

Министерство образования Республики Беларусь

Учреждение образования  
«Гомельский государственный университет  
имени Франциска Скорины»

## **АКТУАЛЬНЫЕ ВОПРОСЫ ФИЗИКИ И ТЕХНИКИ**

**Материалы  
II Республиканской научной конференции  
студентов, магистрантов и аспирантов,  
посвященной памяти первого ректора  
Гомельского государственного университета,  
академика Владимира Алексеевича Белого**

(Гомель, 18 апреля 2013 года)

**В двух частях**

**Часть 2**

Гомель  
ГГУ им. Ф. Скорины  
2013

**«Актуальные вопросы физики и техники»**, II Республиканская научная конференция студентов, магистрантов и аспирантов, (2013, Гомель). II Республиканская научная конференция студентов, магистрантов и аспирантов «Актуальные вопросы физики и техники», 18 апреля 2013 г. : [материалы] : в 2 ч. Ч. 2 / редкол. : А. В. Рогачев (гл. ред.) [и др.]. – Гомель : ГГУ им. Ф. Скорины, 2013. – 212 с.

ISBN 978-985-439-790-0 (Ч. 2)

ISBN 978-985-439-788-7

В сборник вошли материалы докладов, представленные на II Республиканской научной конференции студентов, магистрантов и аспирантов «Актуальные вопросы физики и техники».

Адресуется научным сотрудникам, преподавателям, аспирантам, магистрантам, студентам.

#### **Редакционная коллегия:**

А. В. Рогачев (гл. ред.), О. М. Демиденко, И. В. Семченко,  
С. А. Хахомов, А. Н. Сердюков, Ю. В. Никитюк, А. Л. Самофалов,  
В. В. Андреев, Е. А. Дей, В. Д. Левчук, А. В. Воруев,  
В. Н. Мышковец, В. Е. Гайшун, Е. Б. Шершневу, Т. П. Желонкина

## **Секция 3 «Автоматизация исследований»**

### **Председатели:**

Левчук Виктор Дмитриевич, к.т.н., доцент,

Воруев Андрей Валерьевич, к.т.н., доцент

**О.А. Алексейчук (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Е.А. Левчук**, к.т.н., доцент

### **АВТОМАТИЗАЦИЯ УЧЕТА ЭЛЕКТРООБОРУДОВАНИЯ АЗС**

Работа посвящена разработке программы «Автоматизация учета электрооборудования АЗС». В работе была рассмотрена проблема автоматизации заполнения учетной карточки электрооборудования АЗС. При комплексном обслуживании АЗС инженером ведется учетная карточка для каждой АЗС, в которой указаны основные характеристики самой АЗС и оборудования имеющегося на ней. Учетная карточка заводится при открытии АЗС и с течением времени, в связи с модернизацией или ремонтом ее характеристик и оборудования меняется. При этом несколько раз в месяц необходимо предоставлять отчеты об АЗС, его оборудовании, комплектации, потребляемых мощностях, информация для которых берется из учетной карточки. Вид отчета зависит от требований других обособленных подразделений РУП ПО «Белоруснефть» и каждый раз может быть разным.

Актуальность работы обусловлена ростом количества АЗС и оборудования и необходимостью получения актуальной информации в любое время. Целью данной работы является автоматизация рабочего места инженера управления «Промсервис» и уменьшение рутинного труда по заполнению и редактированию учетной карточки.

Программа «Автоматизация учета электрооборудования АЗС» представляет собой информационно – поисковую систему, основной задачей которой является составление учетной карточки электрооборудования, её редактирование, вывод на экран и печать, перемещение оборудования из одной АЗС на другую, поиск данных по необходимым критериям и составления отчета об найденной информации, посылка учетной карточки электрооборудования по e-mail.

Приложение предназначено для работы с базой данных, таблицы которой и связи между ними спроектированы, следуя формализованным правилам, обеспечивая простоту манипулирования данными. В результате приведения данных к третьей нормальной форме получены 10 таблиц, связанных друг с другом связью один ко многим.

Для написания программы «Автоматизация учета электрооборудования АЗС» используется среда разработки Microsoft Visual FoxPro 9.0. Указанная среда очень удобна для разработки программ такого рода, она максимальным образом ориентирована на создание и управление базами данных и, как следствие, на разработку информационно-поисковых систем, ориентированных на обработку больших массивов данных.

Для вывода учетной карточки и отчетов по поиску используется Microsoft Excel. Структура учетной карточки сделана в соответствии с требованиями, предъявляемыми управлением «Промсервис», а структура вывода отчетов по поиску выбрана таким образом, что бы ее легко можно было редактировать.

Структура пользовательского интерфейса имеет однотипный вид, интерфейс которых аналогичен продуктам компании Microsoft.

После запуска программы появляется окно со следующими пунктами меню:

- Ввод и редактирование учетной карточки.
- Перемещение оборудования.

– Поиск, в который входят такие пункты меню, как по АЗС (вызывает диалоговое меню поиска информации о АЗС и ее характеристиках) и по другим критериям (вызывает диалоговое меню поиска информации о колонках, электротехнологических установках, воздушных и кабельных линиях, высоковольтных распределительных устройствах, высоковольтных двигателях, электродвигателях и другом электронагревательном оборудовании, компенсирующих устройствах, автономных источниках электроснабжения).

- Справка.

Программа «Автоматизация учета электрооборудования АЗС» может использоваться в управлениях «Промсервис» любого города. Использование данной программы уменьшило труд по поиску, корректировке и печати и по отправке информации. Таким образом, данный программный продукт стал незаменимым помощником для инженеров и инспекторов, которые заполняют или пользуют учетную карточку.

**М.И. Батанов (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **П.Л. Чечет**, к.т.н., доцент

## **ИНТЕГРАЦИЯ ПРИЛОЖЕНИЙ В СОЦИАЛЬНЫЕ СЕТИ**

Интеграция приложений является наиболее важным процессом при размещении приложения в социальной сети. Смысл интеграции сводится к внедрению разнообразных функций социальной сети в само приложение.

Например, сюда относятся получение данных о пользователе, списка его друзей, размещение разнообразной информации на своей стене или друга и так далее. В дальнейшем такая интеграция позволяет игрокам плотно взаимодействовать друг с другом внутри приложения, отправлять сообщения и совершать другие действия. Чем больше функций социальные сети дублирует приложение, тем удобнее будет им пользоваться обычным посетителям. А так же более глубокая интеграция увеличивает монетизацию приложения, что очень важно для любого разработчика.

Каждая социальная сеть, которая предоставляет возможность размещения приложений, имеет свое уникальное API (от англ. Application Programming Interfaces – интерфейсы программирования приложений). API определяет функциональность, при этом позволяя абстрагироваться от того, как именно эта функциональность реализована [2].

Как правило, API в социальных сетях разделяется на два части:

- клиентская;
- серверная.

Полноценная интеграция приложений включает в себя реализацию не только клиентского API сети, но еще и серверной ее части. Данная часть отвечает чаще всего за действия, которые должны быть защищены от взлома со стороны пользователей. Сюда можно отнести операции с биллингом, подтверждение списания сумм со счета пользователя в сети, отправка уведомлений о событиях, произошедших в отсутствие пользователя и так далее.

В данной статье рассмотрена клиентская интеграция IFrame–приложения с социальной сетью ВК («ВКОНТАКТЕ»).

Для начала необходимо зарегистрировать приложение в ВК, где необходимо указать основные данные и адрес приложения. ВК использует указанный адрес для отображения содержимого у себя в IFrame.

Для интеграции клиентского ВК API в приложении необходимо добавить строку подключения JavaScript файла в заголовке HTML [1]:

```
<script src=http://vk.com/js/api/xd\_connection.js?2 type="text/javascript"> </script>
```

После загрузки страницы для инициализации приложения можно использовать следующий код [1]:

```
VK.init(function() {  
  //Функция обратного вызова  
});
```

Набор функций клиентского API социальных сетей включает в себя большой набор возможностей. К примеру, в клиентском ВК API представлены следующие функции:

- *callMethod*. Осуществляет внешний вызов метода JavaScript API.
- *addCallback*. Добавляет функцию в качестве обработчика события.
- *api*. Выполняет запрос к ВК API и передаёт полученные данные в функцию обратного вызова.

С помощью этих функций клиент может вызвать окно для приглашения друзей, окно системы биллинга или получить из сети данные по текущему пользователю и т. д.

Следующий пример кода демонстрирует внешний вызов изменения размеров IFrame [1]:

```
VK.callMethod("resizeWindow", 510, 600);
```

В статье были рассмотрены основные принципы интеграции приложений с социальными сетями. На примере социальной сети «ВКонтакте» были представлены основные этапы интеграции IFrame–приложения.

### Литература

1. Дуров, П. ВКонтakte [Электронный ресурс] / П. Дуров. // Социальная сеть. – 2013. – URL: <http://vk.com/> – Дата доступа: 13.03.2013.
2. Уэйлс, Д. Свободная энциклопедия Википедия [Электронный ресурс] / Д. Уэйлс. // Электронная библиотека. – 2003. – URL: <http://ru.wikipedia.org/> – Дата доступа: 15.03.2013.

**Е.Е. Бердников (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **В.А. Дробышевский**, старший преподаватель

### МОДЕРНИЗАЦИЯ ЛВС ПРЕДПРИЯТИЯ ЧСУП «ВТИ»

Современные условия организации труда требуют постоянного улучшения технологического процесса. Основная причина потребности модернизации локальной вычислительной сети в том, что не были учтены темпы роста информационных потребностей заказчика, поэтому нагрузка на существующую сеть увеличивается, а производительность падает.

Это приводит к необходимости модернизации локальной вычислительной сети на предприятии ЧСУП «ВТИ». Существующая локальная вычислительная сеть на предприятии со скоростью передачи данных 10 Мб/сек (Ethernet) не соответствует необходимой скорости для работы предприятия ЧСУП «ВТИ», что привело к необходимости модернизации

локальной вычислительной сети до скорости 100 Мб/сек (Fast Ethernet). Это обусловлено не только развитием новых технологий и увеличением объёмов передачи данных, но и необходимостью решать бизнес задачи стоящие перед компаниями в данный период развития. Модернизацию необходимо проводить в свете уже существующих ИТ-решений в перспективе горизонтов развития бизнеса.

Побочными причинами модернизации ЛВС могут быть нарушение безопасности данных и трудности в управлении. С переходом на более скоростные технологии возникает потребность внесения больших изменений. Упор делается на увеличение производительности сети, для чего нужно устанавливать новые адаптеры и драйверы для них на каждый компьютер.

Модернизация ЛВС затронула и скоростные магистрали. Они должны отвечать высоким требованиям – должна быть возможность подключения сетевых серверов, рабочих станций. Для работы таких устройств, требуется высокая скорость для передачи любых объемов информации и малые задержки, чтобы была возможность перейти на другую технологию передачи данных. Это решение имеет эффект для крупных сетей, но требует высококвалифицированной подготовки персонала. Реструктуризация и модернизация ЛВС позволяет обеспечить повышение производительности сети.

**С.С. Бокатюк (УО БрГТУ, Брест)**

Науч. рук. **В.И. Хвещук**, к.т.н., доцент

### **ИНТЕРНЕТ-МАГАЗИН ДЛЯ ООО «ФОТОПЕРЕХОДНИК»**

В работе представлены результаты создания интернет-магазина (ИМ) по организации продаж запасных частей к фотоаппаратам разных марок и производителей осуществляющей ООО «Фотопереходник». Запасные части хранятся на складе организации. Поставку запасных частей на склад и отправку оплаченных заказов на запасные части реализует предприятие. Пользователями ИМ являются сотрудники организации (руководство, отдел продаж, отдел технической поддержки) и интернет-покупатели.

ИМ обеспечивает автоматизацию следующих функций: для руководства – формирование ассортимента товаров, ведение финансовой отчетности магазина, контроль финансовых операций; для отдела продаж – оформление, ведение и контроль заказов, проверка оплаты заказов, ведение журнал заказов; для системного администратора – контроль за работой ИМ, формирование отчетов о работе ИМ.

ИМ реализован на основе клиент-серверной архитектуры и представлен в виде трех автономных подсистем, взаимодействующих между собой через общую базу данных (БД).

каждая из которых ориентирована на автоматизацию определенного класса задач в рамках деятельности предприятия, а именно:

1 Подсистема покупателей – поиск товаров, просмотр описания товаров, заказ товаров с выбором способов оплаты и доставки, личный кабинет с возможностью отслеживания статуса сделанного заказа, рассылка информационных писем по электронной почте.

2 Подсистема менеджера – ведение журнала заказов с возможностью обновлением деталей и текущего статуса заказа (ждет оплаты, ждет отправки, отправлен).

3 Подсистема руководителя – ведение учета товаров, формирование финансовой отчетности за период, администрирование.

Взаимодействие пользователей с ИМ осуществляется в диалоговом режиме (круглосуточно) и обеспечивается санкционированный доступ пользователей к ресурсам базы данных и к ее подсистемам. Созданы средства архивирования и восстановления ИМ.

Для функционирования ИМ необходимы следующие технические средства: сервер БД, сервер приложения, ПЭВМ менеджера, ПЭВМ руководителя.

Программное обеспечение (ПО) ИМ реализовано в рамках СУБД Microsoft SQL Server 2008 R2, WCF сервисов для взаимодействия с таблицами БД и обработки информации, HTML и Javascript для реализации диалогового пользовательского интерфейса. ПО представляет собой веб-приложение с разграниченным доступом. Доступ к каждой из частей предоставляется после авторизации на основе роли пользователя.

Для ИМ разработан комплект проектных и эксплуатационных документов, необходимых для эксплуатации и сопровождения ИМ.

В настоящее время ИМ находится в стадии опытной эксплуатации.

**М.Ю. Бокий (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Н. Кулинченко**, старший преподаватель

## **РАЗРАБОТКА ПОДСИСТЕМЫ ИНТЕГРАЦИИ ДАННЫХ В ПРОЕКТЕ КОНСОЛИДАЦИИ ФИНАНСОВЫХ ОТЧЕТОВ**

На современном уровне развития средств информатизации организации уже не задают себе вопросы типа: «Надо ли нам создавать хранилище



данных?». Вместо этого они спрашивают: «Можем ли мы консолидировать все наши данные и превратить их в надежный, упорядоченный информационный источник?».

В связи с тем, что большое количество организаций внедрило у себя приложения для управления электронной коммерцией (e-business), управления взаимодействием с заказчиками (CRM), управления предприятием, а также приложения для поддержки принятия решений, процесс сбора, очистки, трансформации и подготовки бизнес-данных для анализа перерастает в серьезную проблему интеграции. От выбора способа интеграции бизнес-данных зависят затраты времени, средств и ресурсов компании, необходимых для расширения и поддержки информационного окружения.

В данном проекте для интеграции данных наилучшим выбором будет использование IBM InfoSphere DataStage. Этот продукт позволяет интегрировать большие объемы данных между многочисленными источниками данных и целевыми приложениями.

Архитектура системы на основе DataStage (рисунок 1):

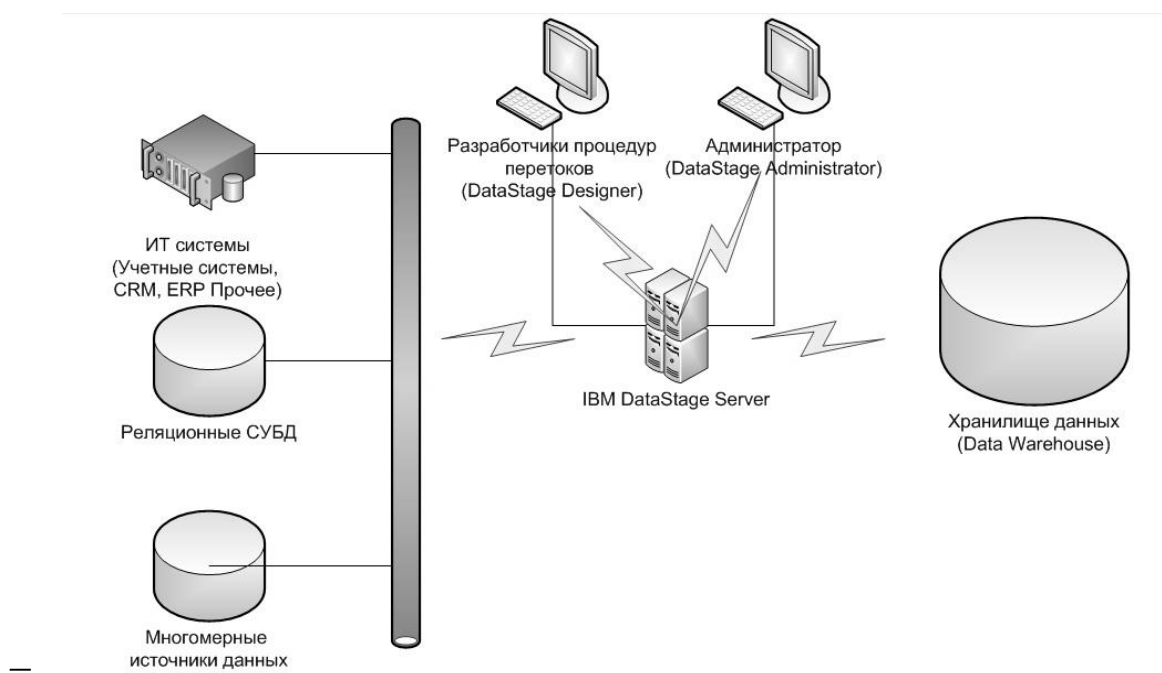


Рисунок 1 – Архитектура системы

Данные из множества внешних источников поступают на InfoSphere DataStage Server, проходят необходимые преобразования и загружаются в витрину данных, использующую схему “Звезда”.

Обработка данных происходит параллельно на нескольких процессорах. Суррогатные ключи генерируются автоматически на основе натуральных.

Бизнес-аналитики, как основные пользователи корпоративного хранилища данных, нуждаются в отслеживании изменений значений атрибутов аналитических измерений. Отслеживание изменений значений аналитических измерений в хранилище данных решается путем применения механизма медленно меняющихся измерений (Slowly Changing Dimensions, SCD).

Данные в таблицах измерений обновляются по принципу SCD type 2, т. е. с сохранением истории, предыдущих версий. Например, при удалении записи в источнике, соответствующая запись в целевой таблице изменений не удалится, а пометится соответствующими идентификаторами как удаленная и не используемая. Выполнение update в этом случае разделяется на 2 составляющие: обновление старой записи с пометкой ее как обновленной и не используемой, и добавление новой записи с тем же ключом и увеличенным на единицу ключом версии, которая будет новой текущей (используемой) записью для этого ключа.

Последовательность действий при получении запроса от пользователя выглядит следующим образом (рисунок 2):

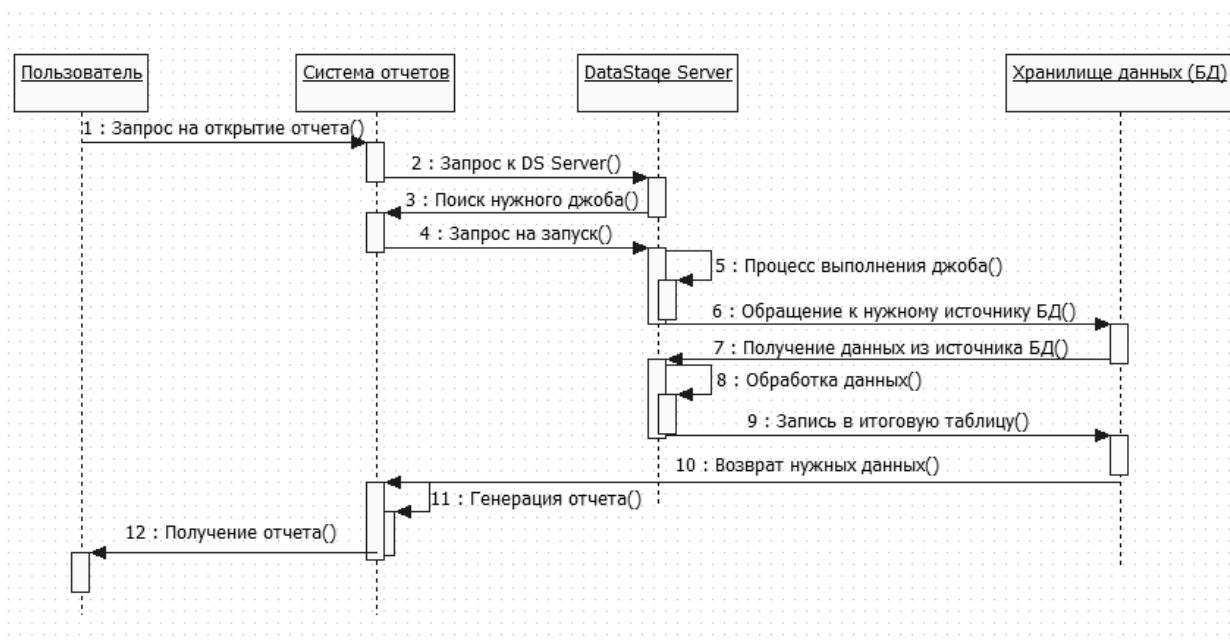


Рисунок 2 – Диаграмма последовательности

Подсистема автоматически интегрирует данные из всех источников в целевую витрину данных Dimensional Data Mart. Она обновляет данные в витрине с сохранением истории. Затем на основе этих данных строятся различные отчеты для пользователей.

Диаграмма вариантов использования продукта (рисунок 3):

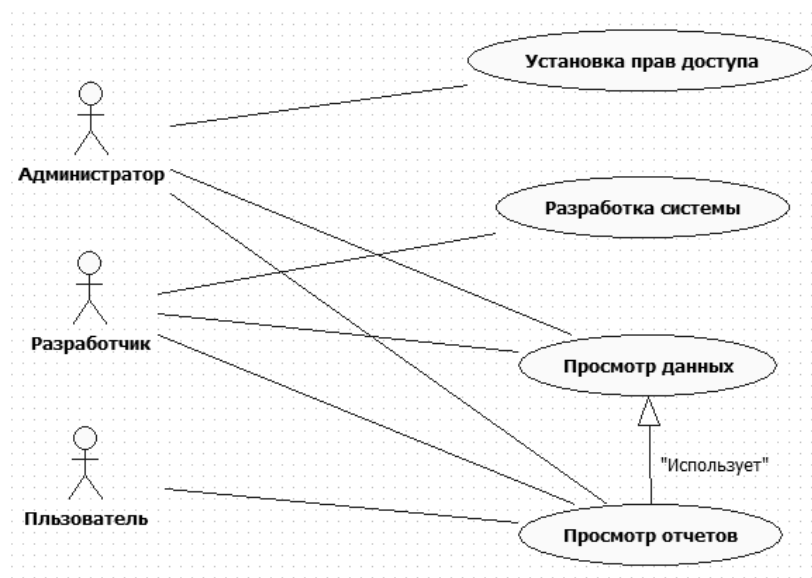


Рисунок 3 – Диаграмма вариантов использования

Загрузку и обновление данных можно инициировать вручную запуском главной последовательности, автоматически при получении запроса от пользователя или производить по расписанию, например, ежедневно.

