теперь можно видеть названия полей. Разумеется, таблица еще пуста, данных пока нет. Также выделяем обе сетки, и устанавливаем в True их свойства ReadOnly – только чтение. Таблицы будут связаны, и нужно исключить ввод данных фрагментарно. Для этого будет отдельная форма, а эти сетки нужны только для просмотра.

Теперь нужно между таблицами установить связь. Это требуется не только для того, чтобы в нижней сетке выходили данные только на сотрудника, выделенного в верхней сетке, но и для того, чтобы пользователь мог в дальнейшем вводить связанные данные в окне редактора. Снова выделяем модуль данных. Щелкаем дважды по первой таблице, чтобы открыть редактор полей. Правой кнопкой щелкаем по этому редактору и выбираем команду Add all fields (добавить все поля).

Редактор полей предназначен для настройки параметров каждого поля, для добавления новых полей или удаления имеющихся. Если в редакторе полей нет ни одного поля, то в компоненте DBGrid будут отображены все поля таблицы, имеющие параметры по умолчанию.

Далее требуется сделать окно редактора данных. Создаем новую форму (File -> New -> Form). Свойство Name переименовываем в fEditor, а при сохранении формы зададим модулю имя Editor. Командой File -> Use Unit подключаем к форме модуль данных DM. Устанавливаем на форму компоненты и получаем следующий вид (рисунок 3):



The screenshot shows a Windows-style application window titled "Ф.И.О.". The window is divided into two main sections. The top section, titled "Ф.И.О.", contains five input fields: "Фамилия", "Имя", "Отчество", "Отдел" (a dropdown menu), and "Контактный телефон". The bottom section, titled "Подробности заявки", contains six input fields: "Суть проблемы", "Имя ПК", "Подключение к принтеру/МФУ", "Используемые программы", "Статус выполнения" (a dropdown menu), and "Исполнитель" (a dropdown menu). At the bottom left of the window is a button labeled "Сохранить и выйти".

Рис. 3. Окно редактора данных

На форму были установлены две панели GroupBox с вкладки Standard, на каждую таблицу свой GroupBox. В свойстве Caption первого компонента GroupBox вписываем «Ф.И.О.» это название отразится в заголовке панели. Далее на эту панель следует установить четыре компонента DBEdit с вкладки DataControls палитры компонентов, один компонент DBComboBox для списка, поясняющие компоненты Label.

Теперь займемся подключением компонентов контроля. Удерживая <Shift>, выделяем все компоненты контроля на первой панели (все компоненты, кроме Label). В их свойстве DataSource выбираем fDM.DSz, подключив компоненты к нужному набору данных (таблице). Снимем общее выделение, и выделим первый DBEdit. В его свойстве DataField выбираем поле «Фамилия». Это свойство подключает выбранный компонент к определенному полю таблицы. Таким же образом подключим к соответствующим полям остальные компоненты.

Таким образом, при заполнении одной таблицы и перейдя к другой, запись в первой таблице еще не будет сохранена. Поле «Код» является автоинкрементным, на него завязаны остальные таблицы. До тех пор, пока запись не сохранена, в этом поле не будет никакого значения. Следовательно, данные в других таблицах не смогут привязаться к какой-то записи главной таблицы. Поэтому добавим события перехода на другой GroupBox.OnExit. В сгенерированной процедуре впишем код (рисунок 4):

```

procedure TfEditor.GroupBox1Exit(Sender: TObject);
begin
if fDM.Tz.Modified then fDM.Tz.Post;
end;

procedure TfEditor.GroupBox3Exit(Sender: TObject);
begin
if fDM.Tpodr.Modified then fDM.Tpodr.Post;
end;

```

Рис. 4. Процедура обработки выхода из GroupBox

Свойство Modified компонента ADOTable имеет логический тип - в нем содержится True, если данные были изменены, и False в противном случае. Метод Post этого компонента, как уже упоминалось, сохраняет измененную запись таблицы. При этом в поле «Код» попадет присвоенное автоматически значение. Таким образом, введенный код означает, что если запись была изменена, то следует ее сохранить.

В процедуре нажатия на кнопку "Сохранить и выйти" код простой(рисунок 5):

```

if fDM.Tz.Modified then
    fDM.Tz.Post;
    close;

```

Рис. 5. Сохранение и выход из режима редактирования

С этой формой закончено, переходим к главной форме. Начнем с кнопки «Новая заявка». Как и в предыдущем примере, нам потребуется добавить новую запись в каждую таблицу, после чего открыть окно редактора как на рисунке 6:

```

procedure TfMain.Button1Click(Sender: TObject);
begin
    fDM.Tz.Append;
    fDM.Tpodr.Append;
    fEditor.ShowModal;
end;

```

Рис. 6. Обработчик нажатия на кнопку «Новая заявка»

Сгенерируем процедуру для кнопки "Редактировать". Тут будет лишь одна строчка кода: fEditor.ShowModal;

Осталось сделать поиск заявителя по введенной фамилии. Он был реализован посредством свойства Filter(рис. 7):

```
procedure TfMain.Edit1Change(Sender: TObject);  
begin  
  if Length(Edit1.Text) > 0 then  
  begin  
    fDM.Tz.Filtered:=false;  
    fDM.Tz.Filter:='ФАМИЛИЯ LIKE ' + #39 + Edit1.Text + '% ' + #39;  
    fDM.Tz.Filtered:=true;  
  end  
  else fDM.Tz.Filtered:=false;  
end;
```

Рис. 7. Реализация поиска по введенным данным

### **Автоматизированная система корпоративного негосударственного пенсионного обеспечения в АО ХК «Якутуголь»**

*Иванов В.Д., студент гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри*

*E-mail: filessib@gmail.com*

*Научный руководитель:*

*к.т.н., доцент кафедры МиИ Похорокова М.Ю.*

В настоящее время все больше и больше российских предприятий используют автоматизированные системы документооборота. Однако существующие на рынке программные продукты не всегда полностью отвечают требованиям руководителей: не лаконичный интерфейс, высокая стоимость, комплексное прикладное решение, не удовлетворяющее условиям конкретной предметной области, что влечет за собой трудности с функционалом.

На предприятии АО ХК «Якутуголь» необходима информационная система для автоматизации операций по корпоративному негосударственному пенсионному обеспечению, которая способствует решению следующих функций: создание и работа с архивом документов, обеспечение его безопасности и сохранности; расчет негосударственного пенсионного обеспечения (НПО) участников; формирование распоряжений и различных отчетов по выплатам.

Для разработки программы автоматизированной системы по формированию пакета документов на участника корпоративного негосударственного пенсионного обеспечения, расчету и выплате ему негосударственной пенсии были использованы среда программирования C++ Builder 10.2 и сервер баз данных FireBird 2.5.

Для создания базы данных была выбрана программа IVExpert – GUI-оболочка, с помощью которой предоставляется возможность разработать и администрировать базу. На рисунке 1 представлена таблица, созданная в IVExpert, для хранения информации об участнике негосударственного пенсионного обеспечения на предприятии АО ХК «Якутуголь».

ID INTEGER NOT NULL											
#	PK	FK	UNQ	Field Name	Field Type	Domain	Size	Scale	Subtype	Array	Not Null
1	1			ID	INTEGER						<input checked="" type="checkbox"/>
2				FIO	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
3				PERSNUMBER	VARCHAR		15				<input type="checkbox"/>
4				DOB	DATE						<input type="checkbox"/>
5				GENDER	VARCHAR		1				<input type="checkbox"/>
6				PASSERIAL	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
7				PASNUMBER	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
8				PASDATEOFISSUE	DATE						<input type="checkbox"/>
9				PASISSUEDBY	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
10				RREGION	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
11				RCITY	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
12				RSTREET	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
13				RHOUSE	VARCHAR		10				<input type="checkbox"/>
14				RAPARTMENT	VARCHAR		10				<input type="checkbox"/>
15				RINDEX	VARCHAR		10				<input type="checkbox"/>
16				FREGION	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
17				FCITY	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
18				FSTREET	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
19				FHOUSE	VARCHAR		10				<input type="checkbox"/>
20				FAPARTMENT	VARCHAR		10				<input type="checkbox"/>
21				FINDEX	VARCHAR		10				<input type="checkbox"/>
22				PHONE	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
23				DIVISION	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
24				POSITION	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
25				DATEOFDISMISSAL	DATE						<input type="checkbox"/>
26				REASONFORDISMISSAL	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
27				SNILS	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
28				INN	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
29				PENSIONERSID	VARCHAR	VARCHAR1	255				<input type="checkbox"/>
30				PENSIONDATEOFISSUE	DATE						<input type="checkbox"/>

Рис. 1. Таблица, содержащая всю информацию об участнике

Для поддержки работы с базой данных в разрабатываемом приложении используются следующие компоненты: IBQuery (не визуальный компонент набора данных, выполняющий запросы SQL) и Datasource (обеспечивает связь между набором данных и компонентами отображения и редактирования данных).

В приложении имеются следующие формы: авторизация, входа в систему, для архива участников НПО и добавления участника. Форма архива содержит таблицу с данными об участниках программы за весь период использования системы. Здесь представлена краткая информация об участнике: ФИО, дата рождения, подразделение и номер по журналу участника (рис. 2):

Архив Участников НПО

Выбрать Справка ?

ФИО	Дата Рождени	Подразделение	Номер по журналу

Рис. 2. Архив участников НПО

При добавлении участника НПО необходимо реализовать расчет следующих данных: надбавок за стаж в ОАО ХК «Якутуголь», персональной надбавки, понижающего коэффициента, размера пожизненной негосударственной пенсии за выслугу лет, суммы пенсионных обязательств по выплате пожизненной негосударственной пенсии за выслугу лет за вычетом суммы, внесенной работником с учетом софинансирования государства (рис. 3):

The screenshot shows a web application window titled "Добавление участника" (Adding a participant). The window has a dark title bar with standard window controls. Below the title bar is a navigation bar with three tabs: "Представление" (Presentation), "Расчет" (Calculation), and "Банковские реквизиты" (Bank details). The main content area is a form with the following fields and sections:

- Personal Information:**
  - Фамилия Имя Отчество (Last Name, First Name, Patronymic)
  - Код физического лица (Individual Code)
  - Дата Рождения (Date of Birth)
  - Пол (Gender): Radio buttons for Male (М) and Female (Ж).
  - Паспортные Данные (Passport Data):
    - Серия (Series), Номер (Number), Дата Выдачи (Issue Date)
    - Кем Выдан (Issued by)
  - Адрес места регистрации (Registration Address):
    - Край, Регион (Region)
    - Город (пос.) (City/Town)
    - Улица (Street), Дом (House), Кв (Apartment)
    - Индекс (Index)
  - Адрес фактического проживания (Actual Residence Address):
    - Край, Регион (Region)
    - Город (пос.) (City/Town)
    - Улица (Street), Дом (House), Кв (Apartment)
    - Индекс (Index), Телефон (Phone)
  - Подразделение (Department) - dropdown menu
  - Должность (Position) - dropdown menu
  - Дата Увольнение (Termination Date)
- Identification and Insurance:**
  - СНИЛС (SNILS)
  - ИНН (TIN)
  - Пенсионное удостоверение (Pension Certificate):
    - № (Справка) (Number (Certificate)), Дата Выдачи (Issue Date)
    - Кем Выдан (Issued by)
  - Журнал регистрации (Registration Journal):
    - Дата Регистрации (Registration Date)
    - Номер по Журналу (Journal Number)
  - Категория участия в программе (Program Participation Category) - dropdown menu
  - Выбор критерия (Criterion Selection):
    - Инвалидность (Disability) - dropdown menu
    - Иждивенцы (Dependents) - dropdown menu
- Work History and Calculation:**
  - Стаж работы в системе ОАО ХК "Якутуголь" (полных лет), в том числе (Length of service in the system of OJSC "Yakutugol" (full years), including):
    - Общий, в т.ч. (Total, including)
    - С РК 1,7 (With 1.7 years of special service)
    - С РК 1,3 (With 1.3 years of special service)
    - По СП.№1 (By SP No. 1)
    - По СП.№2 (By SP No. 2)
  - Ср/мес. зарплата участника для определения персональной надбавки (Average monthly salary of the participant for determining the personal allowance)
  - Выбор Периода (Period Selection) - dropdown menu

Рис. 3. Форма добавления участника

Разработанная система является полностью автоматизированной и способна обеспечить: формирование пакета документов на участника корпоративного негосударственного пенсионного обеспечения, расчет негосударственной пенсии, создание и работу с архивом документов, его безопасность и сохранность.

Список литературы:

1. RAD Studio – Embarcadero [Электронный ресурс]  
<https://www.embarcadero.com/ru/products/rad-studio> (дата обращения 22.05.17)
2. Свободная СУБД Firebird [Электронный ресурс]  
<http://bourabai.ru/dbt/servers/firebird.htm> (дата обращения 28.05.17)

## Обзор информационных систем для ведения бухгалтерского учета

*Королева В.В., студентка гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри  
E-mail: odinochka2504@yandex.ru*

*Научный руководитель:  
к.т.н., доцент кафедры МиИ Похорукова М.Ю.*

Ведение бухгалтерского учета – это трудоемкий и ответственный процесс, как на небольшом предприятии, так и в крупной организации. Зачастую бухгалтерский учет осуществляется вручную и занимает много рабочего времени, но развитие технологий позволяет автоматизировать этот процесс. На данный момент существует множество программ для ведения бухгалтерского учета, которые позволяют значительно облегчить труд бухгалтеров, минимизировать случайные ошибки и повысить производительность в целом.