**И.А. Борихин (УО БрГТУ, Брест)**

Науч. рук. **В.И. Хвещук**, к.т.н., доцент

## СРЕДСТВА УЧЕТА ДИПЛОМНЫХ ПРОЕКТОВ

В работе представлены результаты разработки комплекса средств автоматизации (КСА) деятельности выпускающей кафедры по организации и контролю дипломного проектирования. Для автоматизации были определены следующие основные задачи: формирование тем дипломных проектов (ДП) и распределение их между студентами; формирование приказа об утверждении тем ДП, списка членов государственной комиссии и списка рецензентов; закрепление дипломных проектов за рецензентами и выдача направлений на рецензирование; контроль процесса дипломного проектирования и другие функции. Определен набор документов доступных для просмотра (приказ о закреплении тем ДП, график консультаций, график защит, ход выполнения ДП, график предварительной защиты) и перечень формируемых документов (приказ о закреплении тем ДП, направления на рецензирование) в рамках КСА.

В основу создания КСА положена клиент-серверной архитектура. Комплекс реализуется в виде совокупности диалоговых приложений

взаимодействующих через общую базу данных. Для каждой из групп пользователей КСА обеспечивает следующие функции:

1 Для студентов – предварительное формирование тем ДП, просмотр различных документов в БД.

2 Для преподавателей – формирование тем ДП, регистрация хода дипломного проектирования, допуск к нормоконтролю, просмотр различных документов БД.

3 Для преподавателей (нормоконтролеров) – регистрация результатов нормоконтроля, просмотр различных документов БД.

4 Для преподавателей (членов комиссии по предварительной защите ДП) – регистрация результатов предварительной защиты ДП.

5 Для секретаря кафедры – формирование и документирование графиков и приказов, организацию санкционированного доступа пользователей к ресурсам КСА.

6 Для заведующего кафедрой – утверждение тем ДП, закрепление тем ДП за студентами и руководителями ДП, допуск к рецензированию и другие функции.

КСА создается в рамках локальной вычислительной сети кафедры «Интеллектуальные информационные технологии» БрГТУ.

**Д.О. Бугримов (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Д. Левчук**, к.т.н., доцент

## **РАЗРАБОТКА КОМПОНЕНТА ПО ЗАГРУЗКЕ РЕСУРСОВ В DAM ДЛЯ КОРПОРАТИВНОГО САЙТА WWW.ERAM.COM**

Для загрузки цифровых ресурсов в DAM был разработан компонент Upload Action. Его отличие от стандартного Assets Manager заключается в удобстве и компактности. Нет необходимости иметь доступ в административную панель для того, чтобы загрузить какие-либо данные. Достаточно добавить данный компонент на страницу и произвести необходимые действия для загрузки документов. Ограничения на формат и размер файлов не задается.

При работе с приложением пользовательские данные сначала обрабатываются с помощью JavaScript кода, а не отправляются прямо на сервер. Пока это происходит, страница не блокируется, пользователь даже не замечает, что происходит запрос на сервер. Другими словами, код JavaScript передает запрос в фоновом режиме.

Для рендеринга компонента в меню Actions используется следующий код:

```

<% @page contentType="text/html; charset=UTF-8" trimDirectiveWhitespaces="true"%>
<% @include file="/apps/epam/global.jsp"%>
<epm:uploadAction var="result"/>
<c:choose>
  <c:when test="\${not empty param.nodePath}">
    <c:out value="\${result}"/>
  </c:when>
  <c:otherwise>
<a id="uploadAction" href="javascript:void(0);">Upload</a>
</c:otherwise>
</c:choose>

```

В начале jsp-страницы происходит подключение файла global.jsp, который содержит в себе инициализацию внутренних переменных и объектов. Далее происходит вызов тэга uploadAction, результат выполнения которого заносится в переменную result. После этого, с использованием технологии jstl, осуществляется управление потоком выполнения.

Происходит проверка значения параметра nodePath, в случае, если он не пустой происходит вывод содержимого переменной result. Это сделано для обработки AJAX запросов. В противном случае происходит рендеринг ссылки на окно загрузки.

Внешний вид компонента представляет собой всплывающий диалог, созданный с помощью библиотеки Ext JS. Сама реализация загрузки файла лежит на пользовательском тэге uploadAction.

Весь JavaScript код вынесен в отдельный файл, чтобы не засорять содержимое JSP файла. Для того чтобы js код отработывал в нужное время, используется технология слушателей. Таким образом, происходит добавление слушателя на событие onclick для элемента, имеющего идентификатор uploadAction

Вначале создается панель, на которой будут размещены все элементы. Затем происходит их описание. Через свойство xtype задается тип элемента, в нашем случае это fileupload. Затем задается текст для метки через свойство fieldLabel. Использование fileNameParameter необходимо для задания имени параметра, под которым данные будут приходить на сервер. Свойство allowBlank выставлено в false для того, чтобы это поле было обязательно заполнено. Для того, чтобы запретить ввод данных вручную необходимо выставить true для свойства readOnly.

Для того чтобы запретить отправку незаполненной формы, необходимо задать валидацию.

Также добавляется обработчик на событие beforeaction, который решает, может ли данная форма быть отосланной на сервер. Для этого

отправляется AJAX запрос на сервер, в котором передается имя файла и путь. В случае если файл существует, будет получена единица. Для отсылки данных используется функция `getFile`, которая отсылается данные в синхронном режиме. Если файл существует, будет выведено соответствующее сообщение и отправка данных будет заблокирована до тех пор, пока пользователь не выберет не существующий файл. Данные отправляются в JSON формате.

JSON – как и XML является текстовым форматом обмена данными, при этом намного моложе последнего. Является более прогрессивным методом, благодаря большей гибкости своей архитектуры по сравнению с XML. В зависимости от ответа будут вызваны следующие функции:

```
switch (action.failureType) {
  case CQ.Ext.form.Action.CONNECT_FAILURE:
    CQ.Ext.Msg.alert("Failure", "Ajax communication failed");
  default:
    CQ.Ext.Msg.alert("Failure", "Access denied");
}
```

В случае успешной обработки пользовательского запроса, страница будет обновлена и пользовательский файл станет доступен. В случае неудачи будет сгенерировано сообщение с описанием причины.

Для загрузки файла необходимо нажать на кнопку `Browse` и выбрать необходимый документ. Поле `Path` служит для задания пути внутри репозитория, куда будет загружен файл. Также в появившемся окне реализована возможность для создания папок.

Через технологию AJAX осуществляется проверка на существование папки. Если такая папка уже существует, об этом будет выведено соответствующее текстовое сообщение.

После того, как папка была создана, необходимо обновить окно с выбором путей, чтобы в нем появилась недавно созданная нами папка. Для этого необходимо повесить слушателя на событие `afterrender`.

```
var empty = true;
node.eachChild(
function(iNode) {
  if ("type" in iNode.attributes && iNode.attributes.type == "sling:OrderedFolder") {
    iNode.attributes.cls = "folder";
    empty = false;
  } else {
    iNode.hidden = true;
```

В данном фрагменте добавляется слушатель на событие load, при активации которого на вновь созданную папку добавляется css-класс, меняющий его внешний вид.

Весь функционал по непосредственной загрузке файла на сервер находится в тэги uploadAction. В нем происходит получение сессии, которая необходима для проведения различных манипуляций с репозиторием. Здесь она используется для того, чтобы определить, существует ли узел с переданным именем. Если такой узел есть, происходит передача единицы на вызывающую jsp, в противном случае будет возвращена минус единица. Исходя из ответа можно определить существование узла, отправив AJAX запрос с именем.

Таким образом, для создания данного компонента использовались такие технологии как: java, jstl, jsp, JavaScript, Servlet API, Ext JS, AJAX.

**Д.О. Бугримов (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Д. Левчук**, к.т.н., доцент

## **РАЗРАБОТКА КОМПОНЕНТОВ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ КОРПОРАТИВНОГО САЙТА WWW.EPAM.COM**

Разработка официального сайта EPAM осуществляется с применением самых передовых технологий, которыми является платформа Adobe CQ. Она использует Java Content Repository для хранения и управления контентом, поддерживает как скрипты, так и Java servlets для обработки HTTP запросов в RESTful стиле. Встроенный Apache Felix OSGi фреймворк и консоль позволяют все настраивать в реальном времени.

Основная цель сайта epam.com – это привлечение новых людей. Для ее достижения был написан ряд компонентов, относящихся к вакансиям. Один из них – компонент Send Resume. Компонент представляет собой изображение рекрутера, его имя и фамилию, а также кнопку отправить резюме. Он может настраиваться для отображения текста под кнопкой.

Для хранения информации о рекрутерах были созданы специальные узлы, скрытые от пользователей. Информацию, хранящуюся в них, можно получить только с использованием API. Для получения узлов случайным образом используется тэг getRandomChildNode, который был специально написан для данного компонента. Всю остальную часть работы делает фреймворк и разработчику остается написать только HTML, JS и CSS.

На сайте [www.epam.com](http://www.epam.com) должно быть отражено сотрудничество с университетами и лабораториями. Это позволит будущим студентам лучше ориентироваться в выборе места учебы. Для этого был написан компонент *Laboratories*. Этот компонент является составным.

На белорусской версии сайта [www.epam.com](http://www.epam.com) должно быть отражены основные категории, по которым идет набор сотрудников. Это позволит будущим работникам быстро сориентироваться на сайте и не тратить время на поиск нужной вакансии. Также должен присутствовать список вакансий по городам. Для того чтобы интерфейс был удобным и не загромождал сайт лишним контентом, данный список был вынесен во всплывающее меню.

Кастомизация работы компонентов осуществляется через диалоги. Диалог представляет собой окно, которое появляется при двойном щелчке по компоненту и предоставляет возможность для внесения различных корректировок. Все диалоги создаются с использованием ExtJS. Так, для компонента *Vacancy Cover* все данные вводятся через встроенный Rich Text редактор, который доступен через диалог компонента.

Данные компоненты внедрены на корпоративный сайт.

**Д.В. Будько (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Е.А. Левчук**, к.т.н., доцент

## **ПРИМЕНЕНИЕ ФРЕЙМВОРКА HIBERNATE ДЛЯ РАЗРАБОТКИ ВЕБ-ПРИЛОЖЕНИЯ**

Современные клиент-серверные приложения, так или иначе, взаимодействуют с базами данных при помощи различных средств. *Hibernate* – библиотека для языка программирования Java, предназначенная для решения задач объектно-реляционного отображения.

Было разработано приложение, позволяющее делать заказы сотрудникам ИТ-компаний. Основными фреймворками, которые были использованы для написания данного приложения, являются *Spring* и *Hibernate*.

*Hibernate* предоставляет удобный механизм, который дает возможность разработчику работать с таблицами базы данных как с объектами языка программирования. Для того, чтобы добиться этого результата необходимо провести предварительную настройку фреймворка. В первую очередь необходимо описать фильтр *Hibernate*. Данный фильтр описывается в дескрипторе развёртывания веб-приложения, в нашем случае используется класс *OpenSessionInViewFilter*.



Для работы с базой данных приложению необходимо иметь подключение к ней. Есть несколько вариантов решения этой задачи, но в ходе реализации проекта было принято решение использовать `dataSource`. Использование `dataSource` позволит нам снизить время затрачиваемое для установления подключения к базе. Также для `Hibernate` необходима настройка сессии. Настройка ее производит в конфигурационный файлах `Spring` фреймворка.

После того как была проведена необходимая настройка можно приступать к связыванию таблиц базы данных с классами (бинами). Чтобы сказать фреймворку, что класс является бином достаточно просто указать аннотацию `@Entity` перед именем класса. В этом случае будет использоваться таблица точно с таким же именем, как и имя класса. Если необходимо класс связать с таблицей имя которой отличается от имени класса нужно использовать аннотацию `@Table(name = "storageTypes")`, где в скобках указать имя нужной таблицы. Класс, который объявлен как `@Entity` должен содержать в себе гетеры и сетеры для всех своих полей.

Последним шагом является установление зависимостей между таблицами. В структурированном языке запросов используются внутренние и внешние ключи для связывания таблиц. В `Hibernate` используются аннотации `@OneToMany`, `@OneToOne`, `@ManyToMany`.

В результате получены классы, которые выступают в роли таблиц баз данных с теми же свойствами полей. Разработчику достаточно использование `Hibernate API` для управления базой данных.

**Д.В. Будько (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **В.Д. Левчук**, к.т.н., доцент

## **УПРАВЛЕНИЕ ДОСТУПА К ДАННЫМ В СИСТЕМЕ ЗАКАЗОВ НЕМАТЕРИАЛЬНЫХ РЕСУРСОВ**

Основными фреймворками, которые были использованы для реализации системы заказов, являются `Spring` и `Hibernate`.

`Hibernate` предоставляет удобный механизм, который дает возможность разработчику работать с таблицами базы данных как с объектами языка программирования. Для того, чтобы добиться этого результата, необходимо провести предварительную настройку фреймворка. В первую очередь необходимо описать фильтр `Hibernate`. Данный фильтр описывается в дескрипторе развёртывания веб-приложения, в нашем случае используется класс `OpenSessionInViewFilter`.

Для работы с базой данных приложению необходимо иметь подключение к ней. Есть несколько вариантов решения этой задачи, но в ходе реализации проекта было принято решение использовать `dataSource`. Использование `dataSource` позволит нам снизить время затрачиваемое для установления подключения к базе. Также для Hibernate необходима настройка сессии. В момент, когда происходит инициализация сессия в ней находится информация обо всех классах, которые представляют собой таблицы базы данных. Также указываются настройки обращения фреймворка к базе данных. В следующем листинге приведен пример настройки бина сессии.

```
<bean id="sessionFactory"
      class="org.springframework.orm.hibernate4.LocalSessionFactoryBean"
      >
  <property name="dataSource" ref="dataSource" />
  <property name="annotatedClasses">
    <list><value>com.epam.supply.beans.Role</value></list>
  </property>
  <property name="hibernateProperties"><props>
    <prop key="hibernate.dialect"> org.hibernate.dialect.MySQLDialect
    </prop></property>
</bean>
```

После того как была проведена необходимая настройка можно приступать к связыванию таблиц базы данных с классами (бинами). Чтобы сказать фреймворку, что класс является бином, достаточно просто указать аннотацию `@Entity` перед именем класса. В этом случае будет использоваться таблица точно с таким же именем, как и имя класса. Если необходимо класс связать с таблицей имя которой отличается от имени класса нужно использовать аннотацию `@Table(name = "storageTypes")`, где в скобках указать имя нужной таблицы. Класс, который объявлен как `@Entity` должен содержать в себе геторы и сеторы для всех своих полей. Схема базы данных проекта приведена к третьей нормальной форме, некоторые поля таблиц имеют отношение многие-ко-многим. На основании этого было создано более 15 классов (бинов), которые представляют данную базу данных. Примером объявления бина служит листинг.

```
@Entity
@Table(name = "users")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
```

```

@Column(name = "userId")
private int id;
@Column(name = "name")
private String login;
@ManyToOne(fetch = FetchType.EAGER)
@JoinColumn(name = "roomId")
private Room room;
private boolean notify;
@ManyToMany(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
@JoinTable(name = "users_roles",
            joinColumns = { @JoinColumn(name = "userId") },
            inverseJoinColumns = { @JoinColumn(name = "roleId") })
private Set<Role> roles = new HashSet<Role>(0);

```

Как показано на листинге в данном классе (User) используется связь многие-ко-многим. Это применяется к полю ролей пользователя, так как каждый сотрудник может иметь несколько должностей в компании, но связь многие-к-одному поля комнаты показывает, что сотрудник может иметь только одно рабочее место. Также в таблице представлено объявление уникального поля и метода его генерации.

Последним шагом является установление зависимостей между таблицами. В структурированном языке запросов используются внутренние и внешние ключи для связывания таблиц. В Hibernate используются аннотации `@OneToMany`, `@OneToOne`, `@ManyToMany`.

В результате получены классы, которые выступают в роли таблиц баз данных с теми же свойствами полей. Разработчику достаточно использование Hibernate API для управления базой данных. Для этого необходимо воспользоваться сессией, которую мы настраивали в самом начале. Пример использования приведен в листинге.

```

Criteria criteria = sessionFactory
    .getCurrentSession().createCriteria(User.class);
criteria.add(Restrictions.eq("login", name));
users = criteria.list();

```

Объект `Criteria` используется для указания того, как мы будем выбирать данные из таблиц. В данном примере используется критерия поиска сотрудника по его имени. Если такое было найдено в таблице то оно вернет список с одним или несколькими совпадениями или же `null`.

**О.А. Васильев (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Е.А. Левчук**, к.т.н., доцент

## **РАЗРАБОТКА КЛИЕНТА ПЕРЕДАЧИ ДАННЫХ ДЛЯ WEB-СЕРВИСА AMAZON GLACIER**

На данный момент существует огромное множество сервисов хранения данных, однако большая часть из них предназначены для временно-го хранения данных с негарантированной сохранностью, другие сервисы предназначены для более узкого круга пользователей и являются платными, но несут ответственность за сохранность данных и гарантируют высокий процент безопасности информации. Одним из таких веб-сервисов является недавно запущенный, около нескольких месяцев назад, Amazon Glacier входящий в группу Amazon Web Services (Amazon предоставляет огромное количество различных сервисов, которые могут быть востребованы пользователями: от почтовых сервисов (SMS) до сервисов хранения (Glacier и S3)).

Amazon Glacier – это наиболее недорогой сервис хранения данных, обеспечивающий безопасность и надежность хранения архивов информации и всевозможных бэкапов. Для того чтобы снизить стоимость хранения данных, Amazon Glacier был оптимизирован для хранения информации, к которой не требуется частого доступа и для которой время извлечения из хранилища в течении нескольких часов является допустимым. С помощью Amazon Glacier, пользователи могут надежно сохранять большие или малые объемы информации по предельно низкой цене, где-то около 1 цента, что значительно ниже, чем использование локальных решений.

Множество компаний обычно переплачивают за архивацию своей информации. Во-первых, они вынуждены реализовывать дорогие решения для архивации (причем в стоимость решения могут не входить такие пункты как: электропитание, поддержка, обеспечение и прочее). Во-вторых, они обязаны знать, в каких объемах представлена информация, которую они собираются архивировать, и готовы ли они к неожиданному росту количества информации. Стечение всех вышеперечисленных обстоятельств приводит к тому, что большая часть денег тратится впустую. Amazon Glacier меняет правила архивирования данных и резервного копирования, пользователь платит только за те ресурсы, которые он реально использует, он не платит заранее, платит предельно низкую стоимость за хранения информации, а использование ресурсов может масштабироваться пользователем в сторону увеличения или уменьшения.

Стоит отметить что на данный момент нету достойных клиентский приложений для работы с данным web-сервисом, поэтому темой моей дипломной работы была выбрана “Разработка клиента передачи данных для веб-сервиса Amazon Glacier”. Разработка приложения ведется на языке Java с использованием библиотеки AWS SD, данная библиотека предоставляет разработчикам API для доступа к сервисам Amazon, что позволяет сократить время разработки и отладки приложения в разы, упрощается доступ к сервисам, нет необходимости писать собственный REST-клиент. Сама платформа Java позволяет организовать многопоточное приложение, которое позволяет пользователю гибко взаимодействовать с хранилищем. Так же стоит отметить что парадигма Java: “написал один раз – работает везде”, гарантирует платформонезависимость кода, а само приложение будет стабильно работать на большинстве популярных операционных систем.

**О.А. Васильев (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Е.А. Левчук**, к.т.н., доцент

## **ПРИМЕНЕНИЕ JSON В ПОСТРОЕНИИ КЛИЕНТА ПЕРЕДАЧИ ДАННЫХ ДЛЯ WEB-СЕРВИСА AMAZON GLACIER**

Клиент передачи данных для Amazon Glacier представляет собой клиент-серверное приложение, и как любой приложение подобного рода, должен уметь взаимодействовать с удаленным сервисом, в данном случае набор серверов AWS и в частности сервис Amazon Glacier.

При разработке клиент-серверного приложения основным вопросом стоит то, каким образом клиент и сервер будут взаимодействовать друг с другом. Существует множество технологий, которые позволяют реализовать подобное взаимодействие. Самые популярные технологии это JSON и SOAP. SOAP представляет собой основанный на XML протокол для передачи данных через HTTP, однако данная технология не поддерживается у рассматриваемого веб-сервиса, поэтому выбор остался за JSON. JSON (JavaScript Object Notation) + REST (Representational State Transfer) представляет собой легко имплементируемую связку, а ее преимущества перед SOAP (читаемость, легкая имплементируемость, объем данных меньше по сравнению с XML) дают понять, почему Amazon сделали при построении своего веб-сервиса выбор в пользу JSON+REST.

Наиболее простую схему работы приложения можно представить следующим образом: приложение генерирует HTTP запрос с необходимыми

заголовками (к примеру, HTTP запрос для получения описания хранилища) и отправляет его удаленному сервису, сервис обрабатывает сообщение, собирает требуемые данные, компонует их в формате JSON и отправляет их в виде HTTP ответа приложению, далее приложение парсит полученные данные и сохраняет в базе данных либо выводит в удобном для пользователя формате.

В качестве основного средства для обработки JSON в разрабатываемом приложении использовались как средства AWS SDK так и сторонняя java библиотека `json-simple`, данная библиотека позволяет с легкостью обрабатывать JSON данные, как из файла, так и из внешнего потока.

Для самого приложения была разработана обертка, которая позволяет парсить JSON сообщения, для которых не предусмотрено методов обработки средствами AWS SDK. К примеру, таким образом обрабатывается инвентаризация хранилища (список всех файлов и их характеристик хранимых на удаленном сервере).

**О.А. Васильев (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Д. Левчук**, к.т.н., доцент

## **ОРГАНИЗАЦИЯ МНОГОПОТОЧНОСТИ В КЛИЕНТЕ ПЕРЕДАЧИ ДАННЫХ ДЛЯ WEB-СЕРВИСА AMAZON GLACIER**

Клиент передачи данных для Amazon Glacier представляет собой достаточно сложное многопоточное приложение.

Одними из важнейших вопросов при построении многопоточного приложения было то, как приложение будет управлять потоками, и как потоки будут взаимодействовать с общими разделяемыми данными.

Различные высокоуровневые языки программирования предоставляют свои средства для организации потоков (хотя более правильное название “нить”, потоками обычно называют потоки ввода-вывода), однако у каждого из языков свои возможности для работы с потоками, в каких-то они обширны, в других более скромные.

Язык программирования Java, предоставляет для разработчика средства для организации многопоточной обработки, однако, следуя парадигме языка, эти возможности очень ограничены с целью обеспечить безопасность кода.

Основное средство языка для создания потока инструкций это класс `Thread`. `Thread` удобное средство для создания нового потока, которое позволяет легко запустить поток и контролировать его состояние.

При разработке использовалась библиотека `java.util.concurrent` в частности класс `ExecutorService`, что позволило реализовать возможность возврата

потоком результатов работы, таким образом, поток может уведомить (применяется интерфейс Future) приложение и пользователя о том, как он был завершен или вернуть результат своей работы. В данном приложении данная функциональность позволила реализовать отображение состояний потоков. По окончании, например, потока загрузки файла на удаленный сервер поток вернет состояние, в котором он завершил выполнение (успешное выполнение, провал, прерывание).

Так как Java не предоставляет средств для реализации мгновенного прерывания потока, так как это нарушает безопасность кода, и результат подобного прерывания может привести к непредсказуемому поведению приложения. В связи с этим были созданы собственные обертки для потоков и реализована проверка сигналов о прерывании внутри самого потока с определенной периодичностью. Таким образом удалось реализовать аккуратное прерывание потоков, которое подразумевает закрытие использованных потоком ресурсов, и уведомление пользователя о прерывании.

**Р.А. Веренчиков (УО «ГГУ им.Ф.Скорины», Гомель)**

Науч. рук. **Е.М. Березовская**, к.ф.-м.н., доцент

## **РЕАЛИЗАЦИЯ ШАБЛОНА MODEL VIEW CONTROLLER ПРИ СОЗДАНИИ WEB-СИСТЕМЫ УПРАВЛЕНИЯ ПРОЕКТАМИ**

Java Platform, Enterprise Edition (Java EE или J2EE) – набор спецификаций и соответствующей документации для языка Java, описывающей архитектуру серверной платформы для задач средних и крупных предприятий. Model View Controller это самый популярный архитектурный паттерн, используемый для разработки веб-приложений.

Реализация паттерна заключается в разделении приложения на три функциональных слоя: модель, представление, контроллер. При этом приложение, соответствующее этим принципам, становится более понятным с точки зрения разработки, поддержки и совершенствования, а отдельные его части довольно легко могут быть использованы повторно.

Модель отвечает за управление данными, она сохраняет и извлекает сущности, используемые приложением, как правило, из базы данных и содержит логику, реализованную в приложении. В данном паттерне модель не зависит от представления или управляющей логики, что делает возможным проектирование модели как независимого компонента и, например, создавать несколько представлений для одной модели.

Представление несет ответственность за отображение данных, которые даёт контроллер. С представлением тесно связано понятие шаблона,

который позволяет менять внешний вид показываемой информации. В веб-приложении представление часто реализуется в виде HTML, JSP страниц.

Контроллер связывает модель и представление. Он получает запрос от клиента, анализирует его параметры и обращается к модели для выполнения операций над данными запроса. От модели поступают уже скомпонованные объекты. Затем они перенаправляются в представление, которое передаёт сформированную страницу контроллеру, который, отправляет её клиенту.

В разработанном приложении поток выполнения приложения всегда обязан проходить через контроллер приложения. Контроллер направляет запросы – в данном случае HTTP(S)-запросы – к соответствующему обработчику. Обработчики запроса связаны с бизнес-моделью, и каждый разработчик приложения должен только обеспечить взаимодействие между запросом и бизнес-моделью. В результате реакции системы на запрос вызывается соответствующая JSP-страница, выполняющая в данной схеме роль представления.

**А.А. Витко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **П.Л. Чечет**, к.т.н., доцент

## **СИСТЕМА УПРАВЛЕНИЯ ПРОИЗВОДСТВЕННЫМ ОБОРУДОВАНИЕМ SIMATIC PCS7**

Для построения систем автоматизации производственных и технологических процессов всё чаще и чаще используются распределённые системы. Это означает, что весь комплекс задач управления распределяется между отдельными системами, выполняющими менее крупные подзадачи. В результате возникает необходимость в организации эффективной связи между распределёнными системами. Развитая высокопроизводительная система связи является неотъемлемой частью распределённой системы управления. Требования, предъявляемые к системам связи, работающим в условиях промышленного производства, существенно отличаются от требований к традиционным офисным системам связи.

В данной статье рассматривается одна из ведущих в мире систем управления процессами, SIMATIC PCS7, которая с её функциональным многообразием, гибкостью и производительностью имеет потенциал для реализации инновационных решений, удовлетворяющих особым требованиям непрерывного производства.

PCS7-проекты разрабатываются на инженерных станциях(ES), где



установлены инструменты разработки и имеется коммуникационный доступ к станциям автоматизации(AS) и операторским станциям(OS).

Разработка PCS7-проектов на станциях ES может быть разделена на две фазы: AS-проектирование и OS-проектирование. AS-проектирование включает разработку иерархию объекта управления, функциональных блоков, CFC и SFC-схем, конфигурирование аппаратных средств и компонентов коммуникации. После фазы AS-проектирования проект переносится (направляется) в WinCC, где OS-проектирование начинается с создания операторских функций и графики.

Система операторского управления представляет собой человеко-машинный интерфейс к системе управления процессом SIMATIC PCS7 и поэтому является для пользователя окном в процесс.

Аппаратное обеспечение всех операторских станций, основанное на современных компьютерных технологиях, может быть оптимизировано для работы в качестве однопользовательской ОС, клиентской ОС или сервера.

Многопользовательская система в архитектуре клиент-сервер состоит из операторских терминалов (клиентских ОС), которые получают данные (проектные данные, значения процесса, архивы, аварийные сигналы и сообщения) от одного или более ОС-серверов по локальной сети ОС (терминальной шине). Локальная сеть операторских станций может быть объединена с системной шиной в единую сеть, либо спроектирована как отдельная независимая сеть Ethernet с протоколом TCP/IP.

В системе с такой архитектурой для повышения надежности могут использоваться резервированные ОС-серверы. При этом осуществляется непрерывный взаимный контроль за приложениями, исполняющимися на ОС-серверах, на предмет возникновения ошибок программного обеспечения. При возникновении ошибки автоматически выполняется переключение клиентов на резервный сервер. При восстановлении вышедшего из строя сервера автоматически и с высокой скоростью происходит синхронизация данных ОС-серверов. Такая архитектура представлена на рисунке 1.

ОС-клиенты могут иметь одновременный доступ к данным нескольких ОС-серверов/резервированных пар серверов (режим мультиклиентов). Это позволяет делить всю установку на технологические блоки и соответственно распределять данные между несколькими различными ОС-серверами/парами серверов.

Высокопроизводительная встроенная система архивирования на отдельных станциях и ОС-серверах основана на использовании технологии Microsoft SQL Server, поддерживает настройку в режиме исполнения и сохраняет значения параметров процесса (типичный период от 1 до 4 недель) и сообщения в циклических архивах (типичный период 2 месяца).

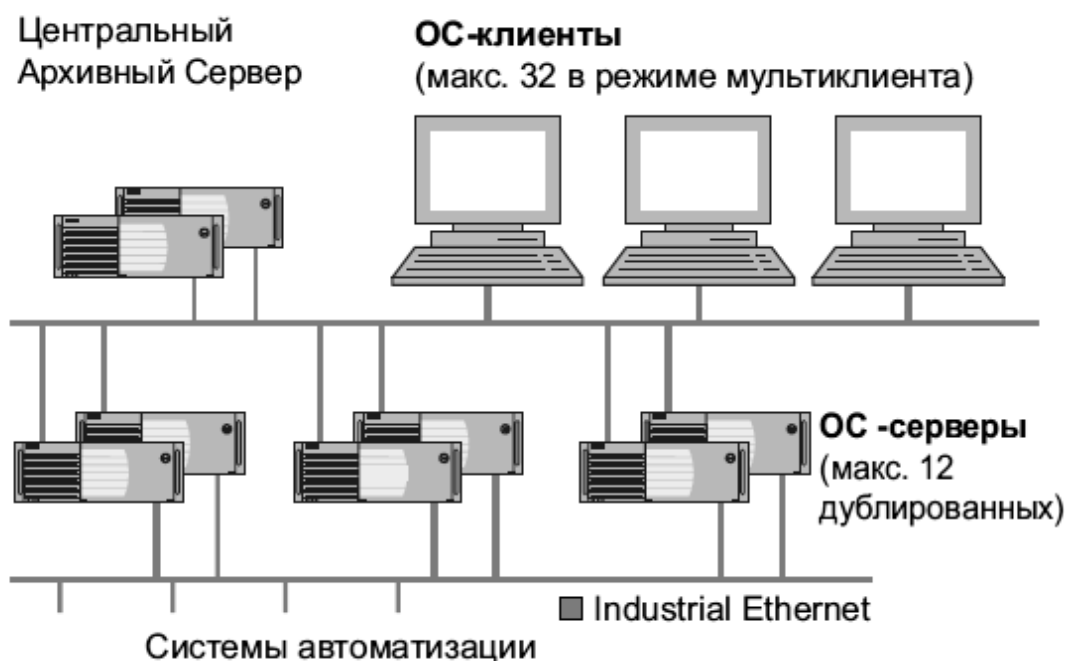


Рисунок 1 – Многопользовательская система с архитектурой клиент-сервер

Системная и терминальная шина для многопользовательских систем в архитектуре клиент-сервер реализуются с помощью Industrial Ethernet, мощной сети предназначенной для промышленного применения, соответствующей международному стандарту IEEE 802.3.

В различных подсистемах SIMATIC PCS7 (ES, OS, AS и т.д.) в качестве коммуникационных интерфейсов используются встроенные интерфейсные модули, простые сетевые карты или специальные коммуникационные процессоры (напр., CP 1613 A2/CP 1623).

В установках, отличающихся высокими требованиями, SIMATIC PCS7 ориентирован на использование мощных коммуникационных процессоров, а также на современную гигабитную технологию и технологию FastEthernet. Они объединяют в себе высокую надежность оптических колец с наращиваемой благодаря коммутационным технологиям производительностью и скоростями передачи до 1 Гбит/с.

Таким образом, однородная и унифицированная система управления технологическими процессами SIMATIC PCS7 с ее расширяемой архитектурой и универсальными системными характеристиками является отличной основой для экономически эффективной реализации и экономичной эксплуатации устройств управления процессами.

### Литература

1. Бергер, Г. Работа с пакетом PCS 7 / Г. Бергер / Siemens AG, 2001 – С. 4–61.

2. Официальный сайт компании SIEMENS [Электронный ресурс]. / 2004–2013. – Режим доступа: <http://www.ad.siemens.de/simatic-pcs7>. – Дата доступа: 14.03.2013.

**А.С. Воробьева, А.И. Кучеров (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **А.И. Кучеров**, м.т.н., старший преподаватель

## **ИССЛЕДОВАНИЕ ИНТЕРНЕТ-МАГАЗИНОВ РЕСПУБЛИКИ БЕЛАРУСЬ**

Современные технологии сказываются на всех аспектах жизни человека. Так, например, большую часть новостей сейчас узнается населением не по телевизору и не из газет, а из сети Интернет. Интернет позволяет общаться людям с разных концов света, получать свежую информацию о положении в мире и даже совершать покупки. Интернет-магазины сегодня вещь очень распространенная и с каждым годом их становится все больше и больше.

Интернет-торговля считается одним из наименее затратных способов розничной реализации товаров. Ведь создать виртуальный магазин гораздо проще и дешевле, чем, допустим, открыть реальную торговую точку, и потребителю проще. Ни для кого не тайна, что цены в интернет-магазине чаще всего ниже, чем в реальном магазине, да и проще сделать заказ, не выходя из дома, и товар будет доставлен в срок в руки заказчику.

По официальным данным количество интернет-магазинов в Беларуси растет. Если в 2009 году их было только 800, то к концу 2012 года их стало уже свыше 4000.

Основная масса интернет-магазинов – это субъекты хозяйствования города Минска. Торговый реестр показывает такое распределение интернет-магазинов по белорусским регионам:

- Минск — 75 %;
- Минская область — 12 %;
- Брестская – 3 %;
- Витебская – 2 %;
- Гомельская – 3 %;
- Гродненская – 2,5 %;
- Могилевская – 2,5 %.

Существует большое количество каталогов интернет-магазинов, в которых можно получить список магазинов определенного профиля. Распределение по профилям приведено в таблице 1.

Так как не все регистрируют свои магазины в каталогах, суммарные данные не совпадают с исходными.

Самыми распространенными в 2012 году являются магазины со следующими специализациями: сложнотехнические товары, бытовая техника, книги, аудио- и видеозаписи, компьютерные игры, цветы и подарки, стройматериалы, автозапчасти, туристические услуги, мебель.

Таблица 1 – Распределение профилей

№	Наименование	Кол-во	№	Наименование	Кол-во
1	Бизнес	274	17	Игры	64
2	Канцтовары	4	18	Книги	30
3	Офисная техника	145	19	Музыка и фильмы	111
4	Предоплаченные сервисы	2	20	Подарки	69
5	Программное обеспечение	2	21	Туризм	10
6	Связь	170	22	Фототовары	123
7	Спецтехника	8	23	Семья и дом	324
8	Услуги	10	24	Авто, мото	38
9	Здоровье и красота	60	25	Аудио и видео техника	131
10	Аксессуары и украшения	4	26	Бытовая техника и электроника	172
11	Косметика и парфюмерия	21	27	Гастрономия	22
12	Медицина	15	28	Домашнее хозяйство	64
13	Спорттовары	21	29	Животные	1
14	Цветы и растения	3	30	Мебель	66
15	Развлечение и отдых	318	31	Одежда и обувь	9
16	Аукционы	3		Итого	2174

На данный момент преобладает наличный расчет за товары, приобретенные в интернет-магазине. В 2013 году все интернет-магазины перейдут на безналичный расчет. Такая технология позволит лучше защитить права потребителя и обеспечить продавцам легальные условия бизнеса. Кроме того, информация обо всех совершенных сделках теперь будет храниться не на бумаге, а в интернет-пространстве, чтобы полностью исключить возможность обмана потребителя.

### Литература

1. BY Shops [Электронный ресурс]. – 2012. – Режим доступа: <http://www.byshops.com> – Дата доступа: 26.12.2012.
2. GeoSeo, Маркетинг по-нашему [Электронный ресурс]. – 2012. – Режим доступа: <http://geoseo.by> – Дата доступа: 26.12.2012.
3. СТВ столичное телевидение [Электронный ресурс]. – 2012. – Режим доступа: <http://www.ctv.by> – Дата доступа: 26.12.2012.

Д.В. Гетиков (УО «ГГУ им.Ф. Скорины», Гомель)

Науч. рук. М.И. Жадан, к.ф.-м.н., доцент

## ИСПОЛЬЗОВАНИЕ АГЕНТОВ В WEB-ПРИЛОЖЕНИЯХ, СОЗДАНЫХ В КОРПОРАТИВНОЙ СРЕДЕ ДОКУМЕНТООБОРОТА LOTUS NOTES

В наше время любое производство немислимо без использования разнообразных систем управления производства, различной сложности и масштабов. При планировании, а также использовании как структурной, так и программной составляющей управления производства в наше время все чаще специалисты прибегают к созданию разнообразных баз данных для хранения, передачи, обработки информации, а также использованию специального программного обеспечения для поддержки такого количества информации – Систем Управления Базами Данных.

На рынке СУБД особое место занимает продукт Lotus Notes. Обуславливается это возможностью использования документно-ориентированной модели базы данных при использовании реляционных баз данных как источника данных. Изначально доступные лишь при помощи клиента Lotus Notes, приложения ныне могут быть спроектированы для работы под Web, используя обычный Web-браузер в качестве клиента доступа к данным. В последнее время данный способ доступа набирает популярность, по этой причине очень частыми являются случаи переориентации приложений под использование в Web.

В любом приложении одним из главных принципов работы является автоматизация обработки данных путем создания специальных программ-агентов, которые могут быть запущены в любое время, по расписанию либо вручную. При написании программ агентов для использования под Web, разумеется, есть свои особенности.

В Web-приложениях агенты наиболее часто используются в качестве обработчиков событий формы WebQueryOpen и WebQuerySave, а также как агенты, запускаемые по URL **...БД/агент?OpenAgent**. В последнем случае, такой URL может генерироваться сервером Domino и самостоятельно. Например, в акции формы @-командой *ToolsRunMacro* вызывается агент, если при этом у БД не установлен флаг *Use JavaScript when generation pages*, то акция формы транслируется в тег гипертекстовой ссылки <A HREF ...>, в атрибут HREF которой передается URL **...БД/агент?OpenAgent**.

Агент **WebQueryOpen** выполняется перед тем, как сервер Domino конвертирует документ в HTML для передачи его в браузер. Разработчики имеют возможность выполнить определенные действия (например, изменить начальное значение определенных полей) перед открытием документа

в Web. Для получения доступа к значениям полей текущего документа из WebQueryOpen агентов, написанных на языках LotusScript и Java, должны использоваться: свойство DocumentContext класса NotesSession для LotusScript и свойство DocumentContext класса AgentContext для Java.

Агент **WebQuerySave** выполняется перед сохранением документа в интерфейсе браузера. Используя данный тип агента, можно выполнить определенные действия, включая запрет на сохранение некорректных документов, определение страницы, на которую нужно перейти после сохранения и т. д.

WebQuerySave агенты, написанные на LotusScript и Java, могут «на лету» сгенерировать страницу, которая будет выводиться по факту сохранения документа. Для LotusScript это достигается обычным оператором языка Print, а для Java-методом println класса PrintWriter. В обоих случаях разработчик операторами печати формирует только содержимое тега <BODY>, остальные теги и их атрибуты генерируется сервером Domino. Кроме этого, с помощью оператора печати в теле агента можно осуществить редирект (переход) на другую HTML-страницу. Это достигается выводом URL требуемой страницы, заключенным в квадратные скобки.

В качестве примера можно использовать обычное действие, допустим, автоматизированный ввод значения в определенное поле. Далее изложены примеры этого самого действия, реализованного разными языками программирования:

- На языке Simple Actions:  
устанавливаем для объекта **WebQueryOpen** значение `@Command([ToolsRunMacro]; "WQO")`. У агента в качестве действий выбираем Action – выбрано значение Modify Field, Field – a1, Value – значение, добавленное WebQueryOpen агентом.
- На языке @Formula:  
`@SetField("a1"; a1 : "значение, добавленное WebQueryOpen агентом")`
- На языке Lotus Script:  

```
Sub Initialize
  Dim session As NotesSession
  Dim doc As NotesDocument
  Dim item As NotesItem
  Set session = New NotesSession
  Set doc = session.DocumentContext
  Set item = doc.GetFirstItem("a1")
  Call item.AppendToTextList("значение, добавленное WebQueryOpen агентом")
End Sub
```

– На языке Java:

```
import lotus.domino.*;
public class JavaAgent extends AgentBase {
public void NotesMain() {
    try {
        Session session = getSession();
        AgentContext agentContext = session.getAgentContext ();
        // (Your code goes here)
        Document doc = agentContext.getDocumentContext();
        Item item = doc.getFirstItem( "al" );
        item.appendToTextList( "значение, добавленное
WebQueryOpen агентом" );
    } catch(Exception e) {
        e.printStackTrace ();
    }
}
}
```

Как результат, мы получаем приложения, использующие всю мощь использования Simple Actions, @formula, Java, LotusScript для написания программ-агентов в среде Lotus Notes для работы с документами посредством Web-браузера.

При помощи использования агент-ориентированного подхода разработки набора функциональных методов, была разработана и реализована база данных «Грузоперевозки», пользовательский интерфейс которой позволяет быстро получить необходимую информацию по данной базе по каждой из имеющихся сущностей: путевые листы, справочник адресов, платежи, водители автомобилей.

Таким образом, Lotus Notes предстает как великолепное кросс-браузерное решение при работе с документами под Web для малых и крупных предприятий, использующих данную технологию для ведения документооборота.

**И.Г. Глушаков (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.А. Дробышевский**, старший преподаватель

## **АВТОМАТИЗАЦИЯ РЕГИСТРАЦИИ ТЕХНИКО-ТАКТИЧЕСКИХ ИГРОВЫХ ДЕЙСТВИЙ ФУТБОЛИСТОВ**

Сущностью проекта является создание программы «Регистратор действий футболистов во время матча» для дальнейшей оценки технико-

тактических действий отдельных игроков и команд в целом.

Подавляющее большинство иностранных футбольных тренеров, а с недавнего прошлого и новое поколение футбольных функционеров постсоветского пространства, огромное внимание уделяют математическим подсчетам тех или иных аспектов игры.

Общий принцип подсчета заключается в детальнейшем разборе показателей каждого конкретного футболиста на поле, начиная от простых передач мяча и заканчивая сложными тактическими маневрами. В конечном итоге данные суммируются, соотносятся удачные и неудачные действия и выводятся разного рода числовые и процентные показатели. Сами по себе цифры незадачливому тренеру ничего не скажут, но находящемуся в вечном поиске наставнику они послужат громадным подспорьем в выполнении работы над ошибками, составлении тренировочного плана и установок на игры.

При создании данного проекта была реализована концептуальная модель базы данных, были определены сущности и атрибуты сущностей. Была выделена одна основная сущность. Из этих данных составлена логическая модель базы данных, схема данных приведена на рисунке 1.

В логической модели базы данных использованы связи один ко многим. Это означает, что один экземпляр первой сущности взаимодействует с несколькими экземплярами другой сущности. Взаимосвязи отображаются линиями, соединяющими две сущности. Каждая сущность делится на 2 группы. В первой группе находятся атрибуты, называемые первичным ключом. Первичный ключ – это набор атрибутов, выбранных для идентификации уникальных экземпляров сущности. Первичный ключ нужен для того, чтобы от него создавать связи между другими таблицами.

При создании сущности необходимо выделить атрибуты, которые могут стать первичным ключом, затем произвести отбор атрибутов, следуя следующим рекомендациям:

1 Первичный ключ должен быть подобран таким образом, чтобы по значениям атрибутов, можно было точно идентифицировать экземпляр сущности;

2 Никакой из атрибутов первичного ключа не должен иметь нулевое значение;

3 Значения атрибутов первичного ключа не должны меняться. Если значение изменилось, значит, это уже другой экземпляр сущности.

В базе данных первичный ключ это «Код», так как он не допускает ввода пустых значений, с его помощью можно идентифицировать экземпляр сущности, и его значение не меняется.