Компания «1С» является автором многих программных обеспечений, в том числе и «1С: Бухгалтерия», которая на данный момент является одной из самых популярных программ для ведения бухгалтерского учета. «1С:Бухгалтерия» совмещает работу с MS Office, так же программный продукт хранит в себе множество отчетов, аналитики и программ (рис. 1):

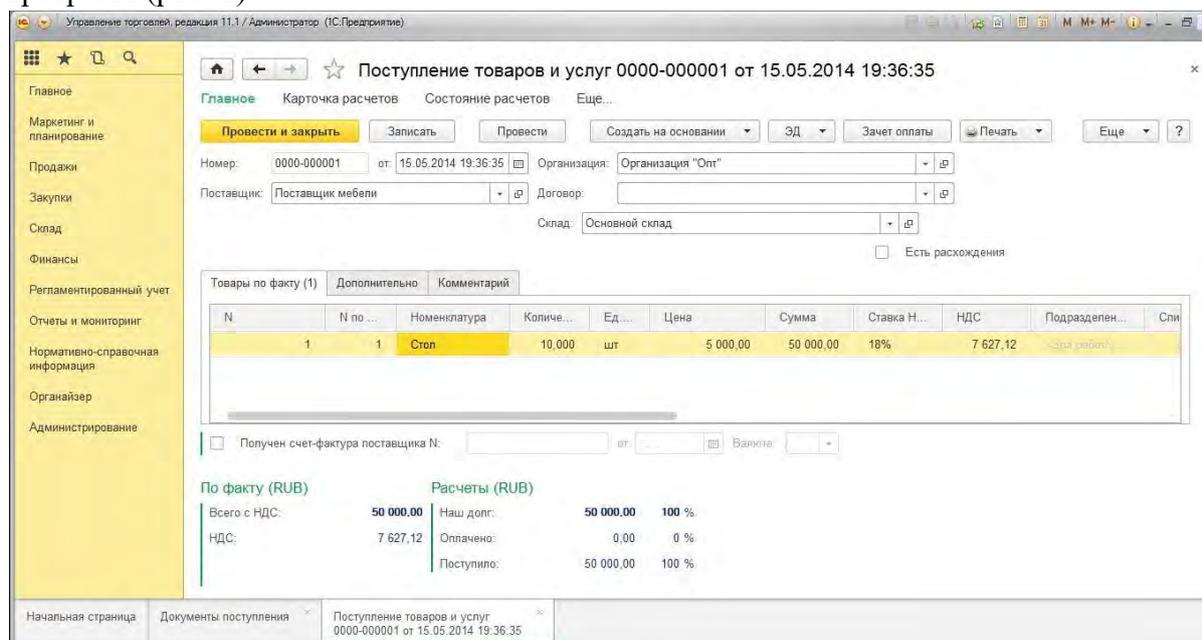


Рис. 1. Интерфейс программы «1С:Бухгалтерия»

Рассматривая плюсы программы «1С:Бухгалтерия», стоит отметить следующие основные моменты:

1. Программа предназначена для учета всех видов деятельности предприятия.
2. Не требуется предварительного обучения и специальных знаний.
3. Самые трудоемкие расчеты автоматизированы.
4. Обеспечивается строгое соответствие законодательству.
5. Проводится учет документации разных предприятий.
6. Имеется возможность корректирования показателей вручную.
7. Имеется встроенный сервис, благодаря которому отчетность можно отправлять через Интернет.

Некоторые считают, что доработка программы под каждое конкретное предприятие является большим минусом, однако, по факту, это является большим плюсом, поскольку имеется возможность настроить программу на работу конкретного предприятия, учесть все особенности, при необходимости дорабатывать в процессе использования. Единственный минус который можно выделить в данной программе это то, что здесь отсутствует функция построения графиков.

Среди отечественных производителей можно выделить такие программы для бухгалтерского учета, как «Галактика» и «Парус», а от зарубежных представителей более актуальны «SAP» и «Microsoft Dinamics AX (Navision Axapta)».

#### «Галактика»

Программный продукт с уверенностью можно назвать достойным аналогом 1С. Данная программа предоставляет пользователям электронный формат бухгалтерской отчетности и обеспечивает проведение электронных платежей через систему и многое другое (рис. 2):

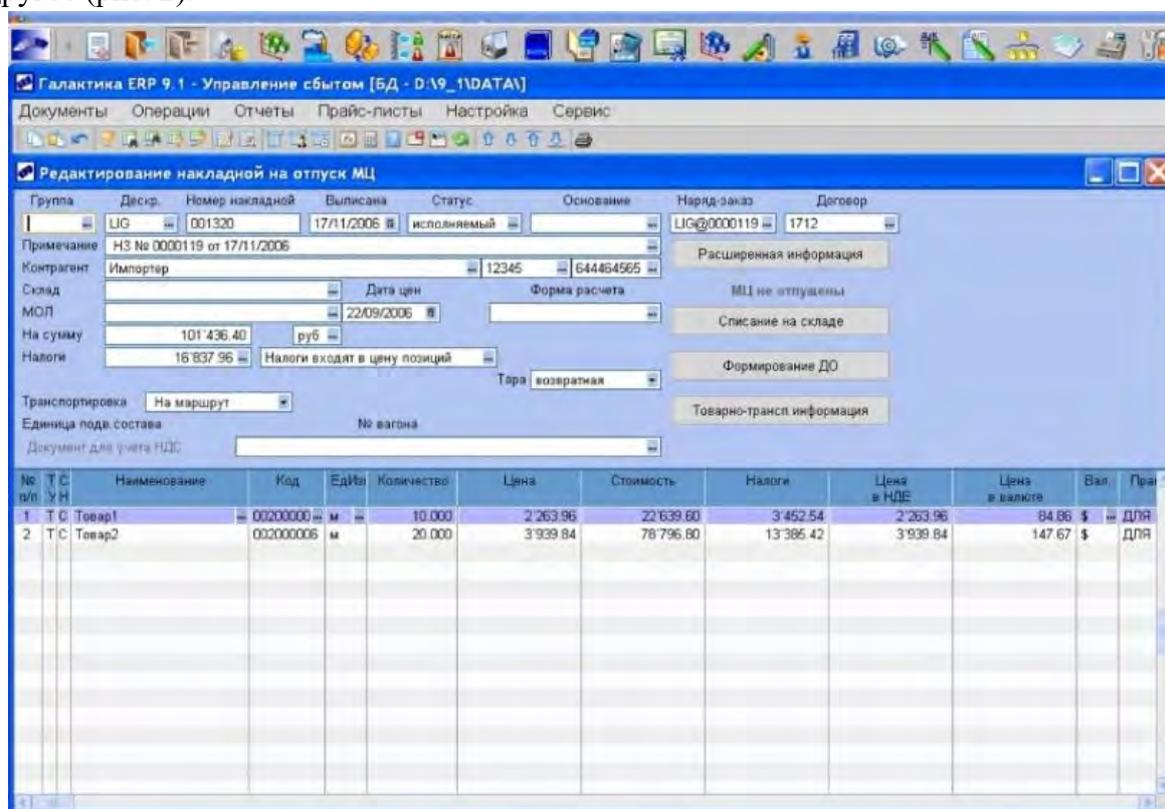


Рис. 2. Интерфейс системы «Галактика»

Эта программа имеет несколько минусов, которые заключаются в следующем:

- 1) не очень дружелюбный интерфейс (определенное сходство с Excel);
- 2) корректировка системы не возможна (кроме небольших исправлений интерфейса);
- 3) все доработки заказываются у разработчика ПО, что, в сравнении с 1С, выходит намного дороже и длительнее по времени;
- 4) внешняя интеграция с системой осуществляется с помощью средств COM, ODBC, XML, ActiveX.

#### «Парус»

Данная программа предназначена для малых и средних хозяйственных предприятий различной отраслевой принадлежности (торговля, сфера услуг,

элементарное производство, реклама и СМИ, общественное питание, туризм, иностранные компании и др.). Это простая, удобная, но в то же время мощная полнофункциональная система, позволяющая автоматизировать бухгалтерский учет, основные торговые процессы и складской учет, расчет заработной платы и кадровый учет. Может эксплуатироваться как на одном, так и на нескольких (в пределах 15-20) объединенных в локальную сеть компьютерах.

Можно выделить следующие особенности программы:

- 1) симпатичный интерфейс (лучше, чем у системы «Галактика»);
- 2) присутствует возможность масштабирования решения;
- 3) имеет модульную систему (MRP, бухгалтерия, CRM, финансы и т.д);
- 4) программу дорабатывают только разработчики (принцип использования «как есть»);
- 5) установка происходит в базу данных Oracle, что, при использовании лицензионного ПО, достаточно дорого.

Интерфейс рассматриваемой программы отображен на рисунке 3:

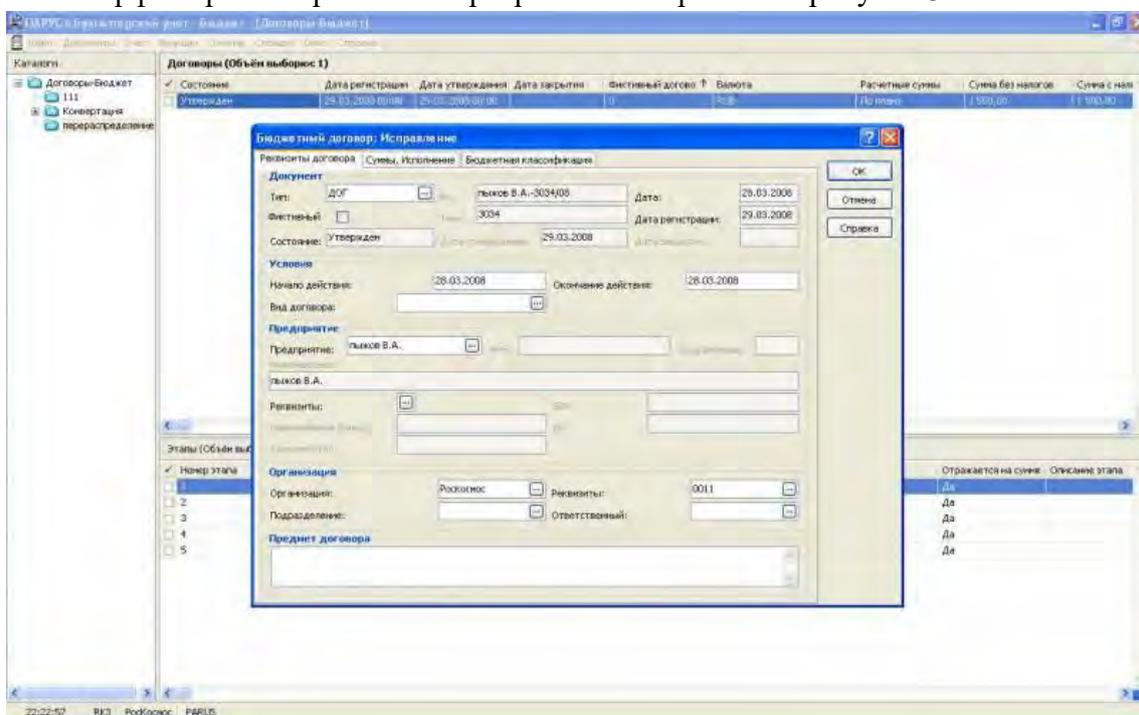


Рис. 3. Интерфейс системы «Парус»

### «SAP»

Бухгалтерская программа SAP является автоматизированной системой, позволяющей планировать ресурсы крупных предприятий, которая позволяет рассчитывать все до мелочей, а также формирует единое информационное пространство.

Особенности этого программного продукта заключаются в следующем:

- 1) хорошая репутация и высокая популярность среди ERP-систем в мире;
- 2) существует достаточно большое количество модулей различного назначения;
- 3) хорошо масштабируемое решение;
- 4) высокая цена на услуги и лицензии (в 3-10 раз выше по сравнению с 1С), поэтому внедрение происходит в основном на крупных предприятиях;
- 5) возможная, но трудоемкая доработка системы, поэтому в большинстве случаев SAP только настраивают.

Интерфейс рассматриваемой программы отображен на рисунке 4.

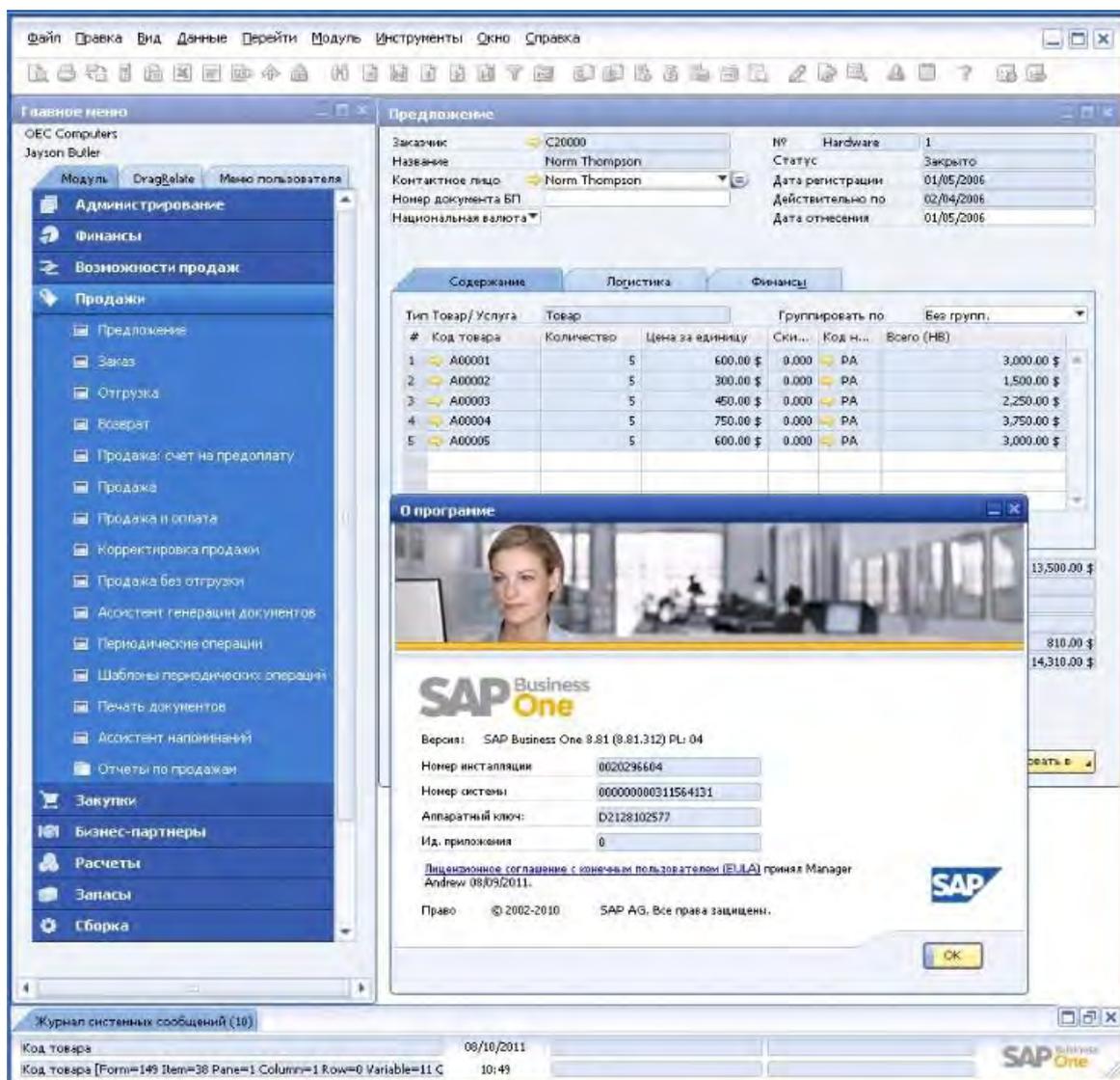


Рис. 4. Главное окно программы «SAP»

#### «Microsoft Dynamics AX (Navision Axapta)»

Функционально решение охватывает все области менеджмента предприятия: управление производством, дистрибуцией в сложных цепочках поставок, розничными сетями, финансами, включая учёт по различным стандартам в холдинговых структурах, проектной деятельностью и сервисным обслуживанием, продажами, маркетингом, взаимоотношениями с клиентами, управление персоналом, а также осуществляет контроль и анализ бизнеса, соответствие корпоративным политикам.