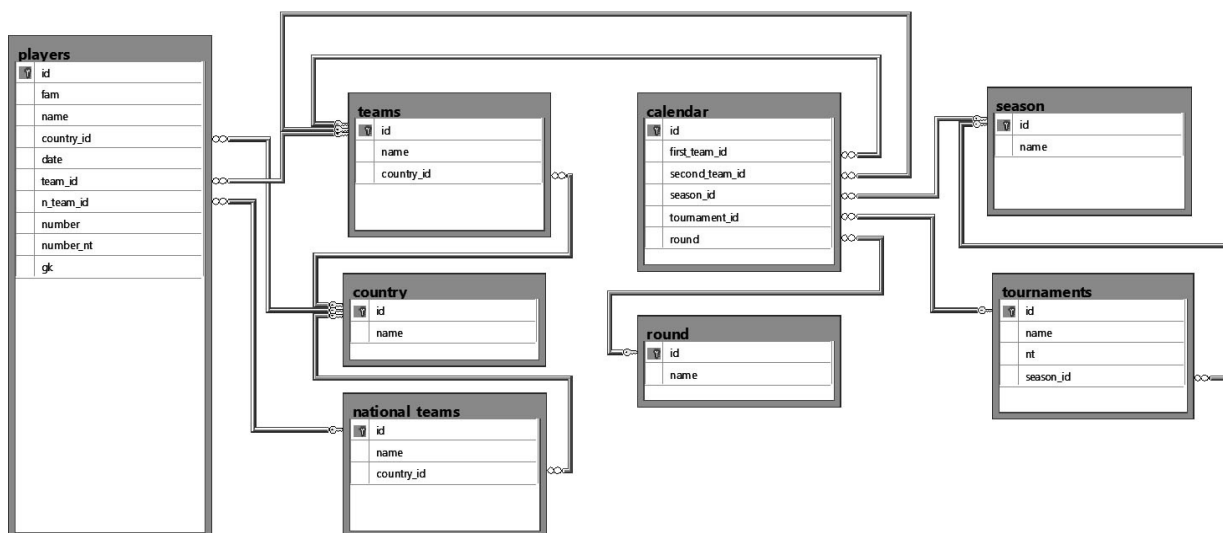


Рисунок 1 – Организация связей между таблицами

На следующем этапе построения логической модели определяем ключевые атрибуты и типы атрибутов. Далее проводим нормализацию базы данных.

Нормализация – процесс проверки и реорганизации сущностей и атрибутов с целью удовлетворения требований к реляционной модели данных. Нормализация позволяет быть уверенным, что каждый атрибут определен для своей сущности, значительно сократить объем памяти для хранения данных.

Целью создания физической модели является обеспечение администратора соответствующей информацией для переноса логической модели данных в СУБД.

В физической модели каждой сущности будет соответствовать таблица базы данных, а каждому атрибуту – поле таблицы. Имена таблиц и полей лучше задавать латинскими буквами и достаточно короткими для удобства программирования и совместимости с системами, не поддерживающими кириллицу.

Для успешной работы с данными между таблицами установлены связи. Для поиска и отбора данных, удовлетворяющих определенным условиям, организованы запросы. Запросы также позволяют обновить или удалить одновременно несколько записей, выполнить встроенные и специальные сообщения.

Для просмотра, ввода или изменения данных прямо в таблице применяются формы. Форма позволяет отобразить данные из одной или нескольких таблиц и вывести их на экран, используя стандартный или созданный пользователем макет.

В окне базы данных можно работать со всеми её объектами. Для просмотра объектов определённого типа следует выбрать соответствующую

вкладку. С помощью кнопок можно открывать и изменять существующие объекты и создавать новые.

Удобный интерфейс программы, с одной стороны, позволяет легко ориентироваться в программе, не требуя от пользователя каких-либо специальных навыков работы с электронно-вычислительными машинами, с другой стороны предоставляет пользователю оперативную информацию.

**А.С. Гончаров (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

## ТЕСТ ТЬЮРИНГА И ВИРТУАЛЬНЫЕ СОБЕСЕДНИКИ

Тест Тьюринга – эмпирический тест, идея которого была предложена Аланом Тьюрингом в статье «Вычислительные машины и разум» (англ. Computing Machinery and Intelligence), опубликованной в 1950 году в философском журнале «Mind». Тьюринг задался целью определить, может ли машина мыслить.

Стандартная интерпретация этого теста звучит следующим образом: «Человек взаимодействует с одним компьютером и одним человеком. На основании ответов на вопросы он должен определить, с кем он разговаривает: с человеком или компьютерной программой. Задача компьютерной программы – ввести человека в заблуждение, заставив сделать неверный выбор».

Элиза (ELIZA) – виртуальный собеседник, знаменитая компьютерная программа Джозефа Вейценбаума, написанная им в 1966 году, которая пародирует диалог с психотерапевтом, реализуя технику активного слушания (рисунок 1).

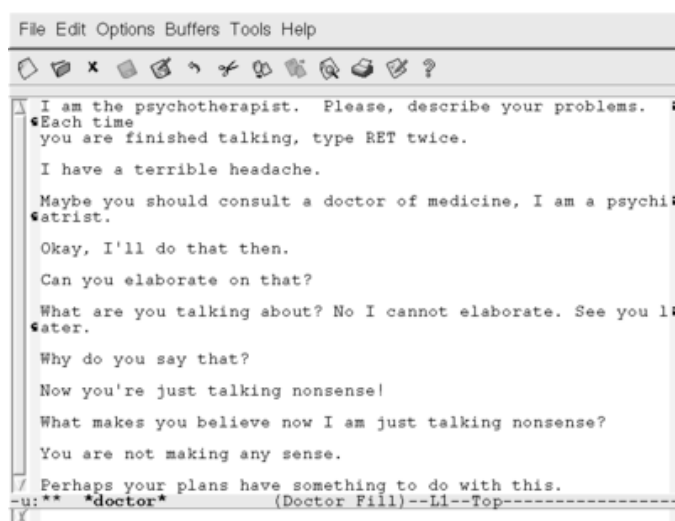


Рисунок 1 – Диалог с программой «Элиза»

Программа стала первым шагом в преодолении теста. Она использовала синтаксический анализатор, выделяя ключевые слова и находя соответствующие ответы в своей базе данных. В 1966 году интерактивные компьютеры были в новинку. Появление Элизы дало толчок прогнозам футурологов относительно будущих виртуальных собеседников и появления «разумных» компьютеров уже около 2000 года.

Неправильное отношение к программе беспокоило Вейзенбаума и заставило его написать книгу «Возможности вычислительных машин и человеческий разум. От суждений к вычислениям», в которой он разъясняет пределы возможностей компьютеров.

Пэрри (PARRY) – помимо Элизы (ELIZA), другой известный ранний виртуальный собеседник (chatbot). PARRY был написан в 1972 году психиатром Кеннетом Колбай, работавшим тогда в Стэнфордском Университете. PARRY пытался моделировать параноидального шизофреника. Программа реализовала грубую модель поведения параноидального шизофреника, основанную на понятиях, представлениях, и вере (суждениях об представлениях: принять, отвергнуть, отнестись нейтрально). Кеннет также воплотил диалоговую стратегию, и PARRY был намного более серьезной и продвинутой программой чем ЭЛИЗА.

A.L.I.C.E. – современный виртуальный собеседник, программа, способная вести диалог с человеком на естественном языке. A.L.I.C.E. – аббревиатура от англ. Artificial Linguistic Internet Computer Entity, что дословно можно перевести как «Искусственное лингвистическое интернет-компьютерное существо». Но это название было подобрано в соответствии с именем компьютера, «Alice», на котором программа впервые была запущена. Она создана по подобию первой программы-собеседника, Элизы, и использует технику эвристического сопоставления фразы пользователя с образцами в базе знаний. A.L.I.C.E. – одна из лучших в своём роде, она три раза (в 2000, 2001, 2004 годах) становилась победителем премии Лёбнера.

Премия Лёбнера (англ. Loebner prize) – премия, присуждаемая победителю ежегодного конкурса (проводится с 1990г.), в котором соревнуются программы в прохождении теста Тьюринга. Наиболее «человечной», по мнению судей, вручается премия в \$2000. По словам Лёбнера, соревнование было организовано с целью продвижения вперед в области исследований, связанных с искусственным интеллектом, отчасти потому, что «никто не предпринял мер, чтобы это осуществить».

Стоит заметить, что полного диалога с машиной пока не существует, а то, что есть больше напоминает общение в кругу друзей, когда отвечаешь на вопрос одного, а следом задает вопрос другой или как бы на твой

вопрос отвечает совершенно посторонний. На этом, в принципе и можно ловить машинную программу.

Судейство на конкурсе очень строгое. Эксперты заранее готовятся к турниру и подбирают весьма заковыристые вопросы, чтобы понять, с кем же они общаются. Их разговор с программами напоминает допрос следователя. Судьи любят, например, повторять некоторые вопросы через определенное время, так как слабые боты не умеют следить за историей диалога и их можно поймать на однообразных ответах.

Пока что ни одна программа и близко не подошла к прохождению теста. Хотя такие программы, как Элиза (ELIZA), иногда заставляли людей верить, что они говорят с человеком, как, например, в неформальном эксперименте, названном AOLiza, но эти случаи нельзя считать корректным прохождением теста Тьюринга по целому ряду причин:

- человек в таких беседах не имел никаких оснований считать, что он говорит с программой, в то время как в настоящем тесте Тьюринга человек активно пытается определить, с кем он беседует;

- документированные случаи обычно относятся к таким чатам, как IRC, где многие беседы отрывочны и бессмысленны;

- многие пользователи Интернета используют английский как второй или третий язык, так что бессмысленный ответ программы легко может быть списан на языковой барьер;

- многие просто ничего не знают об Элизе и ей подобных программах, и поэтому не сочтут собеседника программой даже в случае совершенно нечеловеческих ошибок, которые эти программы допускают.

В 1980 году в статье «Разум, мозг и программы» Джон Сёрль выдвинул аргумент против теста Тьюринга, известный как мысленный эксперимент «Китайская комната». Сёрль настаивал, что программы (такие как Элиза) смогли пройти тест Тьюринга, просто манипулируя символами, значения которых они не понимали. А без понимания их нельзя считать «разумными» в том же смысле, что и людей. «Таким образом, — заключает Сёрль, — тест Тьюринга не является доказательством того, что машина может думать, а это противоречит изначальному предположению Тьюринга».

Стюарт Рассел (Stuart Russel) и Питер Норвиг (Peter Norvig) утверждают, что антропоцентризм теста приводит к тому, что он не может быть по-настоящему полезным при разработке разумных машин. «Тесты по авиационному проектированию и строительству, — строят они аналогию, — не ставят целью своей отрасли «создание машин, которые летают точно так же, как летают голуби, что даже сами голуби принимают их за своих». Из-за этой непрактичности прохождение теста

Тьюринга не является целью ведущих научных или коммерческих исследований (по состоянию на 2009). Сегодняшние исследования в области искусственного интеллекта ставят перед собой более скромные и специфические цели.

**Д.Г. Григорьев (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

## **ПРОШИВКА И МОДЕРНИЗАЦИЯ BIOS**

BIOS (basic input/output system – «базовая система ввода-вывода») – реализованная в виде микропрограмм часть системного программного обеспечения, которая предназначена для предоставления операционной системе API доступа к аппаратуре компьютера и подключенным к нему устройствам. Производители материнских плат сами рекомендуют постоянно обновлять версию прошивки BIOS. Обычно в новых версиях BIOS исправляют многие ошибки, добавляют поддержку новых устройств и некоторые другие функции.

В персональных IBM PC-совместимых компьютерах, использующих микроархитектуру x86, BIOS представляет собой набор записанных в микросхему EEPROM (ПЗУ) персонального компьютера микропрограмм (образующих системное программное обеспечение), обеспечивающих начальную загрузку компьютера и последующий запуск операционной системы. BIOS содержит настройки для оборудования, их можно изменять при входе в меню настройки. Простому пользователю доступна лишь небольшая часть настроек. На самом деле их очень много. Все эти настройки хранятся в EEPROM-памяти CMOS. Именно они сбрасываются, когда вы вытаскиваете маленькую батарейку на материнской плате или переключаете соответствующую перемычку.

С 2000 года Intel трудится над новым типом интерфейса для связи между микропрограммами - EFI (Extensible Firmware Interface). С 2005-го разработку спецификации UEFI (Unified EFI) курирует организация Unified EFI Forum. На сегодняшний день в ее состав входят 11 ведущих компаний: AMD, American Megatrends Inc., Apple Computer Inc., Dell, Hewlett Packard, IBM, Insyde, Intel, Lenovo, Microsoft, Phoenix Technologies. UEFI представляет собой более быструю, стабильную, простую в управлении и к тому же более безопасную платформу. Результаты разработок постепенно внедрялись в производстве комплектующих для PC.

Интерфейс, определённый спецификацией EFI, включает таблицы данных, содержащие информацию о платформе, загрузочные и runtime-

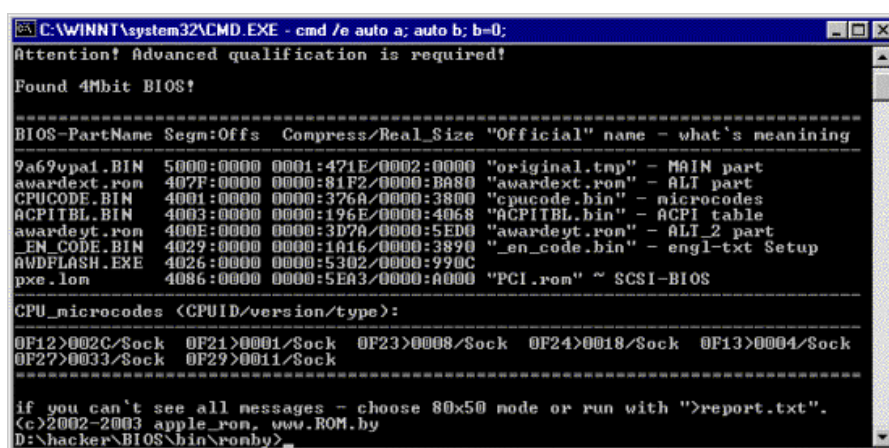
сервисы, которые доступны для загрузчика операционной системы (ОС) и самой ОС. Некоторые существующие расширения BIOS, типа ACPI и SMBIOS, также присутствуют в EFI, поскольку не требуют 16-разрядного runtime-интерфейса.

В UEFI можно интегрировать большое количество драйверов, не привязанных к определенной операционной системе. Это значит, что производителям достаточно написать всего одну версию драйвера для всех платформ. Благодаря инициализации драйверов на этом раннем этапе запуска можно получить доступ к сетевой плате, в том числе функциям загрузки по сети или удаленного обслуживания.

Программная начинка современных компонентов компьютера, которую пользователь «по старинке» называет BIOS, по факту уже является UEFI. Поэтому в дальнейшей части текста, во избежание двойного толкования, чаще будет применяться термин «Firmware».

Есть два способа прошивки Firmware для комплектующих вычислительной системы, это прошивка с помощью программы, и прошивка Firmware встроенной в устройство утилитой. В обоих случаях это сводится к файлообменной операции в рамках какого либо интерфейса. Хороший модуль программы замены прошивки содержит схему проверки соответствия предлагаемой версии Firmware составу оборудования и предотвратит некорректную перепрошивку устройства.

Чтобы модифицировать BIOS, необходимо знать его структуру для этого нужно распаковать файл его образа. В частности, обладатели версий от Award могут запустить cbrom с ключом /D (или br.exe с ключом /e), чтобы узнать – из каких блоков состоит данная версия Firmware и по каким адресам они располагаются (рисунок 1).



```
C:\WINNT\system32\CMD.EXE - cmd /e auto a; auto b; b=0;
Attention! Advanced qualification is required!
Found 4Mbit BIOS!
-----
BIOS-PartName Segn:Offs Compress/Real_Size "Official" name - what's meaning
-----
9a69vpa1.BIN 5000:0000 0001:471E/0002:0000 "original.tmp" - MAIN part
awardext.rom 407F:0000 0000:81F2/0000:BA80 "awardext.rom" - ALI part
CPUCODE.BIN 4001:0000 0000:376A/0000:3800 "cpucode.bin" - microcodes
ACPI_TBL.BIN 4003:0000 0000:196E/0000:4068 "ACPI_TBL.bin" - ACPI table
awardext.rom 400E:0000 0000:3D7A/0000:5ED0 "awardext.rom" - ALI_2 part
_EN_CODE.BIN 4029:0000 0000:1A16/0000:3890 "_en_code.bin" - engl-txt Setup
AWDFLASH.EXE 4026:0000 0000:5302/0000:990C
pxe.lom 4086:0000 0000:5EA3/0000:A000 "PCI.rom" ~ SCSI-BIOS
-----
CPU_microcodes (CPUID/version/type):
0F12>002C/Sock 0F21>0001/Sock 0F23>0008/Sock 0F24>0018/Sock 0F13>0004/Sock
0F27>0033/Sock 0F29>0011/Sock
-----
if you can't see all messages - choose 80x50 mode or run with ">report.txt".
(c)2002-2003 apple_rom, www.ROM.by
D:\hacker\BIOS\bin\romby>
```

Рисунок 1 – Результат работы утилиты BR.EXE

Первым идет **9a69vpa1.BIN** (original.tmp). Основной код Firmware сосредоточен именно здесь. Как видно, original.tmp загружается по

адресу 5000h:0000h и занимает 128Кб, впрочем надолго он в ней не задерживается и перед передачей управления загрузочному сектору он обязан освободить эти адреса.

Следом идет **awardext.rom** (ALT part) и **awardeyt.rom** (ALT\_2 part). Это расширение основного BIOS, которое инициализирует оборудование на финальной стадии загрузки (находит жесткие диски и оптические приводы, выводит таблицу PnP/PCI устройств и т. д.

**CPUCODE.BIN** – это просто набор микрокодов для всех поддерживаемых Firmware моделей процессоров. Микрокоды предназначены для исправления ошибок, допущенных при их разработке, однако для нормальной работы системы этот модуль, в общем-то, не критичен. Хотя, при желании можно скачать свежую версию микрокодов с сайта Intel или AMD и залить их в Firmware.

**ACRITBL.BIN** – еще один модуль данных. Здесь содержится таблица ACPI, т. е. данные по управлению всеми устройствами и автоматически распределяющая системные ресурсы. Модификация этой таблицы открывает большие возможности.

**\_EN\_CODE.BIN** – набор текстовых ASCIIZ-строк, используемых BIOS'ом. Вот где можно развернуться начинающим хакерам!

**AWDFLASH.EXE** – утилита для прошивки Firmware. Исполняемый файл для MS-DOS. Так что для написания универсального прошивальщика Firmware ничего, кроме ее самой не нужно.

**pxe.lom** – это набор драйверов для интегрированных устройств.

В современные Firmware могут входить и другие модули.

Технически процедура моддинга заключается в модификации или замене обнаруженных файлов, настройке закрытых опций и некоторых других операций. Программа, предлагающая удобный интерфейс для проведения этих операций представлена на рисунке 2.

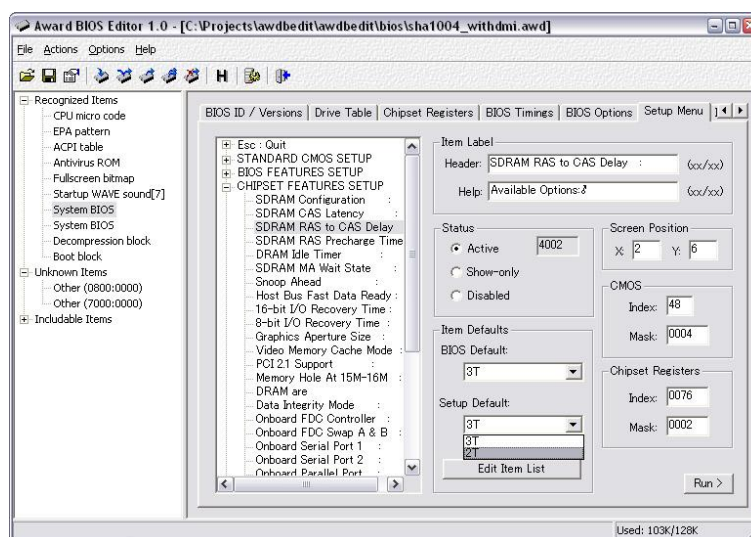


Рисунок 2 – Результат работы утилиты Award BIOS Editor

**Д.Г. Григорьев (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

## **ПРИМЕНЕНИЕ ТЕХНОЛОГИИ ORACLE В БЕЛАРУСИ**

Технология Oracle широко применяется практически во всех странах мира. Также она нашла применение и в Беларуси. Компания Oracle не только внедряет свои технологии в производство Беларуси, но также проводит конференции по поиску наилучшего пути решения многих производственных проблем.

Oracle представила в Беларуси глобальные стратегии и новости мировой IT-индустрии в сфере бизнес-приложений. Произошло это на деловом форуме по стратегиям управления и инновационным решениям Oracle AppsForum 2012. Мероприятие впервые прошло в Минске и собрало более ста пятидесяти зарегистрированных участников, представлявших ведущие предприятия Беларуси.

Бизнес-приложения Oracle полностью оптимизированы для программно-аппаратных комплексов Oracle Exadata Database Machine и Oracle Exalogic Elastic Cloud. Усиленная интеграция дает исключительную производительность для бизнес-приложений Oracle E-Business Suite, Oracle PeopleSoft, Oracle JD Edwards EnterpriseOne, Oracle Siebel CRM, Oracle ATG Commerce Suite и Oracle Supply Chain Management, что открывает новые возможности для предприятий, использующих Oracle на Oracle, такие как превосходная готовность, масштабируемость, энергоэффективность и низкая совокупная стоимость владения.

Стратегическое направление Oracle в области бизнес-приложений – это Oracle Fusion Applications. Новое поколение решений предлагает богатые функциональные возможности более 100 модулей из различных областей управления. Oracle Fusion Applications полностью интегрированы и обладают единой моделью данных и безопасности. Они разработаны так, что заказчик может выбирать модули и модели развертывания в соответствии со своими корпоративными требованиями и бизнес-процессами без ущерба для безопасности, производительности и удобства использования.

Также технологии Oracle широко применяются в различных отраслях промышленности, особенно в добывающей. В условиях рыночной экономики перед предприятиями добывающей промышленности, так же, как и перед предприятиями других отраслей, стоят задачи повышения эффективности деятельности, увеличения доходов, снижения расходов и повышения прибыли. Инновационно-технологические преобразования могут стать основой для обеспечения успешного будущего предприятий отрасли.