Можно выделить следующие характеристики программы:

- 1) значительное сходство интерфейса с Excel-таблицами;
- 2) не очень большое распространение на российском рынке;
- 3) наличие всех современных модулей (MRP, HR, CRM и т.д.);
- 4) интегрирование с программами от Microsoft (Outlook, Excel и т.д.);
- 5) трудоемкая доработка софта, при условии малого количества специалистов на российском рынке.

Интерфейс рассматриваемой программы отображен на рисунке 5:

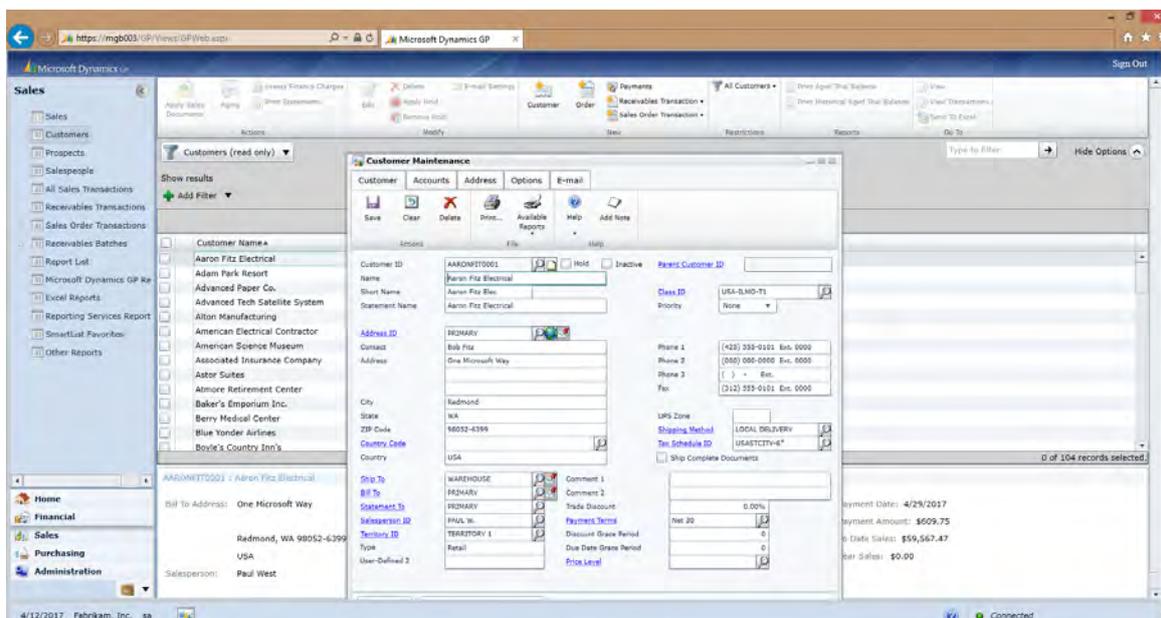


Рис. 5. Интерфейс «Microsoft Dynamics AX (Navision Axapta)»

Проведенный анализ программных продуктов позволяет сделать вывод о том, что на рынке программных продуктов существует множество решений для автоматизации бухгалтерского учета. Причем каждое из них является вполне конкурентоспособным и может быть использовано в любой организации. Однако выбор в пользу той или иной программы осуществляется на предприятии индивидуально, в зависимости от масштаба предприятия, поставленных целей и возможностей.

### Создание имитационной модели для расчета необходимого количества зарядных станций в торговом центре

*Кондратьев Н.А., студент гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри  
E-mail: nikkondr-ykt@yandex.ru*

*Научный руководитель:  
к.п.н., зав. кафедрой МиИ Самохина В.М.*

Имитационная модель – это метод исследования, при котором изучаемая система заменяется моделью, с достаточной точностью, описывающей реальную систему, с которой проводятся эксперименты с целью получения информации об этой системе.

А компьютерная имитационная модель — это построения эксперимента с помощью электронно-вычислительной машины (ЭВМ).

Применение компьютерной имитационной модели дает множество преимуществ перед обычной моделью как:

1. Стоимость. Создание компьютерной имитационной модели гораздо выгодно в денежном плане чем простая имитационная модель, поскольку оно требует лишь покупки программного средства и консалтинговых услуг.

2. Время. В реальности оценить изменения, какого-либо параметра на рынок потребуются длительное время, месяцы, годы. А с имитационной моделью на это потребуются считанные минуты.

3. Повторяемость. С имитационной моделью возможно создать и провести эксперимент неограниченное количество раз с разными параметрами, для определения

наилучшего варианта.

4. Наглядность. Компьютерная имитационная модель обладает возможностью визуализации процесса работы процесса. Что позволяет лучше понять систему.

5. Универсальность. Компьютерное имитационное моделирование позволяет решать задачи любых областей за счёт большого инструментария и возможности для изменения задачи.

Для расчетов установки зарядных станции такая модель была бы весьма полезна для проведения нескольких экспериментов с разными параметрами зарядных станций для расчета наиболее эффективного их количества для организации.

Например, при создании имитационной модели для торгового центра состоящую из 2-х этажей, в которых есть сеть магазинов кинотеатр и кафе.

Для начала нужно создать модель человека. Учесть там такие параметры как:

- какой у него телефон;
- уровень заряда телефона;
- время хода клиента из торгового центра;
- предпочтения клиента при выборе места пребывания.

Далее на рисунке 1 представлена схема разрабатываемой имитационной модели.

Именно в нем производятся действия, такие как выбор клиента места назначения, использования зарядных станций, продолжительность пребывания, уровень заряда батареи мобильного устройства, сбор статистики для модели и т.д.



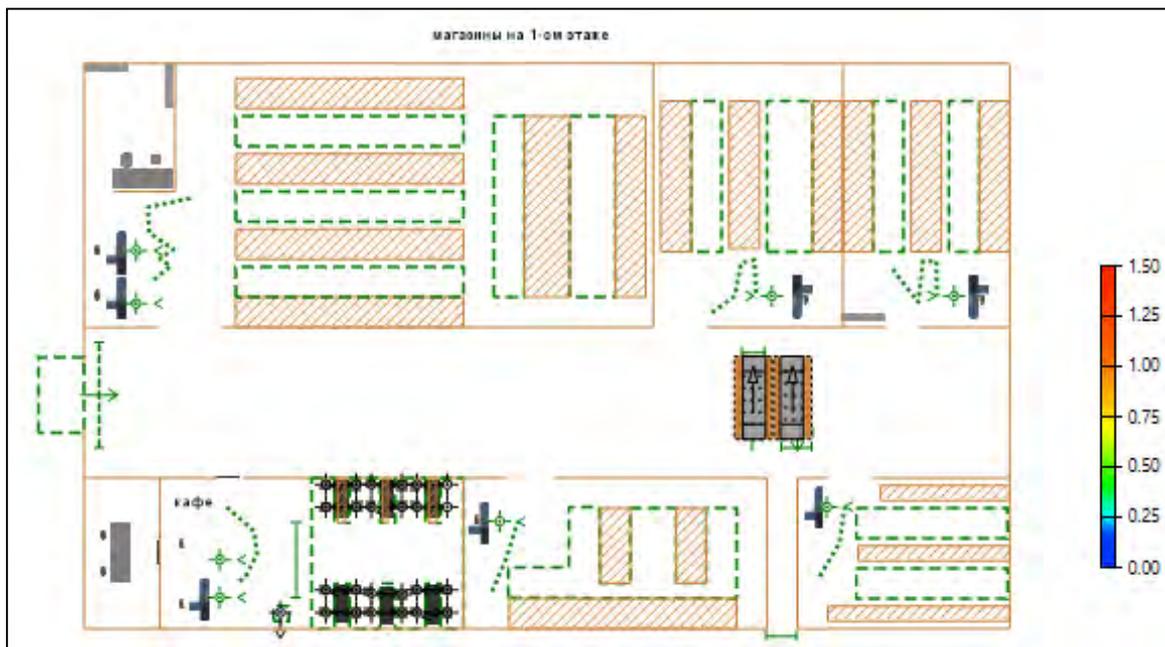


Рис. 2. Первый этаж модели ТЦ

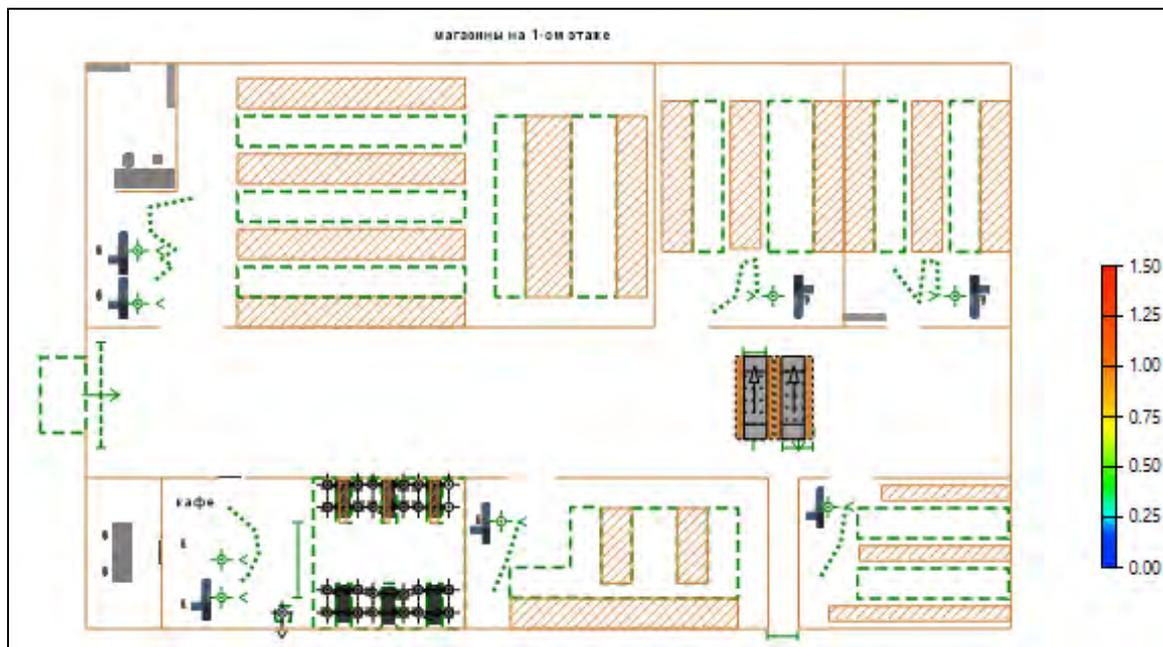


Рис. 3. Второй этаж модели ТЦ

При запуске модели просчитывается работа торгового центра (рис. 4).

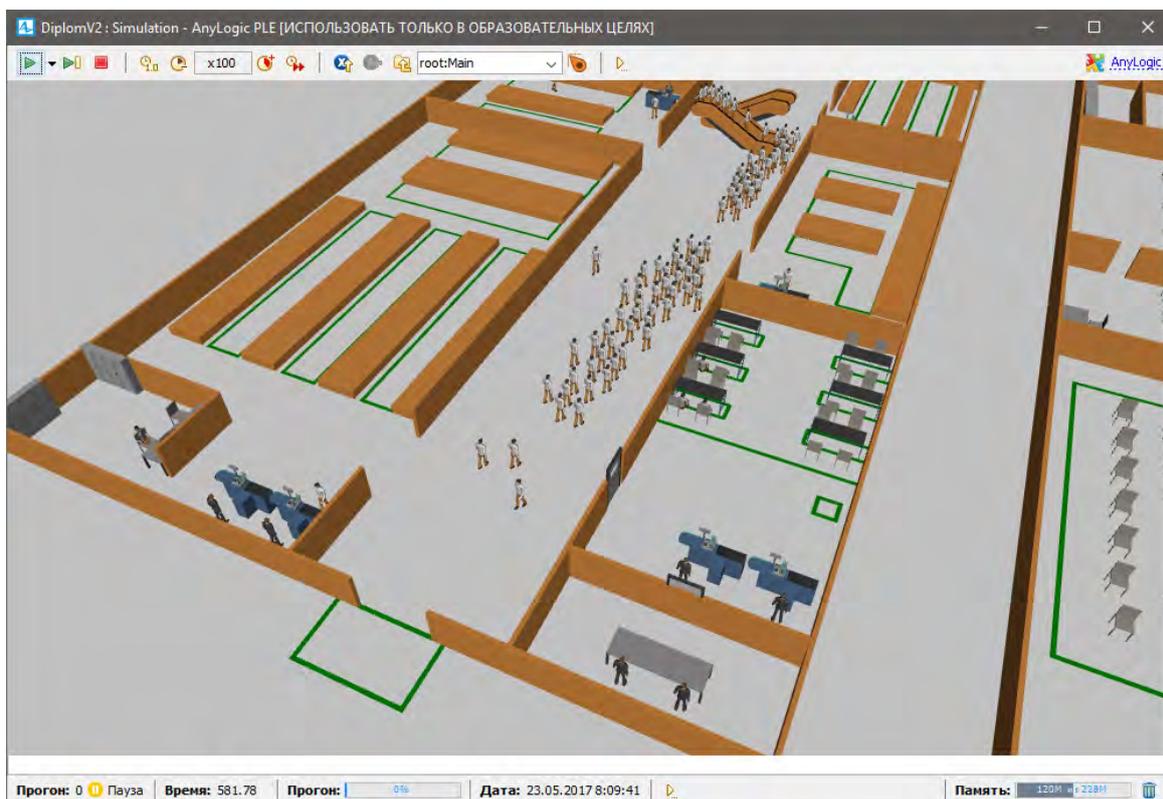


Рис. 4. 3D модель торгового центра

Такие модели весьма полезны для организации при распространении и для расчетов количества станций для организаций-клиентам. Они отличаются своей гибкостью при выборе параметров, всегда можно отключить какой ни будь из параметров при расчете или изменить количество станций и тут же просмотреть его влияние на результат работы модели.

В заключении хотелось бы сказать, что зарядные станции в современном мире необходимы, поскольку большинство людей очень часто используют мобильные устройства и они часто заряжаются при его активном использовании. А зарядные станции привлекут бы клиента, например, в кафе с установленной там зарядной станцией люди пойдут охотнее чем с его отсутствием. И возможность размещения в них рекламы тоже плюс.