**М.И. Гриневич (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **П.Л. Чечет**, к.т.н., доцент

## **СОВРЕМЕННЫЕ ТЕХНОЛОГИИ РАЗРАБОТКИ WEB-САЙТОВ**

Глобальная компьютерная сеть на сегодняшний день содержит миллионы сайтов, на которых размещена всевозможная информация. Служба привлекательна тем, что позволяет организовывать на серверах сайты-хранилища текстовой и любой другой информации, которая может быть просмотрена пользователем в интерактивном режиме. Современные реалии таковы, что разработка сайта всё больше переходит из разряда сложных профессиональных задач в задачу, доступную любому владеющему основами компьютерных и web-технологий. Во многом в этом заслуга доступных современных инструментальных средств.

Интернет сайт (web-сайт) – набор интернет страниц (web-страниц) на которых посредством добавления вводится необходимая информация для отображения в сети интернет. Все Web-страницы на одном сайте подчинены одной тематике, и сайт представляет собой единое целое.

Создание сайта состоит из нескольких этапов:

- 1 дизайн сайта;
- 2 верстка сайта;
- 3 программирование сайта.

*Дизайн сайта* включает в себя цветовое оформление, наличие или отсутствие логотипа. Оформление должно быть тематическим, чтобы посетитель, попав на сайт, сразу понял, чем занимается данная компания. Для создания качественного дизайна может быть использована программа Adobe Photoshop – это многоцелевая программа, лидер в индустрии графики и дизайна, практически безграничные возможности, множество инструментов, функций и эффектов.

Программа обработки изображений Adobe Photoshop является безусловным лидером среди профессиональных графических редакторов за счет своих широчайших возможностей, высокой эффективности и скорости работы. Программа предоставляет все необходимые средства для коррекции, монтажа, подготовки изображений к печати и высококачественного вывода.

*Верстка сайта.* Под этим понятием веб-мастера подразумевают «натягивание» дизайна сайта (картинки) на скелет сайта. Это происходит с помощью языка HTML и таблицы стилей CSS. Верстка сайта – это «превращение» картинки (дизайна) в рабочую страничку сайта. Если сайт имеет страницы с разным дизайном, то верстка производится для

каждой из страниц, поскольку они будут иметь различную структуру, соответственно данное действие займет больше времени. Одним из важных вопросов в процессе верстки сайта является кроссбраузерность – свойство сайта отображаться и работать во всех популярных браузерах идентично. Под идентичностью понимается отсутствие развалов верстки и способность отображать материал с одинаковой степенью читабельности.

*Программирование сайта.* Одним из направлений, значительно упрощающих программирование сайта, является использование системы управления контентом – CMS (сокращение от Content Management System), которая сегодня имеет несколько видов (групп). От системы управления контентом зависит, какие возможности будут у сайта, как он будет работать, что получит рядовой посетитель сайта.

Все системы CMS можно разделить на следующие группы:

- «бесплатные»;
- «коробочные»;
- собственной разработки.

*«Коробочные» CMS.* К положительным сторонам «коробочных» CMS можно отнести универсальность. Однако закрытость кода таких платных продуктов становится главным минусом таких CMS, так как их код модифицировать можно только путем добавления специальных модулей.

*Бесплатные CMS.* Бесплатные «движки» имеют открытый код – и распространяются так называемой открытой лицензией. Пожалуй, это основное их отличие от «коробочных». Открытый код таких CMS позволяет расширить возможности для модификации и решения нестандартных задач.

*CMS собственной разработки.* Системы управления собственной разработки есть практически у каждой веб-студии, однако в последнее время их популярность снижается, так как код таких CMS зачастую закрытый, и из-за этого, выбрав такую систему, заказчик привязывает себя к конкретной студии.

При разработке модулей для сайта наиболее популярным является язык разработки сценариев PHP – язык программирования, используемый на стороне WEB-сервера для динамической генерации HTML-страниц. Об этом говорит и расшифровка его названия: PHP – Personal HyperText Processor. PHP – один из немногих языков программирования, созданных специально для разработки web-приложений. Он включает в себя все функции, необходимые именно для работы на web-сервере, и при этом лишен избыточности, свойственной многим его конкурентам.

Неотъемлемой частью современного сайта являются базы данных, одна из популярных и самых распространённых – MySQL. Она не предназначена для работы с большими объемами информации, но ее применение идеально для интернет сайтов, как небольших, так и достаточно крупных. MySQL отличаются хорошей скоростью работы, надежностью, гибкостью. Работа с ней, как правило, не вызывает больших трудностей. Поддержка сервера MySQL автоматически включается в поставку РНР.

Современные технологии значительно упростили задачу создания современного web-сайта, при этом эти же технологии позволяют качественно решать вопросы безопасности, масштабируемости и быстродействия. Всё это позволяет создавать сайты быстрее и качественнее.

**М.И. Гриневич (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **М.И. Жадан**, к.ф.-м.н., доцент

## **СОЗДАНИЕ ОТЧЁТОВ ДЛЯ ПРЕДПРИЯТИЙ ТОРГОВЛИ**

Целью автоматизации хозяйственной деятельности любой фирмы в конечном итоге является формирование разнообразной отчётности, удовлетворяющей интересам бухгалтерского, финансового и административного учета. Для решения этой задачи в платформе «1С:Предприятие» используется специальный механизм построения аналитической отчетности – система компоновки данных.

В работе описывается общее устройство системы компоновки данных, рассматривается последовательность и назначение каждого этапа компоновки. Также подробно описываются все составляющие схемы компоновки данных, структура отчета и его настроек.

Система компоновки данных – мощный и гибкий инструмент для создания отчетов на основе их декларативного описания. Использование декларативного описания отчетов позволяет реализовать следующие возможности:

- создание отчета без программирования,
- возможность создания различных вариантов отчета,
- возможность создания различных вариантов пользовательских настроек,
- разбиение исполнения отчетов на этапы,
- настройки структуры отчета,
- совмещение в отчете несколько таблиц,
- независимое использование отдельных частей системы компоновки данных,

– создание вложенных отчетов.

Формирование отчета с использованием системы компоновки данных разбивается на несколько этапов. При этом разбиение на этапы – не логическое, а физическое. То есть в каждый из этих этапов разработчик может вмешаться, программно или визуально, выполнить различные этапы компоновки данных в различных модулях и даже на различных компьютерах.

В предлагаемой работе на платформе «1С: Предприятие» были созданы различные виды отчетов, практически реализуемые на любом предприятии торговли. В настоящее время данная разработка находится на тестировании на предприятии «Родная сторона».

**А.С. Демиденко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

## **КОМПЬЮТЕРНАЯ ДИАГНОСТИКА АВТОМОБИЛЯ**

Проверить состояние двигателя и коробки передач автомобиля можно на компьютерной диагностике в автосервисе, что порой дорого для студента. Но iPhone, это тот же компьютер и с помощью специальной программы и кабеля (для разъема) диагностики вполне справляется с этой задачей. Кроме того, можно следить за работой двигателя в движении, что не сделают в сервисе.

Заглянув в App Store, нашел там еще три устройства, позволяющие осуществить мой замысел. Соединение устройства с компьютером автомобиля, выпущенных после 1996 года, происходит через диагностический разъем (OBDII), который обычно находится в салоне автомобиля, рядом с рулевой колонкой.

Варианты подключения диагностических устройств разные. Это и кабель, и wi-fi, и Bluetooth. Стоит сразу обратить внимание на то, что все программы для iPhone англоязычные, понять и применить которые не составит большого труда.

OBD2 от XTOOLTECH (WiFi). Устройство для диагностики двигателя с iPhone – IOBD2, который стоит \$79, обеспечивает соединение мобильного устройства с бортовым компьютером автомобиля через WiFi. Для диагностики служит бесплатное приложение iOBD2. Утилита имеет богатый пользовательский интерфейс и ее можно использовать для диагностики всех транспортных средств, совместимых с OBDII.

Внешне утилита очень напоминает DashCommand, только в отличие от известного приложения, предлагается бесплатно. Думаю, установив

эту программу, можно хорошо сэкономить. Как указано в спецификации, можно проводить замеры работы двигателя на холостом ходу и в движении. Снимать характеристики и статистику со всех доступных датчиков.

Программа для iPhone позволяет определить массу параметров, назову лишь несколько из них:

- количество оборотов двигателя,
- датчики топлива и двигателя (лямбда, температуры охлаждающей жидкости и воздуха, давление в коллекторе и топливной системе).

Все полученные данные можно визуализировать в виде графиков или цифр. По итогам тестирования устройство считывает коды неисправностей и обнуляет их. Мониторинг работы двигателя происходит в реальном времени или на основе данных из КЭШа (рисунок 1).

Спецификации:

- совместимость с устройствами на iOS и Android
- 2 режима тестирования (спортивный и круиз)
- WiFi-соединение
- iOBD2 (Бесплатно)



Рисунок 1 – Меню программы

Bluetooth-передатчик BlueDriver. Еще один диагностический инструмент для тестирования автомобиля, который продается на официальном сайте производителя за \$99. Работает только с утилитой BlueDriver от Lemur Vehicle Monitors и устройствами Apple.

Самый простой, из описанных выше устройств. Но с другой стороны, не перегружен лишней информацией.

BlueDriver может регистрировать различные коды неисправностей для их дальнейшего исправления. А также предупреждать вас о превышении скорости (сомнительная польза). И вот такая опция еще есть: импорт результатов сканирования системы автомобиля на стену Facebook и возможность отправлять твиты BlueDriver (Бесплатно) (рисунок 2).



Рисунок 2 – Меню устройства

При глубоком изучении данной темы, и огромном энтузиазме, возможно, создать такое приложение полностью с нуля.

Востребованность в данной услуге на текущий момент возрастает. Детальная и проверка параметров работы автоэлектроники внешними средствами является первым этапом для выбора действий по профилактике сбоев в работе автотранспортных средств.

Д.В. Деревянко, Е.С. Абрамов (УО «ГГУ им. Ф. Скорины», Гомель)  
Науч. рук. А.И. Кучеров, старший преподаватель

## АВТОМАТИЗАЦИЯ ОПРЕДЕЛЕНИЯ ПОГРЕШНОСТИ ИЗМЕРИТЕЛЕЙ ЛИНЕЙНЫХ РАЗМЕРОВ

Разработанная программа загружает готовую фотографию измерителя линейных размеров и с помощью оператора компьютера определять погрешность относительно эталонного измерителя линейных размеров.

При реализации данной программы была положена следующая идея. Пусть имеется линейный эталонный измеритель, расположим его на ровной плоскости. Рядом с ним располагаются тестируемый линейный измеритель, например рулетка. Над всей длиной данных измерителей располагается рейка, на которой закреплена веб-камера, которая в свою очередь может по ней двигаться. На рейке имеются контрольные точки, попадая на которые веб-камера делает снимки тестируемого экземпляра и отправляет полученные снимки на компьютер оператора.

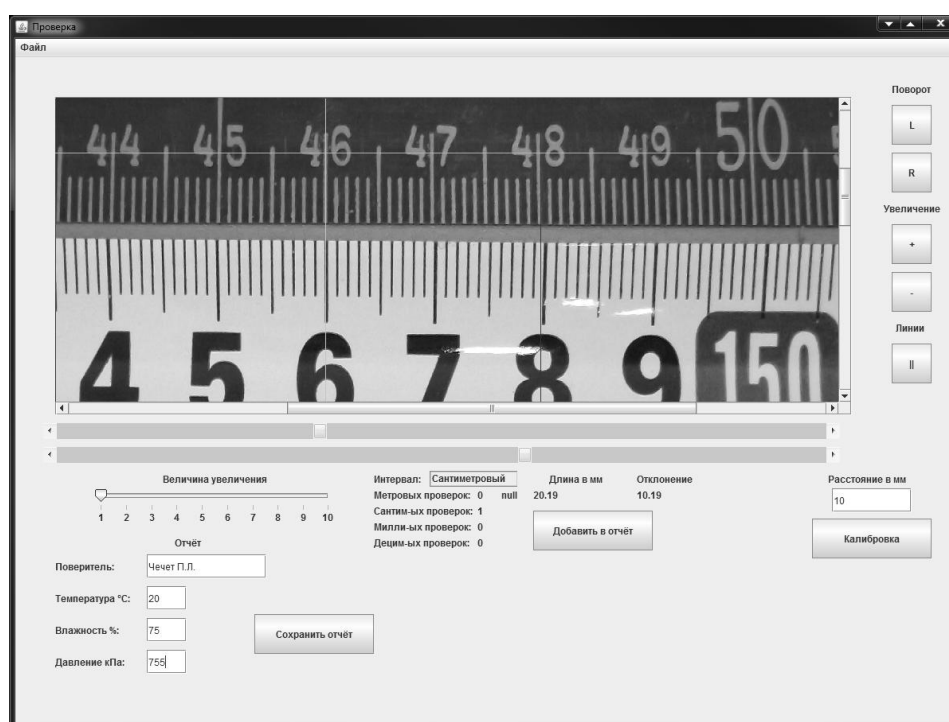


Рисунок 1 – Вид программы

В программе есть два ползунка, которые оператор может перемещать. Так как имеется эталон, при загрузке изображения оператор должен сделать калибровку на эталоне. При калибровке ползунки устанавливаются на эталоне и оператор вводит заведомо известное расстояние

(например, первый ползунок устанавливается на значении 46 см, а второй – на 47 см и в поле указывается расстояние 10 мм). Таким образом, осуществляется привязка количества пикселей к расстоянию в миллиметрах. После калибровки можно проверять тестируемый измеритель, устанавливая ползунки на разные расстояния и получая погрешность. Изображение можно увеличивать и при необходимости вращать, в случае если эталон не перпендикулярен ползункам. В данной программе возможны тесты на несколько диапазонов расстояний: миллиметр, сантиметр, дециметр, метр.

Помимо измерения погрешностей в программе заполняются следующие данные: оператор, температура, влажность, давление, номер протокола, номер тестируемого измерителя, заказчик, дата проведения проверки. При окончании измерения данные отправляются в MS Excel – отчёт, который создаётся автоматически.

Данная автоматизация определения погрешности приборов может быть внедрена на предприятии по стандартизации измерительных приборов, в частности по стандартизации измерителей линейных размеров в совокупности с соответствующим аппаратным обеспечением.

### **Литература**

1. Гери, Д., Хорстман, К. JavaServer Faces. Библиотека профессионала, 3-е изд.: Пер. с англ. – М. : Издательский дом «Вильямс», 2011. – 544 с.

2. Эккель, Б. Философия Java. Библиотека программиста. 4-е изд. – М.: СПб: Питер, 2012. – 640 с.

**Р.В. Дешкович (УО «ГГТУ им. П.О. Сухого», Гомель)**

Науч. рук. **Кравченко А. И.**, к.ф.-м.н., доцент

### **РАЗРАБОТКА WEB-РЕСУРСА РЕШЕНИЯ ЗАДАЧ ПО ФИЗИКЕ ДЛЯ СТУДЕНТОВ**

Сегодня, когда технологический прогресс шагнул далеко, многие ученые ищут способы автоматизации различных научных процессов. Существует множество вычислительной техники самого разного уровня, которая может выполнять различные задачи, начиная от систематизации данных, полученных в ходе исследований, и заканчивая контролем различных научных процессов.

Уже созданы такие машины, как «Watson 2.0», которая оснащена искусственным интеллектом высокого уровня, позволяющая анализировать



большие объемы данных за считанные секунды. Сейчас «Watson 2.0» используется сотрудниками медицины. Он помогает диагностировать, а затем выдавать варианты лечения для пациентов со сложной формой рака. Сейчас ведутся работы по переносу Watson 2.0 на персональные компьютеры и мобильные устройства. Но на сегодняшний день, это невозможно, так как это достаточно ресурсоемкая программа, которая расположена на 20 серверах IBM.

В настоящей работе представлена разработка интернет-ресурса, получившего название Fis-assist, что означает «Ассистент в физике», предназначенного для решения типовых задач по физике, рассматриваемых в курсе общей физики для студентов технических специальностей вузов.

Fis-assist должен обладать свойствами географической доступности (пользователь должен иметь доступ к приложению в любой точке мира), пользовательской доступности (приложение должно быть доступно для пользователей с любыми навыками работы на персональном компьютере). Кроме того, приложение должно иметь динамическое развитие. Оно должно постоянно модернизироваться и при обнаружении неточностей должны приниматься своевременные меры по их устранению. Для реализации этих свойств предусмотрено ведение информационной базы и набор алгоритмов построения решения задач различной сложности. Алгоритмы разбиты на классы, соответствующие типам задач.

Разработан интерфейс приложения, предлагающий пользователю выбрать раздел физики, к которому относится решаемая задача и ввести условие задачи, предварительно ознакомившись с правилами формализации условия или пользуясь подсказками, доступными во время ввода. После ввода, условие передается обработчику, где оно обрабатывается, а именно разбирается регулярными выражениями, на наличие вопроса и полученных данных. После успешного разбора условия в работу вступает алгоритм, строящий решение. Каждый алгоритм может решать определенный тип задач по физике из класса, к которому он относится.

Дизайн Fis-assist построен по стандарту Web 2.0, а разметка на блоках <div>, которые позволяют в любой момент внести необходимые корректировки, или же добавить новые элементы. Внутренняя работа интернет ресурса строится с использованием языка программирования PHP 5. В качестве хранилища данных выбрана реляционная СУБД MySQL, поддерживаемая большинством хостингов. А так рассматриваемое приложение – это интернет ресурс, то на начальном этапе развития будет размещаться на хостинге. В качестве интерфейса для работы с MySQL был выбран привычный интерфейс PhpMyAdmin. Так как именно он обычно установлен в панели управления хостингом.

**Н.Н. Диваков (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **В.Н. Кулинченко**, старший преподаватель

## **СОЗДАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УЧЕТА ПЕДАГОГИЧЕСКОГО И УЧЕНИЧЕСКОГО СОСТАВА ШКОЛЫ «ГИМНАЗИЯ № 56»**

Активная деятельность по отысканию приемлемых способов обобщения непрерывно растущего объема информации привела к созданию в начале 60-х годов специальных программных комплексов, называемых «Системы управления базами данных» (СУБД). Основная особенность СУБД – это наличие процедур для ввода и хранения не только самих данных, но и описаний их структуры. Файлы, снабженные описанием хранимых в них данных и находящиеся под управлением СУБД, стали называть банки данных, а затем «Базы данных» (БД).

Целью курсового проекта является разработка автоматизированной подсистемы учета педагогического и ученического состава. Создание данной базы данных необходимо для автоматизации учета льготников как одного из процессов учета в ЖЭУ. Разработка подсистемы даст возможность обрабатывать большие объемы информации, предоставит удобный интерфейс для работы с данными, позволит пользователю как просматривать необходимую информацию, так и, по мере необходимости, манипулировать ею, например, добавлять, изменять, копировать, удалять, сортировать и т. д., что значительно упростит процесс учета.

Поставленная задача была полностью реализована с помощью СУБД MS Access. Интерфейс пользователя реализован в виде удобной графической формы с интуитивно понятным интерфейсом и исчерпывающей справкой. Были выполнены все поставленные задачи. Полученная база данных легка и понятна в использовании, дает всю необходимую пользователю информацию, а также обеспечивает быстрый и легкий доступ к данным. Программа обеспечивает возможности по пополнению, редактированию, просмотру и анализу базы данных. Пользователь имеет возможность осуществлять поиск и отбор данных в таблице по заданным значениям тех или иных полей, поиск и отбор данных в связанной таблице по заданным значениям тех или иных полей основной таблицы (и наоборот). Также для пользователя была создана справка, обратившись к которой он может получить информацию о работе с конвертором.

По мере необходимости не составит труда добавить в программу различные запросы и отчеты, которые могут понадобиться пользователю. То есть приложение может постоянно совершенствоваться, что еще более увеличивает область его применения.

Разработка данной программы была целесообразна, так как она значительно упрощает работу по учету педагогического и ученического состава школы «Гимназия № 56». Разработанная подсистема дает возможность обрабатывать большие объемы информации, предоставляет удобный интерфейс для работы с данными, позволит пользователю, как просматривать необходимую информацию, так и манипулировать ею, что значительно упрощает процесс учета. Созданная программа является простой в использовании, удобной и практичной.

**О.В. Дробышевская (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **В.Н. Леванцов**, старший преподаватель

## **АЛГОРИТМ СОРТИРОВКИ МАССИВОВ QUICKSORT**

Быстрая сортировка, часто называемая qsort, – это широко известный алгоритм сортировки, который в 1960 году был разработан английским информатиком Чарльзом Хоаром. Qsort является одним из самых быстрых известных универсальных алгоритмов сортировки массивов, однако не лишен некоторых недостатков.

В этой сортировке используется стратегия «Разделяй и властвуй». Алгоритм будет выполняться в соответствии с планом, описанным далее.

В массиве выбирается так называемый опорный элемент. С точки зрения корректности алгоритма неважно, какой именно элемент будет выбран опорным. Известные стратегии: либо постоянно выбирать один и тот же элемент, например, средний или последний по положению, либо выбирать элемент со случайно выбранным индексом. Далее следует операция деления массива: реорганизуем массив так, чтобы слева опорного элемента оказались все элементы со значениями равными или меньше его, а справа – все элементы со значениями выше опорного. После этого рекурсивно упорядочиваются все подмассивы, находящиеся слева и справа от опорного элемента. Базой рекурсии являются наборы, состоящие из одного или двух элементов. Первый элемент возвращается в исходном виде, во втором, при надобности, сортировка сводится к перестановке двух элементов. В процессе деления все такие отрезки уже упорядочены.

Поскольку на каждом следующем уровне рекурсии (итерации) длина обрабатываемого отрезка массива уменьшается по меньшей мере на единицу, терминальная ветвь рекурсии будет достигнута обязательно и обработка гарантированно завершится.

QuickSort – это существенно улучшенный вариант алгоритма сортировки с помощью прямого обмена (известные варианты «Пузырьковая сортировка» и «Шейкерная сортировка»), знаменитого своей низкой эффективностью. Главное его отличие состоит в делении элементов после каждого прохода на независимые группы. Интересно, что после улучшения самого неэффективного алгоритма сортировки в результате был получен эффективный улучшенный метод.

В лучшем случае, во всех итерациях каждый из подмассивов делится на два одинаковых по величине массива, в результате чего число сравнений, совершаемых быстрой сортировкой, равно значению рекурсивного выражения  $C_N = 2C_{N/2} + N$ , что в явном выражении дает примерно  $N \lg N$  сравнений, из чего следует наименьшее время сортировки.

Достоинства этого метода заключаются в быстродействии, простоте реализации, хорошем сочетании с механизмами кэширования и виртуальной памяти, работе на связанных списках и других структурах с последовательным доступом.

Недостатки qsort: большая деградация по скорости при неудачных выборах опорных элементов, что возможно при неудачных входных данных. Этого можно избежать, используя такие модификации алгоритма, как Introsort или вероятностно, выбирая опорный элемент случайно, а не фиксированным образом. Также наивная реализация алгоритма может привести к переполнению стека.