### **Зарядные станции для мобильных устройств**

*Кондратьев Н.А., студент гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри*

*E-mail: nikkondr-ykt@yandex.ru*

*Научный руководитель:*

*к.п.н., зав. кафедрой МиИ Самохина В.М.*

В современном мире мобильный телефон является неотъемлемой частью жизни человека. Более 90% населения России используют мобильные устройства, а некоторые имеют даже несколько устройств. И по данным компании Kantar Worldpanel 80% и сайта E-Katalog мобильных устройств используют устройства на ОС Android, и 17% на ОС IOS. Основная проблема, возникающая при работе таких устройств, это быстрая разрядка при его активном использовании. При этом возможность зарядки телефона

есть не всегда, так как не каждый человек носит с собой зарядное устройство. Чаще всего такая потребность возникает в общественных местах, и установка зарядных станций была бы полезна как для клиента, так и для организаций для привлечения и удержания клиента.

По данным сайта GetCharge у опрошенных людей:

- 89% людей понравились зарядные устройства;
- 79% поделится со своими друзьями о новом способе зарядки телефона;
- 75% предпочтут пойти обедать туда, где можно комфортно зарядить телефон.

Зарядные станции для мобильных устройств можно условно разделить на несколько видов такие как:

Настольные зарядные станции – могут как работать автономно, без подключения к электросети, так и через электросеть. Подходят для размещения в сервисных центрах или для мест общественного питания. Из минусов можно выделить то что этим станциям требуется достаточно места для установки на столе. Внешний вид представлен на рисунке 1.



Рис. 1. Настольная зарядная станция

Портативные зарядные станции – представляют собой портативные аккумуляторы для подзарядки устройства, заряжающиеся на зарядные станции (рис. 2.). Оптимальное решение для кафе, ресторанов, отелей, баров поскольку клиенту не надо оставлять свое устройство возле станции, и он может продолжать его использовать. Из минусов следует отметить поскольку это аккумулятор ему требуется время для полного заряда.



Рис. 2. Портативные зарядные станции

Настенные или напольные зарядные станции предназначены для установки на пол или стену, более подойдут для расположения в местах с большим трафиком таких как торговые центры, аэропорты, вокзалы (рис. 3.). Могут иметь встроенные дисплей для размещения рекламы или воспроизведения видео.

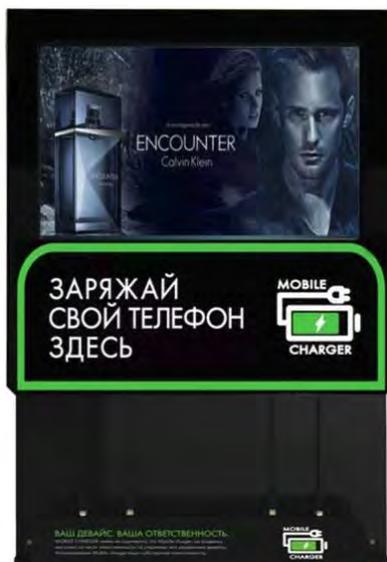


Рис. 3. Настенная зарядная станция

Автоматы – представляет собой сейф для хранения и зарядки мобильного устройства, внешний вид представлен на рисунке 4. Подойдет для мест где использование мобильного устройства не предполагается такие как фитнес-центры, спорт-центры, бассейны, кинотеатры и т.д. Защищенные шкафы обеспечат безопасность не только для мобильного устройства, но и для других мелких вещей.



Рис. 4. Автомат для зарядки

Такие зарядные станции также привлекут новых клиентов, поскольку человеку трудно будет пройти мимо заведения где можно не просто выпить чашку кофе, но и зарядить «умирающий» телефон. И, следовательно, это повлияет увеличение продаж за счет привлечения новых клиентов и за счет того, что люди будут дольше находиться в заведении. Это повлияет за собой увеличение лояльности клиентов, они, например, могут посоветовать заведение своим друзьям. В будущем люди с большим удовольствием пойдут в заведение где есть такое устройство.

#### **Сравнительная характеристика кроссплатформенных фреймворков для мобильной разработки**

*Морозов Д.Е., студент гр. ПМ-13, ТИ (ф) СВФУ, г. Нерюнгри*

*E-mail: morozovd96@outlook.com*

*Научный руководитель:*

*к.п.н., зав. кафедрой МиИ Самохина В.М.*

Так как каждый телефон имеет встроенный браузер, воспринимающий HTML, JavaScript и CSS стили, то это позволяет создавать приложения, которые, как и веб - сайты, работают через механизм встроенного браузера и не требуют от разработчика знания специальных, нативных языков.

HTML – это язык разметки гипертекста. Данный язык применяется для создания

веб-страниц. Он обрабатывается браузером и отображается в виде документа в удобной для восприятия форме. Язык HTML - это логической разметки страницы. Именно HTML позволяет наделять содержимое страницы информацией, а реализуется это с помощью тэгов.

CSS – это язык описания внешнего вида документа, написанного с использованием языка разметки. Название произошло от англ. Cascading Style Sheets — каскадные таблицы стилей. Язык CSS предназначен для того, чтобы создавать необходимый внешний вид HTML-документам.

JavaScript - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript (стандарт ECMA-262). JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Этот язык программирования, наиболее широко применяется в браузерах для придания интерактивности веб-страницам.

Лидерами по количеству реализованных продуктов и числу разработчиков, а так же открытыми и хорошо документированными являются:

Appcelerator Titanium – платформа для создания мобильных и десктопных мультиплатформенных приложений на JavaScript (HTML+CSS), поддерживает всего три платформы – Android, iOS и BlackBerry. Основные достоинства платформы – это полная автоматизация процесса тестирования, подробная информация об ошибках в исходном коде и способы их устранения, большой магазин плагинов, развитая аналитическая платформа. Titanium это не только SDK, но и облачная платформа для сборки и распространения приложений.

Kony Platform – поддерживает полный цикл создания приложений, включающий дизайн, разработку, тестирование, поэтому процесс разработки можно распараллелить между несколькими программистами. В отличие от Titanium имеет возможности настройки прав доступа и является более гибкой с возможностью выбирать фреймворки, есть система отчетов и логирования.

Adobe PhoneGap – технология для реализации кроссплатформенной разработки мобильных приложений. Позиционируется как бесплатный фреймворк с открытым кодом, позволяет благодаря кроссплатформенности разрабатывать приложения 8 мобильных платформ. При этом код может писаться на простых веб - языках, таких как HTML, JavaScript, CSS.

Из представленных программных решений был сделан выбор в пользу web-фреймворков. Такой выбор был обусловлен возможностью разработки в них на едином языке (без использования нативных для платформ аспектов).

В сравнительной характеристике ниже разобраны наиболее функциональные среды по основным критериям кроссплатформенной разработки.

В таблице 1 представлено сравнение популярных фреймворков для кроссплатформенной разработки по наличию собственной IDE, плагинов к популярным IDE и других инструментов разработки.

Таблица 1

## Сравнительная характеристика фреймворков по инструментам разработки

Appcelerator Titanium	Titanium Studio, основанная на Eclipse.
Kony Platform	Основанная на Eclipse Kony Studio (Kony Visualization Cloud – для дизайна и прототипирования, Kony Development Cloud – разработка, Kony Management Cloud – управление приложениями).
Adobe PhoneGap	Собственной среды разработки нет, написание JS кода ведется в любом текстовом редакторе. Есть интеграция с Adobe Dreamweaver и Brackets. Также есть приложение «Developer App», которое позволяет мгновенно смотреть изменения на мобильном устройстве, без необходимости компилировать разрабатываемое приложение и собирать его.
IBM Worklight	Worklight Studio – плагин для Eclipse, превращающий его в среду разработки на Worklight.
Telerik Platform	web IDE (так называемые ICE – integrated cloud environment). Состоит она из 5 модулей для всех этапов разработки: <ul style="list-style-type: none"> <li>• AppPrototyper (дизайн и прототипирование)</li> <li>• AppBuilder (разработка)</li> <li>• AppManager (управление)</li> <li>• AppFeedback (обратная связь с пользователями)</li> <li>• AppAnalytics (соответственно система аналитики)</li> </ul> Кроме того, есть расширение для Visual Studio, клиент для Windows, расширение для Sublime Text.
Verivo Akula	Есть своя IDE Verivo AppStudio. Также для разработки можно использовать любую привычную среду (IDEA, Eclipse, NetBeans и др.)
Xamarin	Есть своя IDE Xamarin Studio. Также есть возможность разрабатывать Visual Studio, установив специальный плагин.

В таблице 2 представлены поддерживаемые платформы рассматриваемых фреймворков.

Таблица 2

## Поддерживаемые платформы

Appcelerator Titanium	Нативные приложения под мобильные ОС: iOS, Android, BlackBerry, Tizen, Denso. Mobile-web приложения под те же платформы. Десктопные приложения под Windows, Mac.
Kony Platform	Десктопные приложения под Windows, Mac. Веб-приложения для браузеров IE 8+, Opera 12+, Safari 5+, Firefox 4+ Chrome 20+. Приложения для смартфонов и планшетов — iOS, Android, Windows Phone, Blackberry.
Adobe PhoneGap	Mobile-web приложения: iOS, Android, Windows Phone, Blackberry, Ubuntu, Firefox OS.

IBM Worklight	Мобильные нативные и mobile-web: Android, iOS, Blackberry, Windows Phone, Java ME. Десктопные: Windows 8.
Telerik Platform	Мобильные: iOS, Android, Windows Phone. Десктопные: Windows 8.
Verivo Akula	Нативные и mobile-web: iOS, Android.
Xamarin	Мобильные нативные приложения: iOS, Android, Windows Phone. Десктопные нативные – Windows, Mac.

Таблица 3

Языки разработки.

Appcelerator Titanium	Kony Platform	Adobe PhoneGap	IBM Worklight	Verivo Akula	Telerik Platform	Xamarin
JS, Python, Ruby, PHP	Нативные языки (Java, Objective-C, C#) — для родных приложений и гибридных решений, для кроссплатформенных - JavaScript				.NET, Java, JS, PHP	Только C#

Список литературы:

1. Основы JavaScript и Node.js [Электронный ресурс] URL: <http://dotnet-am.livejournal.com/3953.html> (дата обращения: 01.05.2017)
2. Дакетт Дж. Javascript и jQuery. Интерактивная веб-разработка. М.:Эксмо, 2017. 640 с.
3. Краткая характеристика языка программирования Python, Ruby и PHP [Электронный ресурс] URL: [http://www.internet-technologies.ru/articles/article\\_1991.html](http://www.internet-technologies.ru/articles/article_1991.html) (дата обращения: 18.04.2017)

**Проектирование информационной системы для автоматизированного расчета платежных поручений в АО «Газпромбанк» г. Нерюнгри**

*Попова М.Н., студентка гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри  
E-mail: ximera.chan@gmail.com*

*Научный руководитель:  
к.т.н., доцент кафедры МиИ Похорукова М.Ю.*

В настоящий момент большая часть деятельности банковских организаций осуществляется на компьютерах в различных базах данных и системах расчёта и калькулирования. Несмотря на высокую автоматизацию в этой сфере, существует необходимость ежемесячной проверки данных даже в самых простых расчётах, таких как, комиссия за денежные переводы по расчётному (платёжному) документу, пересчёт комиссии в зависимости от использования услуги денежного перевода организациями. Использование в расчётах платёжных поручений по возможностям их осуществления разнообразны. Применяя их можно совершать расчёты в хозяйстве по товарным или нетоварным операциям, при этом все без исключения нетоварные платежи происходят только с помощью платежных поручений.

Платёжные поручения представляют собой осуществляемые банком

распоряжения о переводах определённых средств на счёт получателя от вкладчика (плательщика). При расчётах платёжными поручениями банк обязуется по поручению плательщика за счёт средств, находящихся на его счетах, перевести конкретную валютную необходимую сумму на счёт указанного плательщиком лица в этом или ином банке.

Платёжные поручения по договоренности сторон могут быть срочные и досрочные.

Срочные платёжные поручения применяются в последующих вариантах:

- авансовый платёж, то есть взнос до поставки товара, работ, услуг;
- платёж после отгрузки товара, то есть посредством прямого акцепта товара;
- частичные платежи при крупных сделках.

Платёжное поручение (рис. 1) может быть оплачено полностью или частично при отсутствии денег на счёте плательщика, о чем делается отметка на платёжном документе.

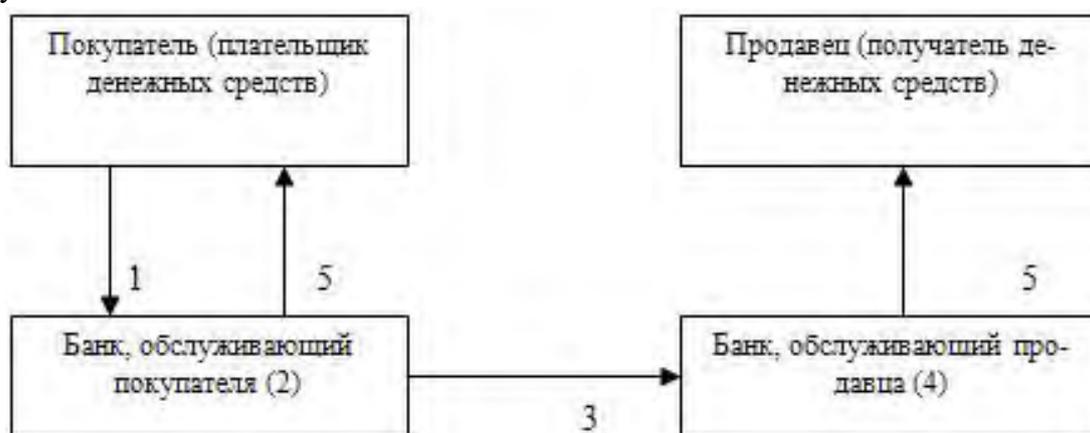


Рис. 1. Схема расчетов платёжными поручениями

1. Покупатель (плательщик денежных средств) предоставляет в банк платёжное поручение в четырех (или пяти) экземплярах и получает назад четвертый экземпляр в качестве расписки банка.

2. Банк, обслуживающий покупателя, на основании первого экземпляра платёжного поручения списывает денежные средства со счета покупателя.

3. Банк, обслуживающий покупателя, направляет в банк, обслуживающий продавца, два экземпляра платёжного поручения и денежные средства.

4. Банк, обслуживающий продавца, используя второй экземпляр платёжного поручения, зачисляет денежные средства на счет продавца (получателя денежных средств).

5. Банки выдают своим клиентам выписки из расчётных счетов.

Актуальность разработки информационной системы заключается в создании инструмента, который будет использоваться контроллером или специалистом из отдела обслуживания юридических лиц, которая обеспечит автоматизированный ежемесячный расчёт количества платёжных поручений, тем самым сократив возможные случайные ошибки и повысив производительность сотрудников.

Программа должна поддерживать работу с базой данных, которая содержит информацию о расчётных данных комиссий за денежные переводы для всех организаций-клиентов банка. Условия для выполнения расчётов или данные, которые должны присутствовать в программе:

1. ежемесячная плата за обслуживание с использованием системы «Клиент-Банк» равняется 600 руб.;

2. ведение банковского счёта в валюте РФ по счёту за отчётный месяц (взимается ежемесячно):

- с использованием системы «Клиент-Банк» равняется 500 руб.;
- без использованием системы «Клиент-Банк» равняется 1500 руб.

3. учет количества переводов денежных средств с банковского счёта Клиента на счёт в другом банке по расчётному (платёжному) документу:

- при получении документа на бумажном носителе, тариф – 150 руб.;
- при получении документа в электронном виде, тариф – 30 руб.

Разрабатываемое программное обеспечение должно иметь следующие возможности:

- возможность самовосстановления после сбоев (отключения электропитания, сбой в операционной системе ит. д.);
- парольную защиту при запуске программы и разграничение пользовательских прав;
- ограничение несанкционированного доступа к данным;
- возможность резервного копирования информационной базы.

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

1. Разделение пользователей системы на группы с соответствующими правами доступа: администратор и оператор.

2. Возможность ввода, редактирования, поиска и отбора необходимой информации из базы данных с использованием поисковой системы: по коду организации или по наименованию.

3. Формирование отчетности в виде электронных таблиц MSExcel и документов MSWord.

Созданный автоматизированный программный продукт позволит снизить трудозатраты и число ошибок специалиста предприятия при обработке данных, а также он приведет к экономии времени сотрудников и повысит оперативность. Централизованное хранение данных повысит точность и достоверность предоставляемых сведений.

### **Мониторинг горных подстанций**

*Протопопов В.А., студент гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри*

*E-mail: [pvjester@gmail.com](mailto:pvjester@gmail.com)*

*Научный руководитель:*

*старший преподаватель кафедры МиИ Соболева Н.И.*

Создание систем мониторинга в настоящее время является актуальной задачей для множества промышленных предприятий. Эта система позволяет постоянно наблюдать за работой различных подсистем в режиме реального времени посредством мониторинга инженерных систем осуществляется удаленный контроль и управление различными процессами, изменение рабочих параметров тех или иных устройств и компонентов, передача данных об их состоянии и ведение протоколов и баз данных со сведениями об их работе [1].

Комплектным распределительным устройством (КРУ) называется устройство, состоящее из шкафов, в которых смонтированы коммутационные аппараты, устройства, защиты, автоматики и телемеханики, измерительные приборы и вспомогательные устройства, поставляемые на место установки комплектно в собранном и полностью подготовленном для сборки виде.

КРУ предназначено для приема и распределения электроэнергии переменного трехфазного тока промышленной частоты 50 Гц, напряжением 6 и 10 кВ [2].

Комплектные распределительные устройства взрывозащищённые КРУВ-6М применяются в шахтах, опасных по пыли и газу, и предназначены для:

- распределения электрической энергии напряжением 6 кВ;
- защиты отходящих электрических сетей от аварийных процессов;
- управления токоприемниками (в том числе по цифровым интерфейсам);
- выполнения функций автоматики (в том числе в составе распределительных пунктов).

Целью создания подсистемы мониторинга позволяет специалистам АО ГОК «Денисовская» повысить надежность и качество электроснабжения потребителей за счет сокращения количества аварий путем их предупреждения и локализации, а также исключения ошибочных действий оперативного и обслуживающего персонала.

В интегрированной инструментальной системе проектирования и разработки ArchestrA IDE создаем арию VRP\_3 (рис.1.) и добавляем в нее объекты шаблонов и арии КРУВ-ов.

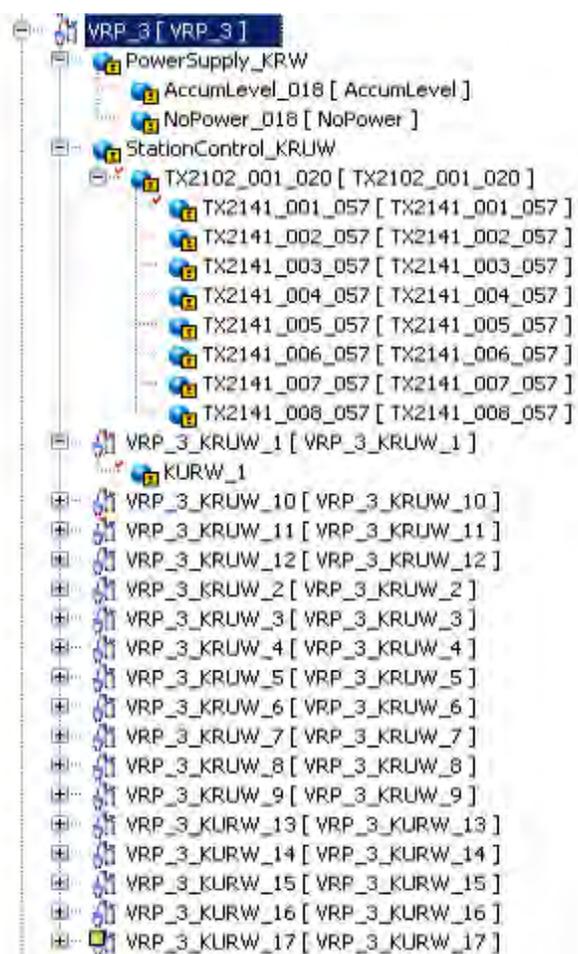


Рис. 1. Готовая ария ВПИ №3

Шаблон блока питания PowerSupply\_KRW включает следующие шаблоны:

- шаблон датчика заряда аккумулятора AccumLevel\_18;
- шаблон датчика наличия питания NoPower\_18.

Шаблон станции управления и сбора данных StationControl\_KRUW содержит один (два) шаблон(а) терминального модуля:

- TX2102\_001 – вложенное имя терминального модуля;
- TX2141\_001 – вложенное имя модуля ввода-вывода.

Шаблон дискретного датчика KURW предназначен для считывания данных с подстанции.



Рис. 2. Главное меню ВРП №3

На рис. 2 представлена главное меню временно распределительных пунктов №3, каждая отвечает за определенную установку.

Подсистема «Мониторинг горных подстанций» выполняет следующие функции:

- отображение состояния комплектных распределительных устройств;
- фиксация переключений за смену, печать состояния ячеек, список аварийных и предупредительных сообщений;
- квитирование предупредительных и аварийных сигналов;
- отображение таблицы параметров, вышедших за предупредительные и аварийные пределы;
- диагностика, состояние и контроль прохождения сигналов на верхний уровень диспетчерского управления (IEC 60870-5-101, IEC 60870-5-104);
- конфигурирование предупредительной и аварийной сигнализации;

Данная подсистема в дальнейшем будет улучшаться за счет добавления неиспользованных КРУВ-ов, а также удаленно управлять питанием.

#### Список литературы:

1. Электроэнергетика: [Электронный ресурс] Оборудование КРУВ 6М.-. режим доступа к стр.: <http://forca.com.ua/info/oborudovanie/kruv-6v.html>
2. Котлярский А.И., Жигарловский Я.А. Проектная документация «Многофункциональная система безопасности» Том 1 «Общесистемные решения, информационное и техническое обеспечение» Книга 1 «Текстовая часть» М.: ПРОМТЕХ, 2014 60с
3. Эволюция InTouch [Электронный ресурс]. Режим доступа: [http://media.klinkmann.ru/catalogue/ru/Wonderware/Wonderware\\_Press\\_InTouch\\_Evolution\\_MirAvtomatisazii\\_04-08\\_ru\\_0111.pdf](http://media.klinkmann.ru/catalogue/ru/Wonderware/Wonderware_Press_InTouch_Evolution_MirAvtomatisazii_04-08_ru_0111.pdf) (дата обращения 27.04.17)

## **Автоматизированная информационная система для учета выплат студентам СВФУ. Выбор средств разработки**

*Простакишин С.В., студент гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри  
E-mail: sergei-prostakishin@mail.ru*

*Научный руководитель:  
старший преподаватель кафедры МиИ Соболева Н.И.*

Как известно, оптимизация рабочего процесса и повышение продуктивности сотрудников положительно сказывается на качестве выполняемой ими работы. Этого можно достичь различными способами, но в современных реалиях одним из самых простых и эффективных путей развития предприятия является автоматизация его инфраструктуры с использованием информационных систем.

Перед тем как начать проектировать и разрабатывать систему, необходимо выявить задачи, подверженные автоматизации, а также уровень и состав информационно-технической базы. В результате проведенного исследования необходимая информация была собрана. Отдел предъявляет следующие требования к программному продукту:

- хранить данные о студентах, которым назначены или будут назначены выплаты, а именно: Ф. И. О., дату рождения, группу, состояние студента (отчислен/обучается), адрес, номер телефона, email, серию и номер паспорта, ИНН, номер ПСС и статус проживания в общежитии;
- иметь классификатор расходов согласно положению о выплатах, иметь функции по редактированию данного классификатора;
- хранить данные о материальных выплатах студентам: студент группа, адрес прописки, вид выплаты (единовременная/ежемесячная), дата подачи всех документов, статья расходов (согласно классификатору), выделенная сумма, а также поля о наличии у студента необходимых документов, а именно: заявления, копий паспорта, ИНН и подтверждающих право на выплаты документов;
- обеспечивать функции поиска и сортировки по вышеописанным данным и полям;
- иметь формы поиска и выборки по следующим критериям: выплаты по студентам за определенную дату, выплаты по виду за определенную дату и выплаты по статье расходов за определенную дату;
- обеспечивать вывод отчетности на основе выборки из предыдущего пункта в документы форматов Microsoft Word (docx) и Microsoft Excel (xlsx);
- иметь простой и понятный для конечного пользователя интерфейс.

С точки зрения разработчика, данные задачи можно автоматизировать с помощью внедрения в структурное подразделение информационной системы на основе СУБД и приложения для простой и комфортной работы сотрудников комплекса с данной СУБД. Максимально быстро достичь качественной программной реализации вышеуказанных требований можно, используя среду разработки Microsoft LightSwitch. Выбор данной среды основан на том, что данная среда изначально поддерживает интеграцию с другими решениями Microsoft, поставляется вместе с бесплатным сервером Microsoft SQL Server Express Edition. В среде используются современные технологии разработки приложений: такие как: мощный визуальный редактор интерфейса и структуры базы данных, язык запросов LINQ, виртуальная среда

исполнения Microsoft .NET Framework, система редактирования интерфейса во время исполнения программы. Кроме того, нельзя упускать из внимания тот факт, что среда разработки Microsoft LightSwitch была специально разработана для создания небольших и средних по функционалу приложений типа оболочка-СУБД, и имеет все необходимое для того, чтобы максимально сократить время на разработку подобных приложений.

Преимущества среды:

- возможность объектно-ориентированного подхода для работы с СУБД;
- возможность расширения функционала среды разработки путем установки дополнительного инструментария и плагинов из официального репозитория Microsoft;
- возможность редактировать пользовательский интерфейс во время отладки программы, без необходимости перекомпиляции.

Недостатки среды:

- низкая скорость компиляции и сборки;
- небольшой выбор языков программирования;
- относительно малое количество документации на русском языке, что может вызвать затруднения у не англоговорящих разработчиков.

Список литературы:

1. Избачков Ю.С., Петров В.Н. Информационные системы: Учебник для вузов Ю.С. Избачков, В.Н. Петров. – 2-е изд. – СПб.: Питер, 2006. – 656 с.
2. Дейт К. Дж. Введение в системы баз данных – 7-е изд. – Вильямс, 2001. – 1072 с.

#### **Разработка информационной системы для оптимизации деятельности отдела по внеучебной работе ТИ (ф) СВФУ**

*Простакишин С.В., студент гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри*  
*E-mail: sergei-prostakishin@mail.ru*

*Научный руководитель:*  
*старший преподаватель кафедры МиИ Соболева Н.И.*

В последнее время автоматизация учетной и офисной деятельности является одной из основных движущих сил в развитии глобальной индустрии информационных технологий. Практически каждый из гигантов информационной индустрии – разработчиков программного и аппаратного обеспечения внес свой вклад в развитие этой отрасли рынка информационных технологий. Результат автоматизации офисной деятельности – снижение затрат на выполнение рутинных операций и сокращение ошибок, вызванных человеческим фактором, что в конечном итоге дает возможность управленцам принимать взвешенное и своевременные решения, что положительно сказывается на эффективности предприятия в целом. Для достижения этих целей был разработан программный продукт для автоматизации некоторых аспектов учета выплат студентам отделом по внеучебной работе ТИ (ф) СВФУ.

Программа разработана на языке программирования Visual C# средствами среды разработки Microsoft Visual Studio Lightswitch. В качестве сервера базы данных используется Microsoft SQL Server редакции Express. Однако, при необходимости, можно использовать и другие источники данных, например, полноценную версию SQL

Server или Windows Azure.

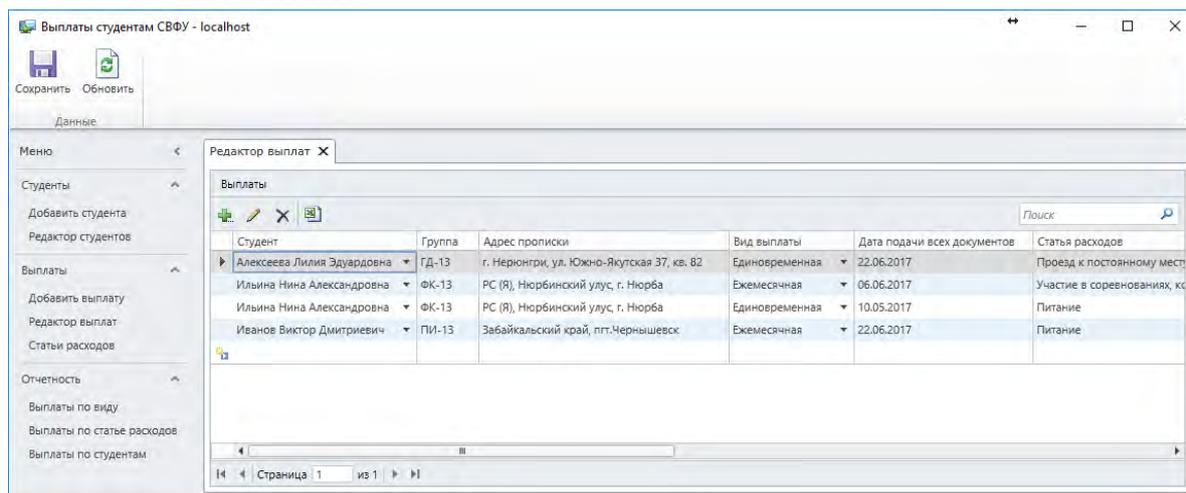


Рис. 1. Главное окно программы

Принцип работы данной программы аналогичен другим подобным решениям на рынке: пользователь ведет учет студентов и выплат путем редактирования таблиц базы данных. Также, у пользователя имеется возможность редактировать классификатор статей расходов. Для того, чтобы сократить время, затрачиваемое сотрудниками на ведение учета, были предусмотрены функции автозаполнения некоторых полей. Например, при назначении студенту выплаты, сотруднику не обязательно вводить все личные данные студента вручную, достаточно лишь ввести первые символы фамилии, как программа предоставит возможные варианты. При выборе студента, все личные данные таблицы выплаты заполняются автоматически.