**А.Г. Зайченко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Л.А. Цурганова**, к.т.н., доцент

## **ИСПОЛЬЗОВАНИЕ ВНЕШНЕЙ ПАМЯТИ ДЛЯ КОНЕЧНО-ЭЛЕМЕНТНОГО МОДЕЛИРОВАНИЯ СИСТЕМЫ «ПЛИТА – ГРУНТОВОЕ ОСНОВАНИЕ»**

При моделировании системы «Плита – грунтовое основание» методом конечных элементов необходимо использовать внешнюю память с целью расширения размера дискретной нерегулярной решетки расчетной области. Конечно - элементное моделирование состоит из следующих этапов: построение матрицы жесткости, построение и учет граничных условий, решение системы линейных уравнений. Для использования внешней памяти в настоящей работе разработаны и программно реализованы алгоритмы трех этапов конечно-элементного моделирования.

При построении матрицы жесткости производится расчет  $2kx$  строк матрицы жесткости перебором связей между узлами конечного элемента,

где  $kx$  – количество разбиений решетки по оси  $Ox$ . Далее записываются во внешнюю память первые  $kx$  строк. Затем делается сдвиг матрицы вверх на  $kx$  строк. После сдвига строк матрицы, вычисляются следующие  $2kx$  строк и записываются очередные  $kx$  строк в файл. Это продолжается до тех пор, пока не будет сформирована вся матрица.

Построение граничных условий связано с постановкой задачи. Внешняя память в этом этапе не используется.

Учет граничных условий приводит к изменению элементов строк и столбцов матрицы жесткости, связанных с узлами, в которых известны перемещения. Такая процедура включает в себя обнуление элементов соответствующих столбцов и строк и присвоение диагональным элементам единиц. При учете граничных условий сначала считываются  $2kx$  строк матрицы жесткости из файла, затем производится их изменение и запись первых  $kx$  строк в файл, далее делается сдвиг на  $kx$  строк вверх и считываются очередные  $kx$  строк из файла. Это продолжается до тех пор, пока не будет преобразована вся матрица.

После учета граничных условий решается система линейных алгебраических уравнений модифицированным методом Гаусса со считыванием по частям матрицы жесткости из файла.

### Литература

1. Быховцев, В.Е. Компьютерное моделирование систем нелинейной механики грунтов / В.Е. Быховцев, А.В. Быховцев, В.В. Бондарева. – Гомель: УО «ГГУ им.Ф. Скорины», 2002. – 215 с.

**Г.А. Зубов (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **М.И. Жадан**, к.ф.-м.н., доцент

## **ИСПОЛЬЗОВАНИЕ ФИЗИЧЕСКИХ ЗАКОНОВ ДЛЯ ОПИСАНИЯ ВЗАИМОДЕЙСТВИЙ ТРЕХМЕРНЫХ ОБЪЕКТОВ**

В современной промышленности широко используются различные виды моделирования. В первую очередь это моделирование различных производственных объектов. Часто также важно создать модель, составляющие которой будут изменять свои динамические характеристики, взаимодействовать с другими объектами.

Для моделирования различных объектов удобно использовать графические приложения. Они предоставляют большое количество готовых к использованию шаблонов, а также возможности их настройки.

Однако не всегда такие программные пакеты предлагают описать физические взаимодействия между графическими объектами.

В ходе работы был сконструирован набор программных библиотек, позволяющих создавать и отображать графические трехмерные объекты. Также, на основе пакета Farseer Physics Engine, разработана модель взаимодействия графических объектов – соударения, изменения динамических характеристик, различные соединения нескольких объектов в один, гравитационные взаимодействия и так далее.

Разработанная библиотека позволяет конструировать различные графические модели, используя готовые шаблоны, с возможностью подключения физических взаимодействий.

.NET Framework — программная платформа, выпущенная компанией Microsoft в 2002 году. Фактически представляет собой операционную систему внутри операционной системы.

Microsoft XNA— набор инструментов с управляемой средой времени выполнения (.NET), созданный Microsoft, облегчающий разработку и управление компьютерными играми. XNA стремится освободить разработку игр от написания «повторяющегося шаблонного кода» и объединить различные аспекты разработки игр в одной системе.

### **Литература**

1. Samuel R. Buss. 3D Computer Graphics, Cambridge / Samuel R. Buss / Cambridge. – 2003. – С. 304–350.
2. S. Cawood Microsoft XNA Game Studio Creators Guide: An Introduction to XNA Game Programming / Cawood S. / McGraw-Hill Osborne Media, 2007. – 322 с.

**И.В. Киселев (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Н.А. Шаповалова**, старший преподаватель

### **ОСНОВНЫЕ ЭТАПЫ СОЗДАНИЯ WEB-САЙТА**

Web-сайт – это объединённая под одним адресом (доменным именем) совокупность документов частного лица или организации. По умолчанию подразумевается, что сайт располагается в сети Интернет, но он так же может публиковаться в локальной сети компании – в сети Интернет.

Глобальная сеть состоит из всех сайтов, доступных для общего пользования. Все страницы каждого сайта объединяются одним корневым адресом, тематикой, системой и дизайном. Каждая страница Web-сайта это текстовый документ, который написан на языке программирования (HTML, XHTML, JAVA и др.).

Основная функция web-сайта – это передача информации. Лучшему восприятию сведений способствует графическое оформление и удобная компоновка материала. Хороший дизайн веб-сайта – это не просто удачное сочетание цветов и грамотно подобранные картинки. Это способ наглядного представления информации. Это удобная навигация, приятные и простые в работе меню, доступность популярных разделов и быстрый поиск остальной информации.

Классификация web-сайтов.

По доступности сервисов сайты подразделяют на:

- открытые – все сервисы полностью доступны для любых посетителей;
- полуоткрытые – для доступа необходимо зарегистрироваться (обычно бесплатно);
- закрытые – полностью закрытые служебные сайты организаций (в том числе корпоративные сайты), личные сайты частных лиц. Такие сайты доступны для узкого круга людей. Доступ новым людям обычно даётся через так называемые инвайты (приглашения).

По природе содержимого сайты подразделяют на:

- статические – всё содержимое заранее подготавливается. Пользователю выдаются файлы в том виде, в котором они хранятся на сервере;
- динамические – содержимое генерируется специальными скриптами (программами) на основе других данных из любого источника.

По физическому расположению сайты подразделяют на:

- внешние сайты сети Интернет;
- локальные сайты — доступные только в пределах локальной сети. Это могут быть как корпоративные сайты организаций, так и сайты частных лиц в локальной сети провайдера.

На первый взгляд создание сайтов не вызывает особых вопросов. Нарисовал дизайн покрасивее, провел его программирование, наполнил ресурс контентом, и сайт готов. Тем более, что в интернете существует масса всяких конструкторов, модулей, компонентов и т. д.

Разработка сайта самостоятельно под силу даже программисту начального уровня. Другой вопрос – это качество подобного web-ресурса. Для тех же, кто далек от азов программирования, создание даже простого сайта превратится в длительный и непонятный процесс. Поиски ответов на элементарные вопросы, которые будут возникать по ходу разработки сайта, заберут уйму времени и сил. На выходе же будет посредственный ресурс.

Опишем семь основных этапов создания любого web-проекта.

1 Сайты создаются для разных целей и для достижения этих целей существуют разные способы. Совместно определив, что вам необходимо,

мы сможем предложить варианты достижения ваших целей. На этом этапе создается структура будущего сайта и проектируется бизнес-логика.

В рамках обсуждения проекта выяснятся, за какой срок можно реализовать решение задачи и какой бюджет потребуется. В результате выполнения первого этапа будет получено описание структуры сайта, схема навигации по сайту и технические требования к сайту.

Обычно первый этап требует 8-15 дней на выяснение необходимых деталей и получение материалов, необходимых для постановки задачи на разработку сайта и написание полноценного технического задания .

2 Выбор доменного имени. Домен – это символическое обозначение web-ресурса. Важно чтобы это обозначение было легко для запоминания и прочтения. Оно должно вызывать ассоциации с вами или вашей кампанией.

3 Разработка дизайна и программной платформы. Это работа для профессионалов. Дизайнер «рисует» сайт, а программист создает на основе этого «рисунка» работающий ресурс. Перед тем, как приступить к разработке дизайна сайта, заказчик рассказывает о назначении проекта и аудитории, для которой он предназначен. Обсуждается список сервисов, рисуется схема сайта. Создание качественного дизайна веб-сайта возможно только при тесном сотрудничестве и полном взаимопонимании заказчика и исполнителя. Терпеливая и слаженная работа представителей обеих сторон позволяет разработать удачную концепцию проекта и грамотно воплотить задумку в жизнь с учетом всех пожеланий и требований клиента.

4 Наполнение ресурса контентом. Контент – эта вся информация, текстовая и графическая, которую вы хотите предоставить своей аудитории. На этом этапе трудится верстальщик. Наполнение сайта контентом — это постоянная работа, которая не заканчивается после запуска ресурса и его успешной раскрутки. Для того чтобы сохранить завоеванные позиции, необходимо постоянно поддерживать актуальность контента. Статьи для сайта могут быть самого разного объема и направленности. Вовсе не обязательно заполнять сайт громоздкими текстами, если того не требует концепция вашего ресурса. Небольшие статьи, начиная от 500–700 символов, вполне могут составлять основу контента. Однако и небольшие тексты непременно должны отвечать всем требованиям качества и уникальности. Тексты, используемые для наполнения сайта, должны быть абсолютно грамотными и лёгкими для восприятия.

5 Размещение в сети. Проект нужно разместить на качественном хостинге, где пользователи будут иметь к нему бесперебойный доступ.

Для получения результата от предыдущих пяти пунктов необходимо выполнить еще два.



6 Поисковая оптимизация сайта. Для поиска информации в сети пользователь использует поисковые запросы. Хотите чтобы он попал к вам на сайт – ресурс должен находиться в верхних строчках поисковых результатов. Опытный веб-мастер, используя поисковую оптимизацию, продвинет ваш проект в топ.

7 Раскрутка. Именно благодаря раскрутке растет число посетителей. Здесь также не обойтись без услуг профессионалов. Показательно, что продвижение web-сайта в поисковых системах в несколько раз эффективней размещения сетевой рекламы: раскрутка проекта при помощи баннеров и рекламных ссылок не позволяет сайту занять первые позиции в поисковиках и привлечь аудиторию, которая ищет конкретные товары или услуги. Продвижение в поисковых системах – грамотная оптимизация web-страниц: данная работа предполагает создание семантического ядра сайта. Формирование ядра осуществляется путем подбора тематических ключевых слов и словосочетаний. Составление списка ключей происходит на основании статистики реальных поисковых запросов.

Даже такое упрощенное описание этапов создания сайта говорит о том, что процесс этот не так прост, как кажется.

**Н.В. Кисель (УО БрГТУ, Брест)**

Науч. рук. **В.И. Хвещук**, к.т.н., доцент

## **СРЕДСТВА АВТОМАТИЗАЦИИ МОДЕЛИРОВАНИЯ И ОЦЕНКИ МОДЕЛЕЙ АВТОМАТИЗИРОВАННЫХ СИСТЕМ**

Концепция автоматизированной системы (АС) – это набор согласованных системных решений по структуре системы, включающей программные, информационные и технические средства. Процесс разработки и оценки концепции АС представляет собой итеративную и трудоемкую процедуру выбора и оценки различных вариантов решений по структуре и элементам АС. В работе предложен подход к автоматизации процесса построения и оценки концепции АС путем разработки соответствующего комплекса программных средств.

В качестве архитектуры АС используется клиент-серверная структура для систем обработки информации, основными компонентами которой являются рабочие места пользователей, рабочие станции (пользовательские и серверные), кабельная система, программные и информационные элементы АС.

Предложенный комплекс построения и оценки концепций АС включает набор следующих групп функциональных средств для описания:

1 Информационное обеспечение АС. Описываются следующие

информационные элементы: файлы архивных и текущих документов, базы данных (локальные общие), архивы. Задается размещение информационных элементов по рабочим станциям АС.

2 Программное обеспечение АС. Описываются системные, инструментальные и прикладные программы. Для прикладных программ указываются взаимосвязи с информационными элементами (базами данных), а также возможные связи между прикладными программами. Задается размещение программных элементов по рабочим местам и станциям АС.

3 Техническое обеспечение АС. Описываются рабочие станции АС: перечень устройств и ПЭВМ.

4 Рабочие места АС. Для каждого рабочего места определяются используемые информационные, программные и технические элементы АС.

5 Размещение рабочих станций АС. Задается перечень и метраж помещений ОА. Приводится расположение рабочих станций и рабочих мест пользователей по помещениям ОА.

Для перечисленных выше элементов АС приводятся оценки их стоимости (приобретения или экспертные оценки реализации), на основе которых определяется общая стоимость АС и ее отдельных компонентов.

Предложенные средства автоматизации построения и оценки концепции АС реализуется в виде набора приложений в клиент-серверной архитектуре, обеспечивающего диалоговый интерфейс, как для студентов, так и для преподавателей.

Результатом работы применения комплекса является набор концепций АС и их оценки, на основе которых можно проводить выбор концепции наиболее полно удовлетворяющей требованиям заказчика. Полученные результаты можно использовать для разработки технических заданий на создание АС.

Планируется использование предложенного комплекса в рамках лабораторных работ и курсового проектирования по дисциплине «Проектирование автоматизированных систем обработки данных».

**Д.А. Кисло (УО БрГТУ, Брест)**

Науч. рук. **В.И. Хвещук**, к.т.н., доцент

## **СРЕДСТВА АВТОМАТИЗАЦИИ ПОСТРОЕНИЯ МОДЕЛЕЙ ПРЕДПРИЯТИЙ**

Одной из важных задач в процессе построения автоматизированных систем обработки данных (СОД) является обследование и

документирование объекта автоматизации (ОА), а также последующее построение и исследование его модели. В целях совершенствования процесса изучения ОА студентами в рамках дисциплины «Проектирование автоматизированных систем» предложен модельный подход, который заключается в разработке и применении конструктора моделей объектов, обеспечивающего использование этих результатов в учебном процессе при создании СОД. ОА является деятельностью предприятий, их отдельных подразделений и/или сотрудников, которая связана с обработкой документов (прием, хранение, обработка, тиражирование и т. д.).

Предложенные в работе средства автоматизации построения модели (САПМ) ОА представляются виде набора конструкций, которые используются для построения четырех взаимосвязанных моделей:

1 Организационная модель ОА представляет в виде иерархии подразделений. Отдельное подразделение состоит из групп сотрудников. Каждая группа включает определенное количество сотрудников. Определяется размещение сотрудников по помещениям ОА.

2 Информационная модель ОА описывается в виде совокупности документов и информационных связей между ними. Все документы разделены на справочные, оперативные и итоговые. Для каждого документа определяется макет, словарь показателей и объем документа. Кроме этого, документы делятся на текущие и архивные документы, для которых приводятся оценки их количества.

3 Функциональная модель ОА определяется в виде совокупности взаимосвязанных процессов (комплексов задач). Процесс состоит из отдельных задач, между которыми существуют определенные связи (по управлению и по передаче данных). Определяется каталог задач ОА. Для каждой задачи описываются входные и выходные документы, выполняемые операции над документами, а также приводятся оценки интенсивности использования и трудоемкости решения. Отдельную задачу или ее часть может выполнять отдельный сотрудник. Для каждого пользователя определяется функциональная модель или перечень решаемых задач или функций по обработке документов.

4 Модель наследуемых средств автоматизации ОА представляется в виде описания используемых технических, программных и информационных средств автоматизации.

САПМ реализуются в виде диалогового средства ориентированного на студентов и преподавателей. Студенты используют САПМ для построения моделей ОА, а преподаватели – для контроля результатов.

Предложенный комплекс САПМ представляется в виде набора приложений в клиент-серверной архитектуре. Для документирования результатов моделирования разработан набор отчетов.

Результаты применения данного комплекса можно использовать при разработке концепции АС. Планируется использование предложенного комплекса в рамках лабораторных работ и курсового проектирования по дисциплине «Проектирование автоматизированных систем».

**М.А. Климович (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **П.Л. Чечет**, к.т.н., доцент

## **ОПТИМИЗАЦИЯ ПРИЛОЖЕНИЙ В UNITY3D**

Unity – это мультиплатформенный инструмент для разработки двух- и трёхмерных приложений и игр. Созданные с помощью Unity приложения работают под многими операционными системами, а также на игровых приставках. Есть также возможность создавать интернет приложения с помощью специального подключаемого модуля к браузеру Unity, а также с помощью экспериментальной реализации в рамках модуля AdobeFlashPlayer. Приложения, созданные с помощью Unity, поддерживают DirectX и OpenGL.

Разрабатывая игру на Unity, стоит учитывать, что, возможно, приложение будет функционировать и на мобильной платформе, для которой все еще актуален вопрос производительности.

В Unity для контроля производительности служит панель *Stats*, которую можно вывести при проигрывании игры. В ней, помимо общего числа треугольников, следует обратить внимание на значение *DrawCalls*. Чем меньше это значение, тем лучше. Оно отображает, сколько раз центральный процессор (CPU) отправил данные в графический процессор (GPU). Именно постоянная пересылка данных между CPU и GPU является тонким местом в любом графическом приложении.

Основные рекомендации по снижению значения *DrawCalls*.

1 Использовать как можно меньше текстур и материалов. Желательно, чтобы как можно больше объектов использовало один и тот же материал.

2 Объединять статические объекты, использующие один и тот же материал, в один общий объект *Mesh*. Для этого все объекты необходимо разместить в одном родительском объекте-контейнере и применить к нему скрипт из стандартного набора компонентов Unity, который находится в меню Component ->Mesh ->CombineChildren.

Так же можно использовать следующие, общие, приёмы для повышения производительности приложения. Для области всего свободного

пространства, в котором строится структура игры или приложения, можно использовать:

1 Использовать только один источник света, для изменения освещения отдельных частей карты использовать объекты *Lightmap*.

2 Как можно реже, или вообще не использовать объекты с компонентом *Rigidbody*, который предназначен для добавления физики реального тела.

3 Использование компонента *Fog* (туман) сильно снижает производительность.

4 Шейдеры для материалов желательно использовать из категории *Mobile*, которые более оптимизированы для мобильных платформ.

Для оптимизации действий, закреплённых на объектах или же скриптах, нужно использовать следующее:

1) физику и расчеты вести в *FixedUpdate()*;

2) не использовать оператор *new* (типа *new Vector3(x,y,z)*, *new Rect()*) в циклах и часто вызываемых местах;

3) функция *CompareTag* более чем в два раза быстрее прямой проверки тэга по имени;

4) вызов *thisTransform.position – target.position).sqrMagnitude* выполняется быстрее, чем *Vector3.Distance*;

5) использовать как можно меньше вызовов *thisTransform.forward* или *position*, лучше эти значения один раз получить в функции *Update*, и использовать во всем скрипте;

6) для включения скрипта на объекте лучше использовать не *GetComponent(...).enabled* с заранее отключенным скриптом, а *AddComponent(...)* при заранее не установленном скрипте на объекте;

7) функции типа *OnTriggerEnter()*, *OnTriggerStay()* сильно снижают производительность, вместо триггеров лучше использовать *Vector3.Distance* между объектами.

Террейн (англ. *Terrain*) – это ограниченное геологическое тело региональной протяженности, которое характеризуется своими структурными особенностями. В Unity используется для создания фауны, ландшафтов, игрового пространства.

При использовании Террейна в целях оптимизации производительности можно руководствоваться следующим:

1) использовать небольшие размеры террейна, до ~150x150 ед.;

2) *PixelError* установить с 0 до максимально приемлемого значения, что уменьшит видимую дистанцию для игрока;

3) отключать флажок *Draw* автоматической отрисовки;

4) не использовать отрисовку травы в инструментах *Terrain*'а, это сильно снижает производительность.

Приведённые выше рекомендации позволяют создавать приложения в Unity 3D, обладающие достаточным быстродействием для комфортной работы с ними на большинстве современных мобильных устройств.

**А.В. Климович (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Н. Кулинченко**, старший преподаватель

## **РАЗРАБОТКА ПРОЕКТА ДИАГНОСТИЧЕСКОГО СТЕНДА ДЛЯ РЕМОНТА БОРТОВЫХ КОМПЬЮТЕРОВ ПРЕДПРИЯТИЯ «ЦВЕТОТРОН»**

В настоящее время на предприятии, где производятся бортовые компьютеры, для тестирования исправных и диагностики неисправных блоков используется специализированный диагностический стенд. Он предназначен для проверки блоков бортовых компьютеров марки «БКИ-01», «БКИ-03», «БИУС-01», «БИУС-02», «БИУС-03».

Диагностический стенд, используемый на предприятии, представляет собой коммутационное устройство, позволяющее подавать многочисленные сигналы на вход бортового компьютера, на дисплее которого отображается вся информация о режимах и состоянии всех узлов комбайна – всего 53 параметра. Для тестирования блоков требуется подключить к диагностическому стенду большое количество оборудования: четыре генератора синусоидальных сигналов; два частотомера; магазин сопротивлений; дополнительный источник питания, напряжением 5 вольт. Далее, в зависимости от тестируемого параметра, необходимо многократно переключать три галетных переключателя имеющих десять положений, магазин сопротивлений, а также изменять частоту одного из четырёх генераторов соответствующего канала. Недостаток - легко запутаться в этой последовательности переключений, поэтому необходимо постоянно иметь перед глазами руководство о последовательности переключений и параметрах по проверке блоков. Другой недостаток – громоздкое подключаемое оборудование, большое количество проводов подключения к стенду. Если в условиях завода это большой роли не играет, то при проверке блоков в полевых условиях работниками дилерских центров, таким диагностическим стендом пользоваться невозможно.

Для решения этой проблемы была поставлена задача создания диагностического стенда, лишённого многих недостатков стационарного стенда. Предлагаемый мной диагностический стенд

в большей степени обладает многими преимуществами по сравнению со стационарной моделью в плане мобильности. У него небольшие габаритные размеры и определяются в основном размерами переменных резисторов для его настройки. Конструктивно стенд значительно отличается от стационарного. Во-первых, он уже содержит встроенные простейшие генераторы с выведенными на панель отградуированными резисторами для регулировки частоты. Поэтому не нужны внешние генераторы, которыми раньше производилась настройка, а точная градуировка резисторов исключает частотомеры. Во-вторых, на панель выведены резисторы для регулировки температуры, давления, зазоров механизмов и уровня топлива, что значительно упростило считывание информации. Теперь отпадает необходимость в магазине сопротивлений, который необходимо было последовательно подключать к каждому каналу отдельно. В-третьих, ряд микрокнопок на панели позволяет быстро проверить срабатывание аварийных и предупреждающих датчиков. В-четвертых, включив стенд с настроенными параметрами и подключённым к нему бортовым компьютером, на мониторе сразу видно цифровое и графическое изображение всех параметров комбайна, по которому можно легко определить его исправность. В-пятых, и это главное, диагностический стенд может быть использован дилерскими центрами в полевых условиях для быстрого определения неисправности, так как он может питаться от аккумулятора комбайна, и имеет малые размеры. Также стенд будет незаменим в ремонтных мастерских занимающихся ремонтом бортовых компьютеров устанавливаемых на зерноуборочных комбайнах. При этом он подойдёт для ремонта блоков и других производителей, таких как Россия и Украина. Для этого достаточно распаять выводы разъёма к блоку ввода-вывода, в соответствии со схемой разработчика, так как у разных производителей своя распайка разъёма к блоку, но поступающие сигналы с датчиков комбайна при этом не меняются.