Также в системе предусмотрены проверки введенных данных. Это нужно для того, чтобы сократить число ошибок, которые может допустить пользователь. Например, пользователь не может добавить в систему студента, моложе 17 лет или назначить выплату учащемуся, который является отчисленным. При возникновении ошибки, программа подсветит поле с ошибкой и выдаст комментарий, который поможет ее устранить.

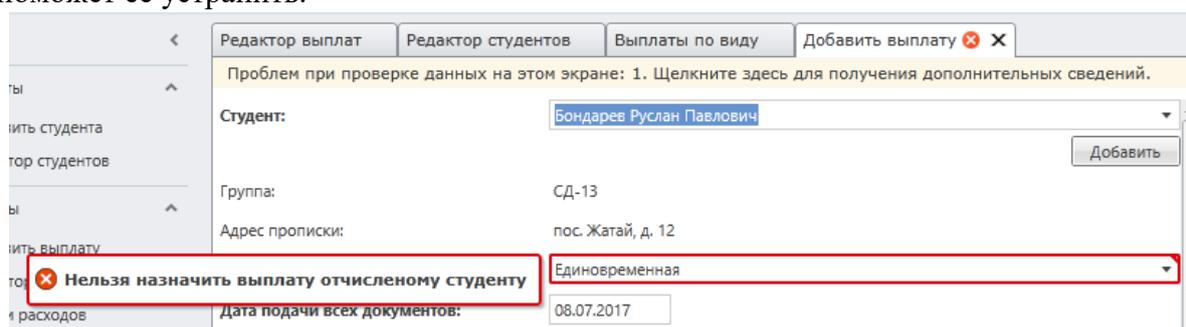


Рис. 2. Ошибка ввода данных

Для обеспечения удобной выборки, сортировки и поиска выплат, в системе были предусмотрены формы с выборкой выплат по студентам, по статье расходов и по виду выплаты. Для примера, чтобы отобразить все выплаты определенного студента за семестр, пользователю системы достаточно выбрать студента из списка и указать диапазон дат. Программа составит выборку и автоматически подсчитает сумму выплат данному студенту за семестр. Аналогичным образом можно получать выборки по статье и виду расходов.

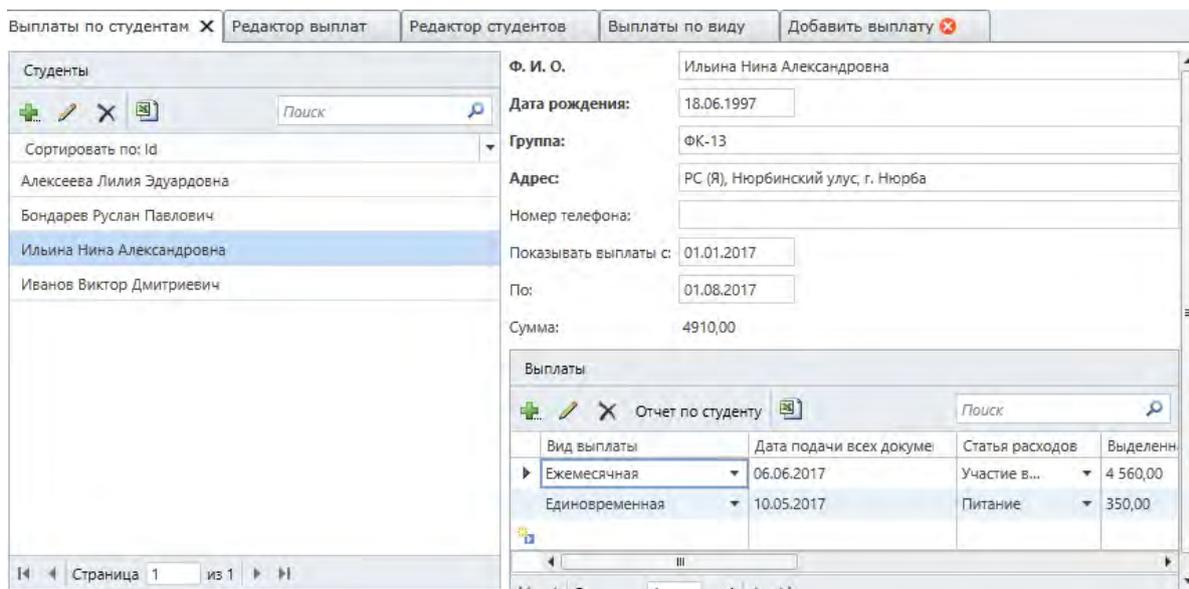


Рис. 3. Форма выборки выплат по студенту

Также в программе имеется возможность вывода данных выборок в виде типовой документации в виде документов Microsoft Word и Excel. Пользователю достаточно указать данные для выборки и нажать на кнопку экспорта отчета или таблицы Excel.

Подводя итоги, можно сказать, что данная система позволяет значительно сократить затраты временных ресурсов на ведение учета выплат студентам а также сокращает число ошибок, вызванных влиянием человеческого фактора. Что и было конечной целью внедрения данного продукта в отдел, и автоматизации в целом.

#### Список литературы:

1. Джейсон К. Visual C# .NET. Полное руководство. - М.: Корона Принт, 2004. - 960 с.
2. Бураков М.В., Латыпова Р.Р. Базы данных и язык SQL. –СПб.: ГУАП, 2014. – 120 с.

### **Проектирование структуры базы данных для автоматизированной информационной системы гостиницы**

*Родионов Д.Н., студент гр. ПИ-14, ТИ (ф) СВФУ, г. Нерюнгри*

*E-mail: flatron.dream@yandex.ru*

*Научный руководитель:*

*к.т.н., доцент кафедры МиИ Похорокова М.Ю.*

Важным этапом в проектировании базы данных является создание необходимых таблиц, содержащих наиболее полную информацию для работы в гостинице. Все данные должны храниться и обрабатываться в виде, понятном для любого пользователя. В связи с этим можно выделить следующие объекты предметной области (таблица 1):

Описание объектов предметной области

Название объекта	Описание
Регистрация	Забронированный номер
	Сотрудник, работающий в забронированном номере
	Клиент, забронировавший номер
	Дата заезда в забронированный номер
	Дата выезда из забронированного номера
Клиенты	Логин клиента для работы в программе
	Пароль клиента для работы в программе
	Фамилия, имя, отчество клиента
	Телефон, e-mail клиента для обратной связи
Номера	Номер
	Описание номера
	Статус номера
Сотрудники	Фамилия, имя, отчество сотрудника
	Адрес сотрудника
	Телефон сотрудника
	Зарботная плата сотрудника
	Должность сотрудника
Должность	Название должности сотрудника
Статус	Название статуса номера

В качестве системы управления базами данных (СУБД) можно выбрать одну из представленных программы: MySQL, PostgreSQL, SQLite, Oracle или Firebird.

MySQL является одной из самых популярных и распространенных СУБД, которая используется во многих компаниях. Эту систему управления базами данных определяют, как хорошую, быструю и гибкую систему, рекомендованную к применению в небольших или средних проектах. MySQL выполняет все команды быстро, обеспечивает работу неограниченного количества пользователей, а число строк в таблицах может быть равно 50 миллионам.

PostgreSQL – это свободно распространяемая система управления базами данных, которая не имеет ограничений ни по максимальному размеру базы данных, ни по максимуму записей или индексов в таблице. PostgreSQL поддерживает различные расширения и варианты языков программирования.

SQLite – это одна из самых компактных СУБД, является также встраиваемой и реляционной. SQLite позволяет хранить все данные в одном файле и, благодаря своему небольшому объему, отличается завидным быстродействием. Из недостатков можно отметить отсутствие системы пользователей и возможности увеличения производительности.

Oracle – эта СУБД относится к объектно-реляционному типу, отличающаяся стабильностью уже не один десяток лет, поэтому ее выбирают крупные корпорации, для которых важна надежность восстановления после сбоев, отлаженная процедура бэкапа, возможность масштабирования и другие ценные возможности. В отличие от других СУБД, стоимость покупки и использования Oracle достаточно высока.

Firebird – компактная, кроссплатформенная, свободная система управления базами данных (СУБД), работающая на Linux, Microsoft Windows и разнообразных

Unix платформах. В качестве преимуществ Firebird можно отметить многоверсионную архитектуру, обеспечивающую параллельную обработку оперативных и аналитических запросов, компактность, высокую эффективность и мощную языковую поддержку для хранимых процедур и триггеров.

В рамках данной работы была выбрана СУБД Firebird, так как она является полностью бесплатной и удовлетворяет всем требованиям к разрабатываемому приложению.

Далее была разработана физическая модель базы данных, которая определяет способы размещения данных в среде хранения и способы доступа к этим данным на физическом уровне (рис. 1).

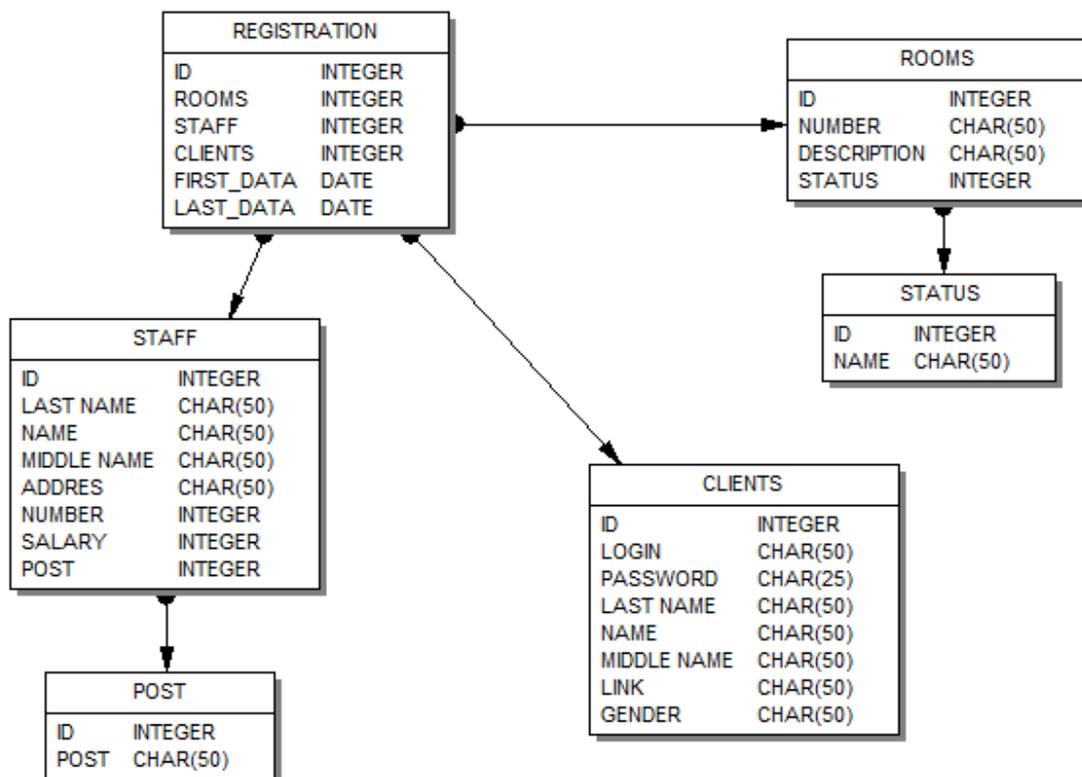


Рис. 1. Физическая структура базы данных

В соответствии с данной структурой записи в таблицах являются уникальными, т.к. имеют свой ID, что запрещает добавление уже существующих записей в таблицу. Спроектированная база данных является основой для дальнейшей разработки автоматизированной информационной системы для гостиницы.

## Оптимизация количества касс в розничной торговой точке с помощью имитационной модели магазина

*Рожкова А.Б., студентка гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри  
E-mail: alla.rozhkova.1995@mail.ru*

*Научный руководитель:  
старший преподаватель кафедры МиИ Юданова В.В.*

Вычисление приемлемого количества касс на торговом предприятии является важным фактором при желании максимизировать прибыль. Дабы не затрагивать систему в целом, на данный момент широко применяется метод имитационного моделирования, который позволяет воссоздать систему по всем заданным параметрам, максимально приближенную к реальной, в виртуальной среде. Данный метод так же позволяет менять отдельные элементы модели, дабы проследить динамику изменений и найти оптимальное решение для каждого случая, а также выполнить прогнозы действующей системы и оценить степень качества ее работы.

Для вычисления оптимального количества касс была построена имитационная модель торговой точки «Альянс» г. Нерюнгри. Данная модель содержит в себе: 2D и 3D модели, диаграмму процесса, а также основные статистические данные о работе системы.

Для определения оптимального количества касс на предприятии в продовольственном зале был создан оптимизационный эксперимент. Данный эксперимент основывается на количестве ресурсов, заданных как параметр. По умолчанию количество касс в зале равняется двум (рис. 1).

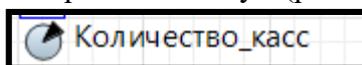
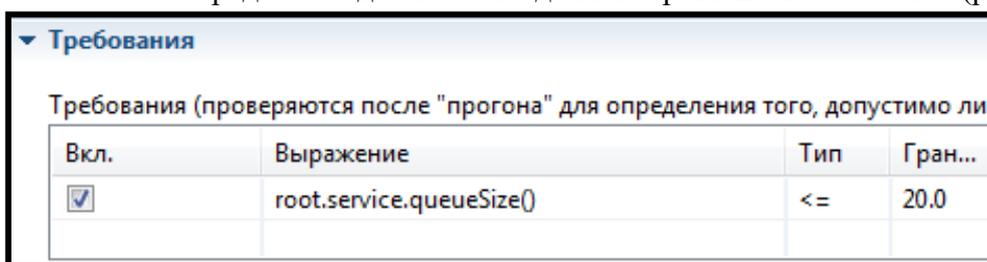


Рис. 1. Параметр, отвечающий за количество касс

Целью эксперимента является минимизация количества касс. основное условие, которое должно выполняться при выполнении итераций оптимизационного эксперимента - это очередь к каждой кассе не должна превышать 10 человек (рис. 2).



Вкл.	Выражение	Тип	Гран...
<input checked="" type="checkbox"/>	root.service.queueSize()	<=	20.0

Рис. 2. Свойства оптимизационного эксперимента

Количество итераций эксперимента равно 500. При поиске оптимального решения прогоны модели выполняются с варьированием параметра «Количество\_касс». Для полноты проводимого эксперимента параметр варьируется от единицы до пяти (рис. 3).

▼ Параметры				
Параметры:				
Параметр	Тип	Значение		
		Мин.	Макс.	Шаг
Количе...о_касс	дискретный	1	5	1
Интенс...ателей	фиксированный	1		

Рис. 3. Свойства оптимизационного эксперимента

Проведение оптимизационного эксперимента:

1) параметр «Интенсивность\_прихода\_покупателей» был задан как фиксированный параметр равный единице, «Количество\_касс» - варьируется в диапазоне от одного до пяти с шагом один (рис. 4):

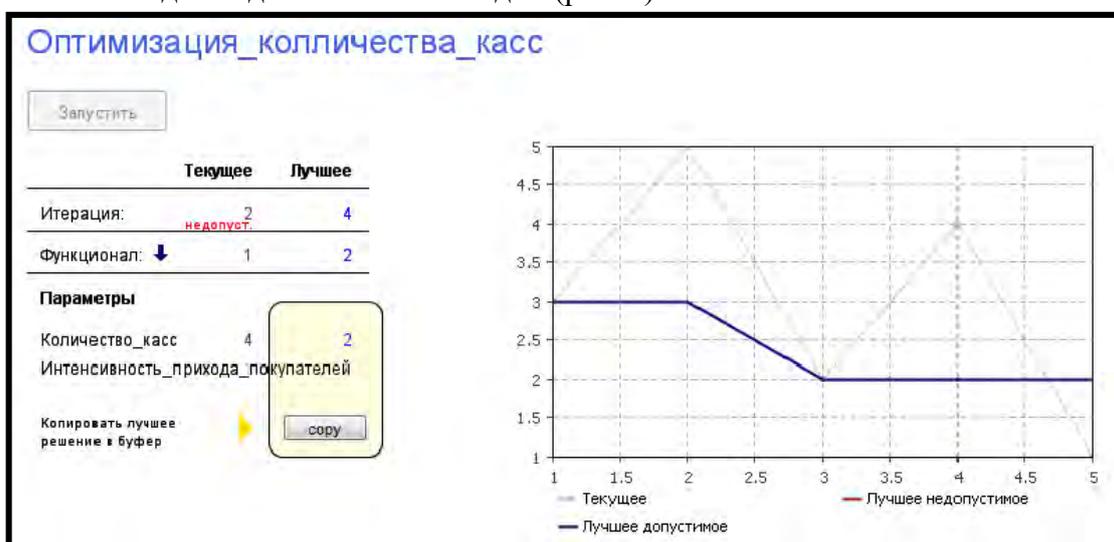


Рис. 4. Проведение оптимизационного эксперимента с фиксированным параметром «Интенсивность\_прихода\_покупателей» равным единице

2) параметр «Интенсивность\_прихода\_покупателей» был задан как фиксированный параметр равный двум, «Количество\_касс» - варьируется в диапазоне от одного до пяти с шагом один (рис. 5):

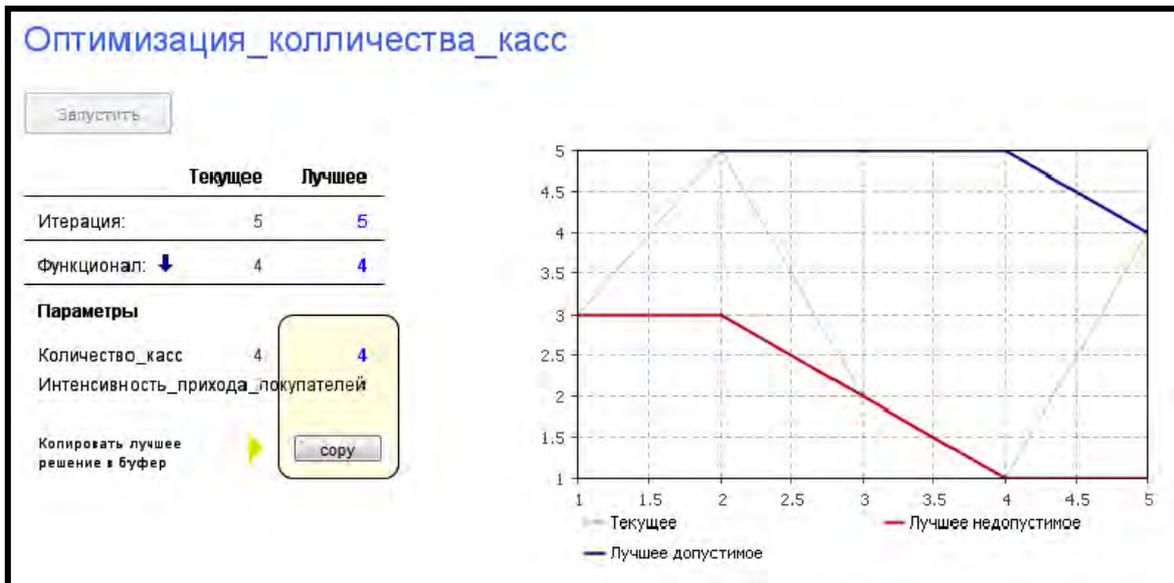


Рис. 5. Проведение оптимизационного эксперимента с фиксированным параметром «Интенсивность\_прихода\_покупателей» равным двум

3) параметр «Интенсивность\_прихода\_покупателей» был задан как фиксированный параметр равный трем, «Количество\_касс» - варьируется в диапазоне от одного до десяти с шагом один (рис. 6):

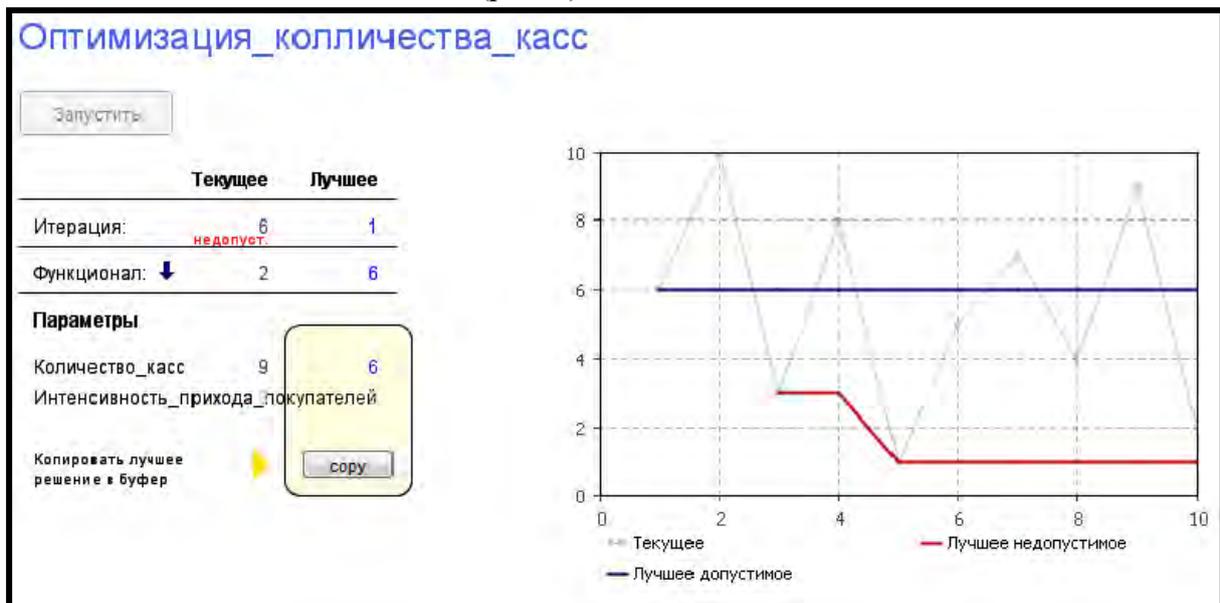


Рис. 6. Проведение оптимизационного эксперимента с фиксированным параметром «Интенсивность\_прихода\_покупателей» равным трем

Рассматривая результаты работы оптимизационного эксперимента, представленного на рисунке 4 можно сделать вывод, что при проведении пяти итераций было найдено лучшее решение, начинающееся с третьей итерации равное двум. При интенсивности прихода покупателей в минуту равной единице, оптимальное количество касс в продовольственном отделе равняется двум.

Для моделирования ситуаций, когда интенсивность прихода покупателей возрастает (праздники, выходные) были созданы оптимизационные эксперименты, представленные на рисунках 5 – 6.

Интенсивность прихода покупателей, как фиксированное значение равное двум представлена на рисунке 5, анализируя которой можно сделать вывод, что при проведении эксперимента оптимальное решение было найдено на пятой итерации и было равно четырем, недопустимыми же значениями при интенсивности прихода – от одного до трех. Таким образом можно сделать вывод, что при интенсивности прихода покупателей в минуту заданной как два, оптимальное количество касс, очередь к каждой из которых не должна превышать десяти человек, равно четырем.

При рассмотрении результатов работы с рисунка 6 было выявлено, что при интенсивности прихода покупателей в торговую точку заданной трем в минуту, наилучшее количество касс равно 6. Количество прогонов оптимизации десять. Недопустимые значения, согласно требованию по количеству очереди к каждой кассе, не превышающей десяти человек, от единицы до трех.

Таким образом можно сделать вывод о том, что наилучшее найденное решение по количеству касс в продовольственном отделе равняется двум, так как интенсивность прихода покупателей в минуту равная двум, что в наибольшей степени характеризует пропускную способность магазина в течении года. Увеличивать же количество касс для улучшения качества обслуживания в праздники не выгодно, в связи с тем, что большую часть времени одна из касс будет простаивать.

#### **Анализ программных средств автоматизации процессов торговли**

*Семашка Е.Э., студентка гр. ПИ-13, ТИ (ф) СВФУ, г. Нерюнгри*

*E-mail: mamrik1990@mail.ru*

*Научный руководитель:*

*старший преподаватель кафедры МиИ Юданова В.В.*

Развитие компьютерной техники, активная конкуренция, ужесточение законодательства (например, в торговле алкоголем или шубами из натурального меха, которые теперь невозможны без автоматизированных систем учета и оборудования для считывания 2D кодов или RFID-меток) — всё это тому, что всё большее количество торговых точек переходят к компьютеризации и автоматизации торговли. Для обеспечения нужд организаций создаются большие количества различного программного обеспечения, рассчитанные на различные формы торговли, имеющие различные степени автоматизации, начиная от достаточно дешевых программ, рассчитанные на минимальный учет товарного и денежного оборота, до дорогих систем, автоматизирующие буквально каждый элемент бизнес-процесса предприятия.

Поэтому рассмотрим различные варианты программного обеспечения, представленные на российском рынке. Интернет-сервис для управления торговлей «МойСклад».

Это не программа, а облачный сервис, разработанный для малого и среднего бизнеса. То есть для работы не требуется устанавливать программу на компьютер, а просто открыть сайт с любого устройства, имеющего доступ в Интернет.

В системе реализовано рабочее место продавца-кассира (рис. 1). Сюда включены: регистрация продаж, расчет сдачи, работа с возвратами, внесения и выплаты, открытие и закрытие смены, возможность наличной и безналичной оплаты, подключение торгового оборудования, печать кассовых и товарных чеков.

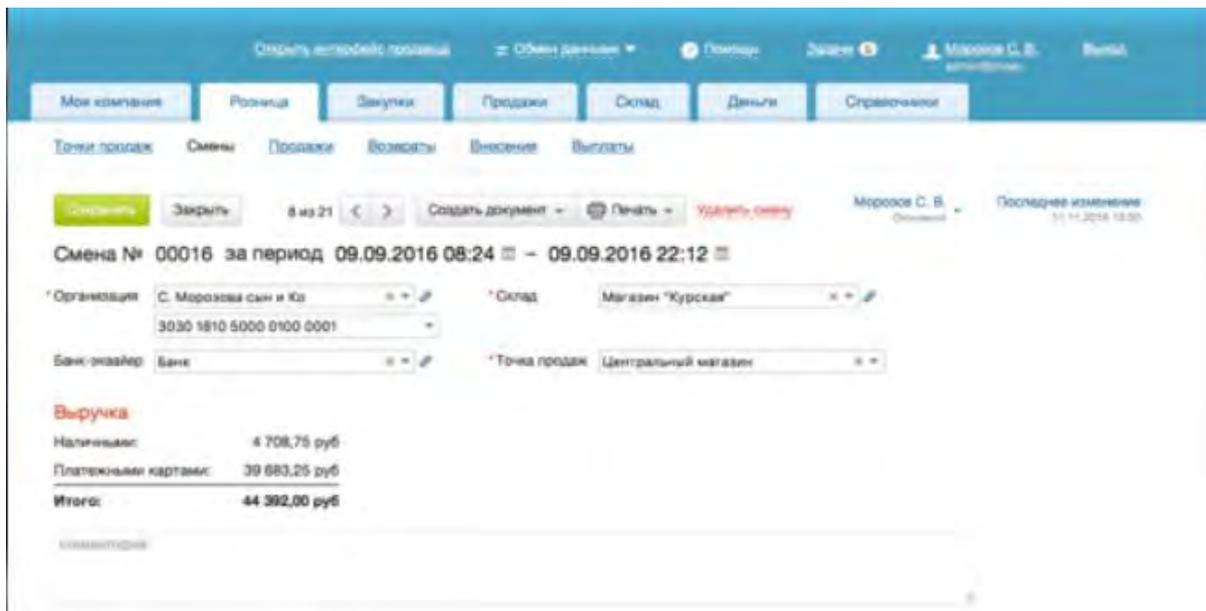


Рис. 1. Интерфейс рабочего места кассира в «МойСклад»

Товароучетная система. Включает возможность управления выбранным количеством торговых точек, контроль сдаваемой выручки, анализ продаж, работа с ассортиментом и ценами товаров.

Минусы данного решения:

- система зависима от доступа в интернет (оффлайн доступ реализован только для рабочего места кассира);
- возможность адаптации системы под свои требования ограничена только теми возможностями, которые реализовали разработчики;
- безопасность – данные хранятся на серверах разработчика, поэтому есть риск получения конфиденциальной информации сторонними лицами.

К семейству подобных систем можно также отнести - Система онлайн учета для бизнеса «Большая птица». Также бесплатная программа для магазина «CloudShop».

Среди нетиповых решений можно выделить наиболее известные: Штрих-М: Торговое предприятие 5 и Далион.

Минусами систем данного семейства являются:

- сложность внедрения на предприятия со сложной структурой;
- большая стоимость некоторых прикладных решений;
- требовательно к программному обеспечению, для больших предприятий требуется покупка дорогостоящего оборудования.

Во всех вышеописанных системах, как в онлайн-сервисах, так и полноценных программных решений, реализована возможность заказа поставщикам. Функционала их недостаточно, для решения поставленной нами цели, поэтому для разработки решения было выбрано создание собственного прикладного решения для автоматизированного формирования заказа.