Модель диагностического стенда была испытана в стационарных условиях, при ремонте блоков БКИ (блок контроля и индикации). Скорость диагностики неисправности в несколько раз превзошла поиск той же не исправности на заводском стенде, так как нет необходимости в многократных переключениях. Испытания в полевых условиях подтвердили сказанное о необходимости создания этого стенда, показав очень хороший результат при проверке блоков БКИ. Не заводя комбайна можно было легко определить, исправен БКИ или в нём есть какая-нибудь неисправность. В случае неправильной настройки блока, его можно настроить на месте, введя правильные параметры.

Пользоваться диагностическим стендом довольно просто, достаточно подключить к нему бортовой компьютер и включить питание. На дисплее мы увидим процесс загрузки, а через несколько секунд появится изображение со всеми параметрами комбайна. Учитывая то, что в учебных центрах уже проводятся каждый год курсы обучения своих представителей для настройки бортовых компьютеров, то не потребуются проводить дополнительные занятия с обслуживающим персоналом. Тем более, что работа со стендом интуитивно понятна.

Сейчас ведётся доводка стенда в плане дизайна и улучшения его потребительских качеств.

**А.А. Ковалёв (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Е.А. Левчук**, к.т.н., доцент

## **ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ СЕРВИСА ДЛЯ ЭКСПОРТА ЖУРНАЛА РЕГИСТРАЦИИ ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ В ЧАСТНОЕ ОБЛАКО**

Разработанный сервис позволяет отправлять с сервера файлы журнала регистрации исключительных ситуаций в файловое хранилище облачного сервиса Amazon S3. Эта операция происходит по расписанию с заданным интервалом времени. В результате при возникновении ошибок на сервере, достаточно просмотреть поведение веб-приложения в это время. Такой механизм помогает быстро найти место ошибки, понять и устранить причину, исключив взаимодействие с владельцем веб-приложения.

В веб-приложениях, написанных на Java, для ведения журналов регистрации исключительных ситуаций чаще всего используется библиотека log4j. Для формирования ссылки для отправки любого файла, при использовании REST API, нужно приложение curl. Для работы сервиса также необходима утилита, предоставляющая набор команд, в основе которых лежит предоставляемый компанией Amazon для доступа к хранилищу сервис REST API (Aws cli tool). Сервисом используется команда s3put, отправляющая файл в облачное хранилище данных Amazon S3.

```
s3put $s3_bucket/$s3_path/${log_file#$apin/} $logfile
```

Сервис является системным скриптом, поставленным в расписание ОС сервера на ежечасное выполнение. Для начала необходимо определить переменные среды для последующей быстрой настройки.



```
app_dir=./Applications
s3_path=$(hostname)/$(date -d last-hour '+%Y')/$(date -d last-hour '+%m-%d')
timestamp=$(date -d last-hour '+%Y-%m-%d-%H')
...
s3_bucket=company.logs.$app_name
```

В переменную `app_dir` помещается путь к директории приложения. Переменная `s3_path` содержит структуру директорий внутри корзины. Имя корзины помещается в переменную `s3_bucket`. Переменная `timestamp` – это строка содержащая текущую дату и время в нужном формате. В таком же формате записывается строка расширения файлов журнала, сохраняемых каждый час. Теперь нужно проверить директорию `app_dir` на наличие директории `logs` и приложений, содержащих директорию `logs`. После нахождения нужных директорий они проверяются на наличие сохранённых за прошлый час файлов журнала. Когда один из нужных нам файлов будет найден, он отправляется в соответствующую корзину.

**А.А. Ковалёв (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **В.Д. Левчук**, к.т.н., доцент

## **АРХИТЕКТУРНЫЕ РЕШЕНИЯ РЕАЛИЗАЦИИ СЕРВИСА ДЛЯ ЭКСПОРТА ЖУРНАЛА РЕГИСТРАЦИИ ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ В ЧАСТНОЕ ОБЛАКО**

Крупным веб-приложениям требуется непрерывная техническая поддержка и сопровождение. Чаще всего они сохраняют свои действия в журнал регистрации исключительных ситуаций. Это делается для того, чтобы при возникновении ошибок на сервере, просмотреть поведение веб-приложения в это время. Такой механизм помогает быстро найти место ошибки, понять и устранить причину.

Однако после выхода приложения, оно помещается на сервер, доступ к которому обычно закрыт для разработчика из соображений безопасности. В данной ситуации, при возникновении ошибок, достаточно проблематичным является получение журнала разработчиком и занимает довольно длительное время.

В настоящее время стремительно набирает популярность использование облачных технологий в различных сферах. В связи с этим растёт количество всевозможных облачных сервисов, как платных так и бесплатных для некоммерческого использования.

Разработанный сервис позволяет отправлять с сервера файлы журнала регистрации исключительных ситуаций в файловое хранилище облачного сервиса Amazon S3. Эта операция происходит по расписанию с заданным интервалом времени.

В веб приложениях, написанных на Java, для ведения журналов регистрации исключительных ситуаций чаще всего используется библиотека log4j.

Log4j — библиотека журналирования Java программ, часть общего проекта «Apache Logging Project».

Для настройки логирования используется конфигурационный файл log4j.xml. В основе библиотеки Log4J лежит три понятия – логгер (logger), аппендер (appender) и компоновка (layout). Логгер представляет собой объект класса org.apache.log4j.Logger, который используется для вывода данных и управления уровнем (детализацией) вывода. Log4J поддерживает следующие уровни вывода, в порядке возрастания: TRACE, DEBUG, INFO, WARN, ERROR, FATAL, OFF. Установка логгеру определенного уровня означает следующее – сообщения, выводимые с этим или более высоким уровнем, попадут в лог. Сообщения, выводимые с уровнем ниже установленного в лог не попадут. Логгеры связываются с аппендерами в соотношении "многие ко многим" – у одного логгера может быть несколько аппендеров, а к одному аппендеру может быть привязано несколько логгеров. Важно понимать, что аппендеры наследуются от родительских логгеров.

В приложении используется DailyRollingFileAppender. Он ротирует файл с определенной частотой. Она зависит от шаблона, указанного в конфигурации:

- '.уууу-ММ – файл ротируется раз в месяц;
- '.уууу-ww – файл ротируется раз в неделю;
- '.уууу-ММ-dd – файл ротируется раз в день;
- '.уууу-ММ-dd-a – файл ротируется раз в полдня;
- '.уууу-ММ-dd-НН – файл ротируется раз в час;
- '.уууу-ММ-dd-НН-mm – файл ротируется раз в минуту.

При ротации к имени файла в конце приписываются текущие дата и время, отформатированные согласно указанному шаблону (с помощью класса java.text.SimpleDateFormat). В кавычках в начале шаблона указан символ, который будет использоваться как разделитель между значением даты/времени и именем файла.

Так как необходимо настроить логирование на почасовую ротацию, в конфигурационном файле будет использован шаблон '.уууу-ММ-dd-НН.

Ниже приведена часть кода конфигурационного файла:

```

<appender name="FlatFileAppender"
class="org.apache.log4j.DailyRollingFileAppender">
  <param name="append" value="true" />
  <param name="encoding" value="UTF-8" />
  <param name="file" value="mylogfile.log" />
  <param name="DatePattern" value="'.yyyy-MM-dd' />
  <layout class="org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="[%-25d{ISO8601}] %-5p %x
%C{1} -- %m\n" />
  </layout>
</appender>

```

Для формирования ссылки для отправки любого файла, при использовании REST API, нужно приложение curl.

Программа cURL может автоматизировать передачу файлов или последовательность таких операций. Например, это хорошее средство для моделирования действий пользователя в веб-обозревателе.

Программа поддерживает протоколы: FTP, FTPS, HTTP, HTTPS, TFTP, SCP, SFTP, Telnet, DICT, File: , LDAP а также POP3, IMAP и SMTP. Также cURL поддерживает сертификаты HTTPS, методы HTTP POST, HTTP PUT, загрузку на FTP, загрузку через формы HTTP.

Для работы сервиса также необходима утилита, предоставляющая набор команд, в основе которых лежит предоставляемый компанией Amazon для доступа к хранилищу сервис REST API. Aws cli tool – утилита, предоставляющая набор простых, интуитивно понятных команд. Сервисом используется команда s3put, отправляющая файл в облачное хранилище данных Amazon S3.

```
s3put $s3_bucket/$s3_path/${log_file#$apin/} $logfile
```

Сервис является системным скриптом, поставленным в расписание ОС сервера на ежечасное выполнение. Для начала необходимо определить переменные среды для последующей быстрой настройки.

```
app_dir=./Applications
```

```
s3_path=$(hostname)/$(date -d last-hour '+%Y')/$(date -d last-hour '+%m-%d')
```

```
timestamp=$(date -d last-hour '+%Y-%m-%d-%H')
```

```
...
```

```
s3_bucket=company.logs.$app_name
```

В переменную app\_dir помещается путь к директории Applications с описанной выше структурой размещения файлов журнала. Переменная s3\_path содержит структуру директорий внутри корзины. Имя корзины помещается в переменную s3\_bucket. Переменная timestamp – это строка содержащая текущую дату и время в нужном формате. В таком же

формате записывается строка расширения файлов журнала, сохраняемых каждый час.

Теперь нужно проверить директорию `app_dir` на наличие директории `logs` и приложений, содержащих директорию `logs`. После нахождения нужных директорий они проверяются на наличие сохранённых за прошлый час файлов журнала. Это делается при помощи описанной выше переменной `timestamp`. Когда один из нужных нам файлов будет найден, необходимо поместить его в соответствующую корзину с правильной структурой директорий внутри, используя утилиту `aws cli tool`.

Для выполнения данного скрипта требуется любая операционная система Linux с установленными на ней приложениями `curl` и `aws cli tool`. К аппаратному обеспечению никаких требований не имеется, требования устанавливаемой системы Linux, так как скрипт не запрашивает больше чем простые операции, выполняемые им. Конечные пользователи являются разработчиками и никак не взаимодействуют с приложением напрямую. Им только нужно знать имя сервера, приложения, дату и время возникновения интересующего их события. Дальше просто найти нужный файл в интуитивно-понятной структуре директорий.

**Е.А. Ковалёва (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Н. Леванцов**, старший преподаватель

## **ПРОЕКТИРОВАНИЕ СИСТЕМЫ АВТОМАТИЗАЦИИ УСЛУГ СТРОИТЕЛЬНОЙ ОРГАНИЗАЦИИ**

В настоящее время на рынке существует большое количество решений для автоматизации практически любой деятельности организаций. Говорить о том, что каждое из этих решений идеально впишется в рамки бизнес процессов, протекающих в любой организации, не приходится. Стандартные решения автоматизации способны удовлетворить многие, но не все требования организаций. В независимости от отрасли, или сферы ведения хозяйственной деятельности одно и то же готовое средство может оказаться излишне функциональным – в лучшем случае, или иметь изъян в виду своей ограниченности.

Сейчас в мире идет процесс максимального внедрения компьютерных систем и программ во все сферы производства, создание прикладного программного обеспечения. Сегодня уже не найти сферу человеческой деятельности, где бы не использовались информационные технологии. Для работы в разных сферах деятельности необходимо участие

человека, умение проводить аналитические расчеты, знание логических взаимосвязей, заполнение форм бухгалтерской отчетности и многого другого.

Больших успехов достигла строительная отрасль с внедрением персональных компьютеров в управление производством.

К примеру, в строительстве компьютерные технологии помогают делать необходимые расчеты, сметы, создавать проектную документацию, разрабатывать схемы, вести учет материалов, вести полную бухгалтерскую отчетность, чертить схемы, наблюдать за выполнением многих задач и очень много других операций. Сегодня уже невозможно представить даже небольшое строительное производство без использования возможностей персонального компьютера.

Проектирование является основополагающей стадией в разработке программного обеспечения. Поэтому очень важно на этом этапе учесть все требования к программному продукту, создать соответствующую предметной области базу данных, предусмотреть связь между данными, обеспечить их непротиворечивость и целостность. Выбор средств реализации проекта также является ответственным шагом в проектировании прикладных приложений, так как используемые средства разработки должны обеспечивать возможность создания приложений с учетом всех заявленных требований от заказчика, быть современными, и находиться в ценовой категории тех, которые может позволить себе заказчик.

**И.А. Козлов (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.С. Давыдов**, к.т.н., доцент

## **ПРОГРАММА «REMEMBER THESE VERBS»**

Разработанная мною программа «Remember these verbs» предназначена для заучивания/запоминания английских фразовых глаголов. Данная программа подойдет для всех людей, которые желают улучшить свои знания в английском языке.

«Remember these verbs» работает так: через заданный промежуток времени (в моём случае, через каждые 15 секунд) выводит по середине экрана фразовый глагол. Фразовые глаголы программка считывает из файла phrasal\_verbs.txt, в который я поместил TOP-170 фразовых глаголов. Вдобавок к этому показывает текущее время (часы, минуты) и текущую дату (день, месяц, год) (см. рисунок 1).

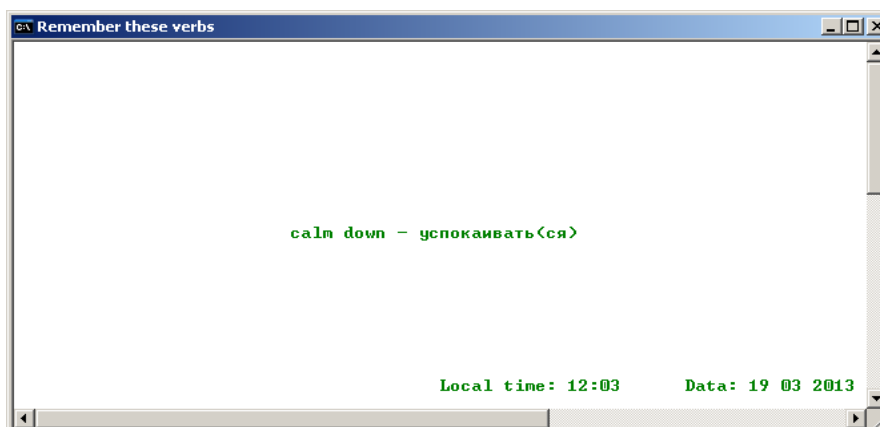


Рисунок 1– Пример работы программы «Remember these verbs» после смены N числа глаголов

Программный код:

```
#include <stdio.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>
#include <time.h>
#include <windows.h>
//remove cursor
void ShowCursor(bool show)
{
    HANDLE hConsoleOutput;
    CONSOLE_CURSOR_INFO structCursorInfo;
    hConsoleOutput = GetStdHandle( STD_OUTPUT_HANDLE );
    GetConsoleCursorInfo( hConsoleOutput, &structCursorInfo );
    structCursorInfo.bVisible = show;
    SetConsoleCursorInfo(hConsoleOutput, &structCursorInfo );
}
main(){
void ShowCursor(bool show);
int vsp;
int c=0;
char string[1000];
SYSTEMTIME lt;
FILE *f=fopen("D:\\for konkurs\\phrasal_verbs.txt", "r");
if (f==NULL) perror("Can't open file");
else {
    //заголовок
    system("TITLE Remember these verbs");
    while((c=getc(f))!=EOF){
```

```

//часы
HANDLE hd = GetStdHandle(STD_OUTPUT_HANDLE);
COORD cd;
cd.X=40;
cd.Y=22;
SetConsoleCursorPosition(hd,cd);
GetLocalTime(&lt);
printf("Local time: %02d:%02d      Data: %02d %02d %04d\r",
lt.wHour, lt.wMinute, lt.wDay, lt.wMonth, lt.wYear);

//читаем построчно
fgets(string,1000,f);

//кодируем строки, чтобы считывало русские символы
CharToOem (string,string);

//выравнивание по центру
if (strlen(string)%2==0) vsp=(80-strlen(string))/2;
else vsp=(80-(strlen(string)-1))/2;

//пишем по центру
cd.X=vsp;
cd.Y=12;
SetConsoleCursorPosition(hd,cd);

//прячем курсор
ShowCursor(false);

//меняем цвет: F-белый цвет фона, 2-цвет текста
system("color F2");

//выводим строку
puts(string);

//задержка на 15 сек; 1 сек = 1000 мсек
Sleep(15000);

//сигнал о завершении показа глагола
Beep(50, 10);

//чистим
system("cls");
}
}
while(!kbhit());
}

```

**И.В. Козырев (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Н. Леванцов**, старший преподаватель

## **ИНТЕГРАЦИЯ СИСТЕМЫ 1С: ПРЕДПРИЯТИЕ 8 С ДРУГИМИ ИНФОРМАЦИОННЫМИ СИСТЕМАМИ**

Часто при решении задач комплексного учета возникает необходимость в рамках одного программного комплекса использовать возможности другого программного комплекса, проводить обмен данными между различными системами и т. п.

К примеру, при необходимости работы с каким-либо оборудованием «общение» с ним производится через специализированную программу (драйвер). В компании может быть принято оформлять договоры в формате Microsoft Word. В связи с этим возникает необходимость работы с данной программой, хранения полученных документов в информационной базе. При организации обмена с такой же системой (информационной базой) или с другими возникает как задача формирования (и чтения) файла обмена, так и задача его доставки.

На самом деле все подобные задачи не перечислить. Но возможность их решения определяет, насколько велик потенциал используемого программного комплекса по интеграции (организации совместной работы) с другими системами. Система «1С:Предприятие» является открытой. Предоставляется возможность для интеграции практически с любыми внешними программами и оборудованием на основе общепризнанных открытых стандартов и протоколов передачи данных.

В системе «1С:Предприятие» имеется целый набор средств, с помощью которых можно:

- создавать, обрабатывать и обмениваться данными различных форматов;
- осуществлять доступ ко всем объектам системы «1С:Предприятие», реализующим ее функциональные возможности;
- поддерживать различные протоколы обмена;
- поддерживать стандарты взаимодействия с другими подсистемами;
- разрабатывать собственные интернет-решения.

Проведем краткий обзор технологий, которые могут использоваться платформой «1С:Предприятие» при решении задач интеграции.

Текстовые документы. Встроенный язык системы позволяет разработчику создавать, динамически формировать и записывать текстовые документы.

Обмен данными с использованием текстовых документов может



быть одним из менее «ресурсоемких» способов организации взаимодействия с другими информационными системами.

Помимо обычных способов работы с текстовыми документами (чтение, запись, вставка и добавление строк, получение строк), разработчик имеет возможность динамического формирования текстовых документов на основе заранее созданных шаблонов.

Текстовые файлы. Обмен с помощью текстовых файлов – наиболее простой механизм обмена данными. Он может быть использован для решения самых разнообразных задач. Его основное преимущество – простота освоения и удобное текстовое представление информации.

Форматированные документы. Форматированный документ предназначен для создания и редактирования текста различных форматов, содержащих оформление (картинки, текст, таблицы).

XML-документы. Система позволяет организовывать интеграцию с прикладными системами с использованием XML-документов, являющихся на сегодняшний день общепринятым средством представления данных.

DBF-файлы. Механизм работы с базами данных формата DBF предназначен для обеспечения возможности манипулирования ими непосредственно из встроенного языка системы. Возможно практически любое манипулирование данными.

Внешнее соединение. Основная задача, решаемая с помощью внешнего соединения – обеспечение надежного и быстрого программного доступа к данным системы «1С:Предприятие» из внешних приложений. В общем и целом работа с системой через внешнее соединение подобна работе с системой в режиме Automation-сервера. Существенное отличие заключается в том, что запускается вариант «1С:Предприятия», не содержащий интерфейсной части (подразумевается работа с данными только программными средствами, без использования интерфейсных возможностей).

Automation Client/Server. Основное назначение Automation-сервера «1С:Предприятия» – управление приложением системы «1С:Предприятие» из других приложений и выполнение действий, аналогичных интерактивным действиям.

HTML-документы. Средства работы с HTML-документами позволяют встраивать их в формы прикладного решения и выполнять редактирование средствами встроенного языка.

Работа с файлами. «1С:Предприятие» средствами встроенного языка предоставляет доступ к функциям работы с файловой системой. Эта возможность может быть использована при организации взаимодействия с другими информационными системами через общие каталоги.

Макеты ActiveDocument. Технология ActiveDocument предназначена для редактирования документов внешними по отношению к

«1С:Предприятию» редакторами.

Технология внешних компонентов, разработанная для решения специальных задач интеграции, в которых требуется тесное взаимодействие между системой «1С:Предприятие» и другими программами.

Работа с Интернетом, которая возможна непосредственно из встроеного языка. Разработчик может выполнять отправку и прием писем электронной почты, а также осуществлять обмен данными по протоколам HTTP (HTTPS) и FTP.

Web-расширение является отдельным программным продуктом и позволяет встраивать доступ к данным «1С:Предприятия» в существующие веб-сайты и веб-приложения, а также создавать готовые веб-приложения, использующие информационную базу «1С:Предприятия».

Обмен данными. Механизмы, реализованные в технологической платформе «1С:Предприятие», позволяют создавать территориально распределенные информационные системы как на основе информационных баз «1С:Предприятия», так и с участием других информационных систем, не основанных на «1С:Предприятии». Например, можно организовать работу главного офиса, филиалов и складов предприятия в единой информационной базе или обеспечить взаимодействие информационной базы «1С:Предприятия» с существующей базой данных, например, MySQL.

Web-сервисы. Механизм Web-сервисов позволяет создавать web-сервисы в конфигурации «1С:Предприятия», а также взаимодействовать в конфигурации «1С:Предприятия» с веб-сервисами, опубликованными сторонними поставщиками.

XDTO. Механизм XDTO предназначен прежде всего для описания типов параметров и возвращаемых значений Web-сервисов. Также этот механизм может использоваться для обмена данными между различными конфигурациями «1С:Предприятия» или другими информационными системами.