## Python

*Чертов Д.А., студент 1 курса  
электроэнергетического факультета,  
Дальневосточный государственный аграрный университет, г. Благовещенск  
E-mail: maddclif@mail.ru  
Научный руководитель:  
к.т.н., доцент Коваль А.А.*

**Python** - это динамический язык общего назначения. С динамической строгой типизацией. Основные реализации: CPython, Jython, Cython, PyPy.

Гвидо ван Россум задумал Python в 1980-х годах, а приступил к его созданию в декабре 1989 года в центре математики и информатики в Нидерландах.

Язык Python был задуман как потомок языка программирования ABC (императивный, процедурный, структурный высокоуровневый язык программирования общего назначения и IDE), способный к обработке исключений и взаимодействию с операционной системой Амёба (открытая, микроядерная, распределённая операционная система). Ван Россум является основным автором Python и по сей день продолжает выполнять центральную роль в принятии решений относительно развития языка.

Версия Python 2.0 была выпущена 16 октября 2000 года и включала в себя много новых крупных функций — таких как полный Сборщик мусора (одна из форм автоматического управления памятью) и поддержка Unicode.

Однако наиболее важным из всех изменений было изменение самого процесса развития языка и переход на более прозрачный процесс его создания.

Первая обратно-несовместимая версия Python 3.0 была выпущена 3 декабря 2008 года после длительного периода тестирования. Многие её функции были импортированы в обратно совместимые Python 2.6 и Python 2.7.

### Философия Python

- Красивое лучше, чем уродливое.
- Явное лучше, чем неявное.
- Простое лучше, чем сложное.
- Сложное лучше, чем запутанное.
- Плоское лучше, чем вложенное.
- Разреженное лучше, чем плотное.
- Читаемость имеет значение.
- Особые случаи не настолько особые, чтобы нарушать правила.
- При этом практичность важнее безупречности.
- Ошибки никогда не должны замалчиваться.
- Если не замалчиваются явно.
- Встретив двусмысленность, отбрось искушение угадать.
- Должен существовать один — и, желательно, *только* один — очевидный способ сделать это.
- Хотя он поначалу может быть и не очевиден, если вы не голландец
- Сейчас лучше, чем никогда.
- Хотя никогда зачастую лучше, чем *прямо* сейчас.
- Если реализацию сложно объяснить — идея плоха.

- Если реализацию легко объяснить — идея, *возможно*, хороша.
- Пространства имён — отличная штука! Будем делать их побольше!

Язык очень популярен для Веба и нейронных сетей. Используется в MIT для обучения программированию курс по питону там заменил курс по схеме SCHME. Очень простой синтаксис.

Имеет большое количество библиотек.

### Возможности

- Синтаксическая возможность для аннотации параметров и результата функций (например, для передачи информации о типе или документирования).

- Полный переход на unicode для строк.

- Введение нового типа «неизменяемые байты» и типа «изменяемый буфер». Оба необходимы для представления бинарных данных.

- Новая подсистема ввода-вывода (модуль io), имеющая отдельные представления для бинарных и текстовых данных.

- Абстрактные классы, абстрактные методы (доступно уже в 2.6).

- Иерархия типов для чисел.

- Выражения для словарей и множеств  $\{k: v \text{ for } k, v \text{ in } a\_dict\}$  и  $\{e1, e2, e3\}$  (по аналогии со списковыми выражениями). Эта возможность была также добавлена в Python 2.7

- Изменения print из встроенного выражения во встроенную функцию. Это позволит модулям делать изменения, подстраиваясь под разное использование функции, а также упростит код. В Python 2.7 эта возможность активируется вводом `from __future__ import print_function`.

- Перемещение reduce (но не map или filter) из встроенного пространства в модуль functools (использование reduce существенно менее читабельно по сравнению с циклом).

- Удаление некоторых устаревших возможностей, поддерживаемых в ветке 2.x для совместимости, в частности: классы старого стиля, целочисленное деление с обрезанием результата как поведение по умолчанию, строковые исключения, неявный относительный импорт, оператор exes и т. п.

- Реорганизация стандартной библиотеки.

- Новый синтаксис для метаклассов.

- Изменён синтаксис присваиваний. Стало возможным, например, присваивание `a, *rest, b = range(5)`. С другой стороны, формальные параметры функций вроде `def foo(a, (b, c))` более недопустимы.

Питон прост. Например, поиск простых чисел:

```
def f(x):
    for y in xrange(2, x):
        if not x % y:
            return False
    return True
```

```
U = filter(f, xrange(2, 40))
```

- Будучи скриптовым языком, Питон медлителен. Для борьбы с этим используются: специальный транслятор для написания шустрых модулей к питону Cython, JIT-компилятор PyPy и прочее, их тысячи. Да и CPython на месте не стоит.

- Python, наряду с C++, Java и Go – один из языков, принятых к использованию в Google. В последнее время, впрочем, на нем ничего нового не пишут, а то, что есть,

постепенно переписывается на Java. А все из-за того, что Гвидо в свое время поссорился с руководством и ушел в Dropbox.

### **Применение**

Python — стабильный и распространённый язык. Он используется во многих проектах и в различных качествах: как основной язык программирования или для создания расширений и интеграции приложений. На Python реализовано большое количество проектов, также он активно используется для создания прототипов будущих программ. Python используется во многих крупных компаниях: Dropbox, Google (например некоторые части Youtube и Youtube API написаны на Python), Facebook Instagram.

Python с пакетами NumPy, SciPy и Matplotlib активно используется как универсальная среда для научных расчётов в качестве замены распространённым специализированным коммерческим пакетам Matlab, IDL и другим. Библиотека Astropy — популярный инструмент для астрономических расчётов.

В профессиональных программах трёхмерной графики, таких как Autodesk Maya, Blender, Houdini и Nuke, Python используется для расширения стандартных возможностей программ.

## Великий русский математик Николай Иванович Лобачевский

Николай Иванович Лобачевский родился 1 декабря (20 ноября) 1792 года в Нижнем Новгороде в бедной семье мелкого чиновника.

Девятилетним мальчиком он был привезен матерью в Казань и ее стараниями устроен вместе с двумя братьями в гимназию на казенное содержание. Юный 14-летний Лобачевский становится в феврале 1807 года студентом университета и вскоре проявляет особенную склонность к изучению физико-математических наук, обнаруживая выдающиеся способности.

Успехи студента Н.И.Лобачевского, соревнующегося в своих занятиях с И.П.Симоновым, впоследствии известным астрономом и участником кругосветного плавания, неизменно вызывали одобрение М.Ф.Бартельса и других профессоров.

3 августа 1811 г. Лобачевский утверждается магистром. Его руководитель профессор М.Ф.Бартельс был квалифицированным математиком и опытным преподавателем, но не вел творческой работы. Представив два научных исследования по механике и по алгебре («Теория эллиптического движения небесных тел» (1812 г.) и «О разрешимости алгебраического уравнения  $x^n - 1 = 0$ » (1813 г.), он был ранее срока в 1814 г. произведен в адъюнкт-профессоры (доценты).

Со следующего года он ведет самостоятельное преподавание, постепенно расширяя круг читаемых им курсов и уже задумываясь над перестройкой начал математики. Еще через год он получает звание экстраординарного профессора. Лобачевский Н.И. преподает математику на всех курсах, читает физические курсы и заведует физическим кабинетом, читает астрономию и геодезию, приняв в свое ведение обсерваторию, работает деканом физико-математического отделения. Колоссальный труд вкладывает он в упорядочивание библиотеки и в расширение ее физико-математической части. Наконец, несмотря на тысячи текущих дел и обязанностей, Лобачевский не прекращает напряженной творческой деятельности. Он пишет два учебника для гимназий: «Геометрию» (1823 г.) и «Алгебру» (1825 г.).

Его искания завершаются гениальным открытием: разрывая оковы тысячелетних традиций, Лобачевский приходит к созданию новой геометрии. 23 (11) февраля 1826 г. он делает на факультете доклад о новой «Воображаемой геометрии». Этот доклад «Сжатое изложение начал геометрии со строгим доказательством теоремы о параллельных» был передан на отзыв профессорам И.М. Симонову, А.Я. Купферу и адъюнкту Н.Д. Брашману. Лобачевский хотел знать мнение своих сотрудников об открытии, величие которого он сознавал, и просил принять свое сочинение в предполагаемое издание «Ученых Записок» отделения. Но отзыва не последовало. Рукопись доклада до нас не дошла. Материал этого доклада был включен Лобачевским в его первое сочинение «О началах геометрии», вышедшее в 1829 - 1830 гг. в



«Казанском вестнике».

Открытие Лобачевского было сделано им на путях принципиального критического пересмотра самых первых, начальных, геометрических понятий, принятых в геометрии еще со времен Евклида (3 век до н.э.). Это требование безусловной строгости и ясности в началах, это пристальное внимание к вопросам основ науки и углубленный анализ первоначальных понятий характерны вообще для творчества Лобачевского. Избранное им направление исследований способствовало тому, что он не только в геометрии, но и в ряде других областей математики превосходит достигнутый в то время уровень науки. Так, им дано уточнение понятия функции, приписанное впоследствии Дирихле; он четко разграничивает непрерывность функции и ее дифференцируемость; им проведены глубокие исследования по тригонометрическим рядам, опередившие его эпоху на много десятилетий; им разработан метод численного решения уравнений, несправедливо получивший впоследствии название метода Греффе, тогда как Лобачевский и независимо от него бельгийский математик Данделен разработали этот метод значительно раньше.

В 1827 г. великого геометра избирают ректором, и 19 лет он самоотверженно трудится на этом посту, добиваясь расцвета Казанского университета.



Лобачевский стремился претворить в жизнь свою широкую передовую программу университетского образования, представление о которой дает его речь «О важнейших предметах воспитания», произнесенная им через год после назначения ректором.

Лобачевский добивается существенного повышения уровня научно-учебной работы на всех факультетах. Он проводит строительство целого комплекса университетских вспомогательных зданий: библиотеки, астрономической и магнитной обсерватории, анатомического театра, физического кабинета и химической лаборатории. Он пытается создать при университете «Общество наук», но не получает на это разрешения. Журнал смешанного содержания «Казанский вестник» он заменяет организованным им строгим научным журналом «Учеными записками Казанского университета», первая книжка которого выходит в 1834 г. и открывается предисловием Лобачевского, освещающим цели научного издания. В течение 8 лет он продолжает

одновременно с ректорством управлять библиотекой. Он сам читает ряд специальных курсов для студентов. Он пишет наставление учителям математики и заботится о постановке преподавания также в училищах и гимназиях. Он принимает участие в поездке в Пензу в 1842 г. для наблюдения солнечного затмения. Умело оберегает он сотрудников и студентов университета во время эпидемии холеры в 1830 г., изолировав университетскую территорию и проводя тщательную дезинфекцию. Он организовал спасение астрономических инструментов и выноску книг из загоревшейся библиотеки во время громадного пожара Казани в 1842 г. Наконец, он организует чтение научно-популярных лекций для населения и открывает свободный доступ в библиотеку и музей университета.



И вместе с тем он находит время для непрерывных и обширных научных исследований, посвященных, главным образом, развитию новой геометрии. Его идеи были настолько непривычны, глубоки и новы, он настолько обогнал свою эпоху, что современники не смогли понять его и правильно оценить. Его первая работа «О началах геометрии» (1829 - 1830 гг.) была представлена Советом университета в 1832 г. в Академию наук. Но даже академик М.В. Остроградский не понял ее значения и дал на нее отрицательный отзыв.

Встретив непонимание и даже издевательство, Лобачевский не прекратил своих исследований. После работы 1829 - 1830 гг. «О начала геометрии» Лобачевский печатает в «Ученых записках»: в 1835 г. «Воображаемую геометрию» в 1836 г. «Применение воображаемой геометрии к некоторым интегралам». С 1835 по 1838 гг. он публикует свою наиболее обширную работу «Новые начала геометрии с полной теорией параллельных». Наконец, в 1840 г. выходят на немецком языке «Геометрические исследования по теории параллельных», где содержится предельно ясное и лаконичное изложение его основных идей. Эта мужественная борьба за научную истину резко отличает Лобачевского от других современников, приближавшихся тоже к открытию неевклидовой геометрии.



Ни одного положительного отклика не получает Лобачевский, кроме единственного высказывания профессора механики Казанского университета П.И. Котельникова, который в актовой речи в 1842 г. отметил, что изумительный труд Лобачевского, построение новой геометрии на предположении, что сумма углов

треугольника меньше двух прямых, рано или поздно найдет своих ценителей.

Многолетние плодотворные труды Лобачевского не могли получить положительной оценки у правительства Николая I. В 1846 г. Лобачевский оказался фактически отстраненным от работы в университете. Внешне он получил повышение - был назначен помощником попечителя (однако жалованья ему за эту работу не назначили), но при этом он лишился кафедры и ректорства.

Следует отметить, что менее чем за год до этого он был утвержден в шестой раз ректором университета на очередное четырехлетие. Вместе с тем более года он управлял Казанским учебным округом, заменив М.Н. Мусина-Пушкина, переведенного в Петербург. Указывая на эти свои служебные обязанности, Лобачевский незадолго до неожиданного предписания Министерства рекомендовал вместо себя на кафедру математики учителя Казанской гимназии А.Ф. Попова, защитившего докторскую диссертацию. Он считал необходимым поощрить молодого способного ученого и находил несправедливым занимать при таких обстоятельствах кафедру. Но, лишившись кафедры и ректорства и оказавшись в должности помощника попечителя, Лобачевский потерял возможность не только руководить университетом, но и вообще действительно участвовать в жизни университета.

Непонимание значения его новой геометрии, жестокая неблагодарность современников, материальные невзгоды, семейное несчастье и, наконец, слепота не сломили его мужественного духа. За год до смерти он закончил свой последний труд «Пангеометрия», диктуя его своим ученикам.

24 (12) февраля 1856 г. кончилась жизнь великого ученого, целиком отданная русской науке и Казанскому университету.

**ПРИКЛАДНЫЕ ИССЛЕДОВАНИЯ  
В ОБЛАСТИ МАТЕМАТИКИ И ИНФОРМАТИКИ**

Материалы региональной научно-практической конференции студентов и школьников  
по прикладной математике и информатике,  
посвящённой 225-летию со дня рождения Н.И. Лобачевского  
(25.03.2017 г.)

**Технический редактор *Л.В. Николаева***

Подписано в печать 30.11.2017. Формат 60x84/16.  
Бумага тип. №2. Гарнитура «Таймс». Печать офсетная.  
Печ. л. 3,7. Уч.-изд. л. 4,6. Тираж 10 экз. Заказ 387.  
Издательство ТИ (ф) СВФУ, 678960, г. Нерюнгри, ул. Кравченко, 16.

---

Отпечатано в ТИ (ф) ФГАОУ ВО «СВФУ»  
г. Нерюнгри