### **Литература**

1. Радченко, М.Г. 1С: Предприятие. Практическое пособие разработчика. Примеры и типовые приемы / М.Г. Радченко, Е.Ю. Хрусталева – М.: ООО «1С-Публишинг», 2009 – 874 с.
2. Томас, К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / К. Томас, Б. Каролин – М.: Вильямс, 2004. – 143 с.
3. Хрусталева, Е.Ю. Разработка сложных отчетов в 1С: Предприятие 8. Система компоновки данных / Е.Ю. Хрусталева – М.: ООО «1С-

Паблiшинг», 2009 – 513 с.

4. Бойко, В.В. Проектирование баз данных информационных систем / В.В. Бойко, В.М. Савинков – М.: Финансы и статистика, 1989. – 351 с.

**Я.А. Коледа (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **В.Н. Кулинченко**, старший преподаватель

## **СОЗДАНИЕ НАГЛЯДНОГО ПОСОБИЯ ДЛЯ КУРСА «АРХИТЕКТУРА ЭВМ»**

Целью данного проекта являлось создание наглядного пособия для курса «Архитектура ЭВМ» с помощью языка разметки страниц html и прототипно-ориентированного скриптового языка программирования JavaScript. В содержание пособия входит подробное описание материнских плат и их элементов. Приложение выполнено как множество связанных html-документов, такая организация хранения данных обеспечивает наиболее быстрый и лёгкий способ доступа к информации, возможность её корректировки и изменения.

Список системных плат, описанных в созданном приложении:

- Gigabyte GA-8PEMT4;
- Albatron kx600 pro;
- ASRock 775i 915p;
- Foxconn 6100 k8mars.

К данному приложению были предъявлены следующие требования:

- информативность;
- наглядность;
- самодостаточность;
- понятный интерфейс.

Аналогом данного пособия является наглядный стенд, на котором располагаются материнские платы и приведен список их характеристик. Такой вид представления материала является очень наглядным, однако для поиска описания конкретного элемента необходимо задействовать другие ресурсы (например, справочные пособия сервисные мануалы или другие источники).

Преимущество разработанного проекта в том, что он в равной степени является и наглядным, и информационно достаточным. Описание каждой материнской платы включает в себя не только перечисление характеристик значимых элементов, но и описание их устройства, совместимость и др. Кроме этого, страницы пособия можно легко редактиро-

вать, дополнять новым материалом и информацией.

Для работы с созданным пособием необходимо открыть документ Index.html с помощью любого браузера, поддерживающего язык разметки страницы html и язык JavaScript. В открывшемся окне Список материнских плат в левом столбце нужно выбрать интересующую материнскую плату и нажать один раз левой кнопкой мыши по ссылке (названию платы). В этом же окне будет открыт следующий документ – описание выбранной платы.

В ходе разработки были выявлены некоторые особенности работы с языком разметки страниц html и прототипно-ориентированным скриптовым языком программирования JavaScript, такие как: создание html-документов, работа с таблицами, фреймы, функции и формы web-страниц. При создании пособия особое внимание уделялось непосредственной разработке структуры документов и корректному их взаимодействию.

### **Литература**

1. Флэнаган, Д. JavaScript. Подробное руководство: Учебное пособие/ Д. Флэнаган – СПб.: Символ-Плюс, 2008. – 992 с.
2. Мержевич, В. Html и CSS на примерах: учебный курс/ В. Мержевич – СПб.: БХВ-Петербург, 2005. – 448 с.
3. Мюллер, С. Модернизация и ремонт ПК, 14 издание: Научно-популярное издание/ С. Мюллер – М.: Издательский дом «Вильямс», 2003. – 1184.: ил.

**М.Г. Кондратенко (УО «ГГУ им. Ф.Скорины», Гомель)**  
Науч. рук. **С.В. Карпенко**, к.э.н., доцент

### **ИНТЕРНЕТ-БАНКИНГ В РЕСПУБЛИКЕ БЕЛАРУСЬ**

По данным международного исследовательского агентства Gemius, в Беларуси количество пользователей интернета за 2012 год увеличилось на 17 % — с 3,67 млн. до 4,29 млн. человек. Причем большинство из них женщины (50,35 %). Что касается возраста, то интернет больше популярен среди молодых людей в возрасте 25—34 лет (30,27 %) и молодежи 19—24 лет (20,67 %). Среди пожилых людей в возрасте от 55 лет и выше всемирную паутину освоили лишь 7,17 % белорусов.

По данным мартовского 2012 г. исследования, подавляющая часть белорусской интернет-аудитории (77,32 %) проживает в Минске либо

крупных городах.

По роду деятельности лидируют специалисты (25 %), на втором месте — студенты и учащиеся (около 16 %).

Размер интернет-аудитории Беларуси в сентябре 2012 года составил 4,383 миллиона человек. Это на 14 % больше, чем в сентябре 2011 года.

Количество интернет-пользователей в Гомельской области выросло на 26 % за год.

В настоящее время более 460 тысяч жителей Гомельской области пользуются услугами Интернет-провайдеров, что составляет 14 % всей Интернет-аудитории Беларуси.

Функциональные возможности ИБ:

- регистрация клиентов в режиме самообслуживания, вход зарегистрированных пользователей в систему;

- операции с платежами (оплата коммунальных услуг, погашение задолженности по кредитам и др.): создание нового платежа, добавление подписки, оплата по подписке, редактирование подписки, удаление подписки, получение истории платежей;

- операции с карточками: активация и деактивация карточек для работы в системе, получение информации об условиях пользования карточками, получение остатка и выписки о транзакциях по карточкам, подключение дополнительных услуг, оформление заявки на изготовление карточек, перевод денежных средств с карточки на карточку, блокировка/разблокировка карточки;

- операции с кредитами: просмотр задолженности по основному долгу; просмотр суммы к погашению и срока погашения основного долга и начисленных процентов по кредиту; просмотр информации о размере процентной ставки и сумам начисленных процентов по кредиту за период; просмотр истории, совершенных платежей по кредиту за период.

- операции со счетами и вкладами: просмотр информации об остатках по счетам и вкладам (депозитам), размере процентной ставки по счетам и вкладам (депозитам), даты открытия/закрытия договора.

- переписка: чтение и отправка сообщений в системе;

- операции с настройками: изменение регистрационных данных пользователя, смена пароля пользователя в системе.

Функциональные возможности в мобильной версии интернет-банкинга ограничены: реализована не полная функциональность. Некоторые функциональные возможности полной версии интернет-банкинга в ней отсутствуют, а именно:

- регистрация в системе интернет-банкинг;

- оформление заявки на изготовление карточек;

– активация первой карточки.

Главные недостатки интернет-банкинга:

Во-первых, это относительно более низкий уровень защиты по сравнению с системой «Клиент-Банк» и документарным оформлением транзакций. Хотя технология SSL и является стандартом интернет-безопасности, она в силу своей распространенности хорошо известна потенциальным взломщикам и не может гарантировать тот же уровень безопасности, что и система «Клиент-Банк», которая работает, как правило, в замкнутых интернет-сетях, не имеющих выхода в Интернет.

Во-вторых, высокие начальные капиталовложения. Для запуска одной системы интернет-банкинга требуется от 1 до 5 млн. долларов США.

Стоимость регистрации в системе «ИБ» в учреждениях банка составляет: для студентов (курсантов, учащихся) дневной формы обучения, получающих высшее, среднее специальное и профессионально-техническое образование в учебных заведениях Республики Беларусь – 2 000 бел. рублей; для держателей карточек Visa Gold и MasterCard Gold, БелКарт-М «Студенческая» – бесплатно; для иных категорий клиентов – 10 000 бел. рублей. Стоимость on-line регистрации в «ИБ» составляет 20 000 бел. рублей для всех категорий клиентов. Плата носит разовый характер. В 2012 г. плата не изменилась.

**А.В. Королёва (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Н. Леванцов**, старший преподаватель

## **РАЗРАБОТКА ПОДСИСТЕМЫ ПО УЧЕТУ ЗАПАСОВ ПРЕДПРИЯТИЯ В СРЕДЕ 1С: ПРЕДПРИЯТИЕ**

Использование ЭВМ в качестве средства хранения информации повлекло за собой появление программ, позволяющих создавать и редактировать базы данных, начался постепенный переход бумажного потока документов в электронный. Среди всех сред разработки, учитывая задание по курсовой работе, я остановилась на системе 1С: Предприятие. Несмотря на то, что эта система является далеко не единственной на рынке программных продуктов аналогичного назначения, она безусловно относится к наиболее популярным.

Для реализации автоматизированной системы по учету запасами предприятия был рассмотрен упрощенный документооборот и основные операции предприятия.

Для простоты использования программы мною были реализованы подсистемы с расположенными в каждой из них элементами. Каждый

элемент имеет свои атрибуты и обработчики событий. Каждый элемент вызывает работу определенных функций по созданию и сохранению данных, а также по получению данных.

Для хранения списочной информации были созданы справочники, справочник Ответственные для хранения списка сотрудников, которые работают на предприятии. Для хранения списка клиентов предприятия был создан справочник Клиенты. Учет номенклатуры ведётся в справочнике Номенклатура и отвечает за список продаваемых и получаемых материалов. Перечисление складов на которых хранится материал содержится в справочнике Места хранения. Во всех из перечисленных выше справочников можно изменить данные, удалить, а также дополнить их в подсистеме Продажи на панели действий Создать.

Так как для учета запасов предприятия важно фиксировать приход материалов был создан документ ПоступлениеМЦ, который содержит в себе данные о материале, его стоимости и количестве. Документ РеализацияМЦ необходим для того, чтобы знать какие материалы были затрачены на предприятии.

Для правильно работы документов созданы четыре регистра. Регистр накопления ПартииТМЦРегламентные содержит информацию о том, сколько и каких материалов есть на складах. Для фиксирования движений документов по регистрам была использована функция Движения.

Так как стоимость услуг имеет особенность меняться со временем был создан регистр сведений Цены, который хранит развернутые данные развернутые во времени розничные цены материалов и стоимости услуг. При выборе номенклатуры в документ подставляется актуальная цена, для этого была использована функция ПолучитьПоследнее, которая сохраняется в переменной ЗначенияРесурсов. Регистр ГрафикиРаботы содержит информацию о графиках работы сотрудников.

Для фиксирования денежных средств начисленных сотруднику, премий и лишений из-за невыхода на работу создан регистр расчета Начисления.

Для того чтобы пользователь мог получать необходимые ему выходные данные созданы отчеты в которых пользователь сам может задавать критерии создания отчетов, такие как даты начала и окончания.

**В.В. Костюкевич (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Н.А. Аксёнова**, ассистент

## **СОЗДАНИЕ COGNOS ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ COGNOS BI**

Для создания cognos-приложения был использован такой программный продукт, как Cognos Report Studio.

Cognos – основная платформа бизнес-анализа корпорации IBM. В основе идеологии Cognos лежит концепция Corporate Performance Management, которая подразумевает тесную интеграцию не только средств для визуализации и анализа данных, но также и для планирования и бюджетирования, системы сбалансированных показателей (Balanced Scorecard) и др.

Компоненты программного обеспечения IBM Cognos BI предоставляют следующую функциональность:

- отчетность – разработка и предоставление любых типов отчетов, оперативно и легко адаптируемых к любому источнику данных;
- аналитика – проведение многомерного анализа и формирование отчетов;
- информационные панели – оперативная доставка сложной информации благодаря возможности легко создавать различные информационные панели;
- управление по ключевым показателям – использование карт показателей (метрик) для обеспечения согласованности проектов и тактических действий с корпоративной стратегией.

Проект Cognos BI подразделяется на следующие студии, выполняющие отдельные функции:

Analysis Studio, используется для многомерного анализа и исследования крупных источников данных;

Report Studio, предназначается для создания любого типа отчета, включая диаграммы, карты, списки и репитеры, используя любой источник данных (реляционный или многомерный). Предлагается расширенный набор функций, доступных в режиме быстрого создания отчетов;

Query Studio, инструмент для создания простых отчетов и запросов в IBM Cognos;

Event Studio, инструмент для создания уведомлений, который позволяет уведомлять ответственных лиц организации о происходящих событиях, чтобы они могли своевременно и эффективно принимать решения.

Преимущества системы:

- простой и интуитивно понятный инструментарий позволяет создавать аналитическую отчетность любой сложности. Интерфейс конечного пользователя рассчитан на пользователей информации и не требует постоянной поддержки ИТ-специалистов;
- поддержка в отчетах функций drill-down и drill-through позволяет мгновенно переходить от общего к более детальному представлению информации;



– система легко интегрируется со всеми уже существующими источниками информации на предприятии, что позволяет получить эффект от внедрения в короткие сроки;

– встроенная функция оповещения позволяет не тратить время на ежедневное отслеживание каких-либо показателей или событий. Надо лишь определить элементы отчета, по которым необходимо получать оповещения, задать условия и способ информирования, и система будет посылать оповещения при наступлении данных событий.

Доступ к каждому режиму создания отчетов определяется правами пользователя в отношении безопасных функций и свойств. На рисунке 1 представлен внешний вид рассматриваемого программного продукта.

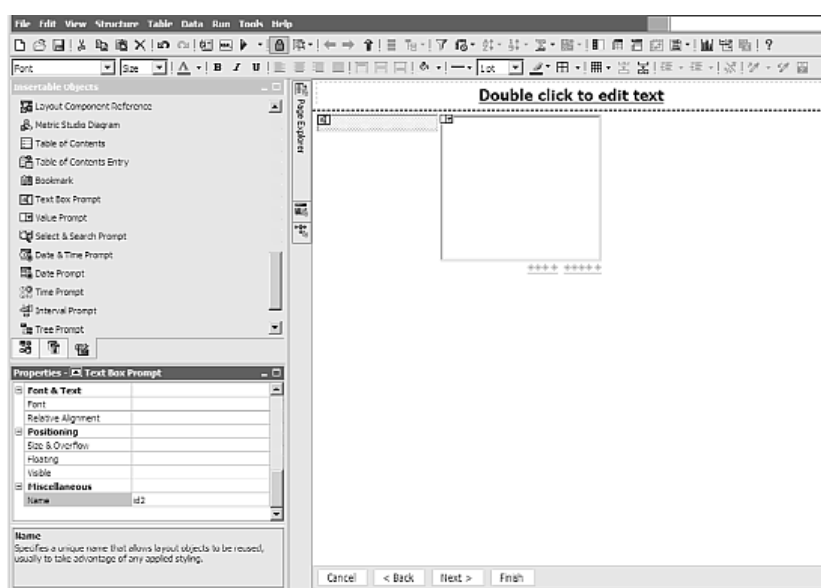


Рисунок 1 – Интерфейс приложения Cognos Report Studio

Report Studio выполняет следующие функции:

- создание многостраничных отчетов с произвольным расположением таблиц и графиков;
- полный контроль над свойствами элементов отчета;
- возможность получать и объединять данные из нескольких метамodelей в рамках одного отчета;
- внедрение мультимедийных объектов;
- условные вычисления на основе данных метамодели или полученного отчета;
- расширенные возможности условного форматирования - возможность персонализации вида отчета (скрытие строк или столбцов, цветовое кодирование значений) на основе неограниченного числа критериев;

– задание параметров и внедрение элементов управления фильтрацией данных в тело отчета.

К видам создаваемых отчетов можно отнести следующие наиболее часто используемые: List (список), Crosstab (перекрестная таблица), Map (карта), Chart (диаграмма), Blank (пустая область).

List – это отчет, который показывает данные, расположенные в строках и столбцах. Каждый столбец показывает все значения элементов данных в базе или данные, рассчитанные на основе данных элементов в базе.

Crosstab – это многомерные объекты, имеющие грани строк и грани столбцов. Каждая грань состоит из набора узлов перекрестных таблиц. Каждый узел перекрестной таблицы содержит следующие элементы: один или более членов узла перекрестной таблицы; ни одного или один вложенный узел перекрестной таблицы, в котором содержится один или несколько членов узла перекрестной таблицы или вложенные узлы перекрестной таблицы. Каждый член узла перекрестной таблицы обращается к элементу данных, содержащему выражение, которое определяет члены, отображаемые в перекрестной таблице.

Map – используется для отображения информации в виде карт для заданных территорий.

Chart – используется для графического отображения информации, которая требует анализ. Диаграммы бывают нескольких видов, наиболее часто используемые: столбцы (column chart), линии (line chart), радар (radar), точечная диаграмма (point chart), калибровочная диаграмма (gauge chart) и т. д.

Blank – пустая рабочая область, в которую из панели инструментов можно перетащить любой из видов отчетов, либо создать их комбинацию.

Программный продукт Cognos BI обладает интуитивно понятным интерфейсом, что позволяет пользователю, обладающему даже базовыми навыками работы в данном приложении, создавать свои информационные панели с набором стандартных функций.

**В.В. Костюкевич (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Н.А. Аксёнова**, ассистент

## **ПРОГРАММНЫЙ ПРОДУКТ FRAMEWORK MANAGER**

Framework Manager – программный продукт компании IBM. Framework Manager хранит все виды информации о данных в рамках модели. Этот процесс известен как моделирование метаданных. Используемые элементы

в Framework Manager из нескольких таблиц базы данных могут быть собраны в единый элемент (известный как запрос субъекта), назван удобным и осмысленным именем, а затем опубликован в виде пакета данных для дальнейшего использования разработчиками. Опубликованный пакет с данными в дальнейшем используется разработчиками cognos-приложений, которые позволяют просматривать и анализировать полученные из базы данные. Функция тестирования модели позволяет разработчику увидеть, как будут выглядеть данные, когда модель будет опубликована.

Как только информация, такая как Joins и калькуляции опубликованы, её можно использовать снова и снова, без риска получения неправильных Joins. Потенциальные ошибки в отчётах, известные как «reporting traps» могут быть исправлены в рамках модели данных.

Framework Manager имеет сложную систему безопасности, так что разработчик может ограничить доступ к определённым элементам для отдельных пользователей, групп пользователей или пользователей, которым назначены назначенных определённые роли, например, администратор.

Данные могут быть смоделированы relationally (реляционно) или dimensionally, что означает, что данные могут быть представлены в виде иерархий с несколькими уровнями.

Калькуляции могут производиться между различными элементами данных, используя широкий спектр функций и затем быть сохранены в Framework Model, чтобы каждый разработчик cognos-приложений мог получить к ним доступ. Фильтры могут быть встроены в модель данных и доступны для любого количества отчетов с помощью опубликованного пакета данных. После смоделированные данные могут быть упакованы по-разному, в зависимости от их функций.

Использование Framework Manager позволяет на уровне модели данных изменять иерархии данных, создавать новые элементы и калькуляции, что позволяет упростить работу разработчиков по созданию отчётов. Создание различных калькуляций и фильтров на стороне модели позволяет увеличить производительность создаваемых отчётов и избежать ошибок на уровне разработки самих cognos-приложений.

**В.И. Костюченко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **М.И. Жадан**, к.ф.-м.н., доцент

## **WEB-ОРИЕНТИРОВАННОЕ ПРИЛОЖЕНИЕ «ISSUE TRACKER»**

Крупные фирмы, занимающиеся разработкой программного обеспечения всегда пытаются улучшить свои продукты. Иногда это достигается

путём создания новых программных продуктов, а иногда – доработкой старых. В случае с доработкой старых всегда есть необходимость изменять и устранять ошибки, по причине которых данный продукт является ненадёжным или устаревшим. Чаще всего такие продукты являются достаточно объёмными, и их разработкой занимается не один разработчик, а, следовательно, и модифицировать или править такой продукт будет тоже не один. Возникает проблема – как же отдавать задания тому, или иному члену команды и при этом попытаться загрузить их равномерно? Для таких целей и было создано приложение «IssueTracker».

Данное приложение позволяет в онлайн режиме отслеживать и оповещать о наличии, либо исправлении той или иной проблемы связанной с функционалом и работоспособностью продуктов. Так же можно отслеживать, кто и когда занимался редактирование той или иной проблемы работоспособности.

Это приложение является web-ориентированным. Неплохим решением рассматриваемой задачи является реализация «IssueTracker» на языке Java. Java имеет свои собственные инструменты для реализации любого рода приложений, таких как: Servlets и JSP. По сути это два компонента, на которых и основана вся работа приложения.

Servlets – это интерфейс расширяющий стандартный возможности сервера и работающий по принципу «запрос-ответ». Вместе с JSP они реализуют такой общепринятый паттерн проектирования как MVC.

JSP – это «приятное» дополнение к servlets, которое позволяет отделить обработку данных от их отображения конечному пользователю. JSP позволяет динамически формировать HTML-страницу, а следовательно не приходится под каждый конкретный случай писать новую. Для расширения возможностей JSP, или если быть более точным HTML, используется каскадная таблица стилей CSS, которая позволяет быстро и удобно декорировать стандартные элементы, и JavaScript – позволяет максимально быстро выполнять те или иные действия, преимущественно без использования принципа «запрос-ответ».

Основной сферой применения данного приложения являются крупные компании-разработчики программных продуктов. На данный момент аналог такой системы используется в компании «ЕРАМ».

**А.А. Котович (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **В.Н. Кулинченко**, старший преподаватель

## **ОБЛАЧНЫЙ ХОСТИНГ И ОБЛАЧНЫЕ ВЫЧИСЛЕНИЯ**

Вычислительные системы все производительнее и с каждым днем все мобильнее. Чем мощнее система, тем сложнее предназначенные ей задачи. Не всегда есть возможность разработчику иметь все свои вычисления и наработки с собой, для этого были созданы так называемые облачные хранилища данных. В облаке данные можно не только хранить, но и просматривать, редактировать, обмениваться. Данные хранятся на многочисленных, распределённых в сети, серверах.

Для организации работы были созданы облачные шлюзы, предоставляющие удобное представление данных. С помощью соответствующего программного обеспечения, хранилище в облаке может быть представлено для клиента как локальный диск на компьютере.

Однако, повышение характеристики энергонезависимости устройств совсем не освобождает от сбоев питания, разрядки аккумуляторов и прочих трудностей. Нередки так же случаи сбоя операционных систем, нехватки ресурсов для вычислений. Как раз на такие случаи, и не только, были разработаны средства облачной обработки данных, где вычисления производятся на распределённых серверах. Изначально появление таких ресурсов было вызвано новой парадигмой в науке, основанной на необходимости обработки в реальном времени больших объемов данных, поступающих с разных источников (Data Intensive Science).

Наиболее характерной чертой облачного хостинга, выделяющей его среди других разновидностей, являются гибкие настройки и наличие масштабируемых ресурсов. В зависимости от того, какова нагрузка, клиент может увеличивать или уменьшать рабочую мощность сервера.

Большинство сервис-провайдеров предлагают облачные вычисления в форме VPS-хостинга, виртуального хостинга, и ПО-как-услуга (SaaS, Software-as-a-Service). Облачные услуги долгое время предоставлялись в форме SaaS, например, Microsoft Hosted Exchange и SharePoint. Хостинг на основе виртуальных частных серверов (VPS) является связующим звеном между виртуальным и выделенным хостингом. С помощью программ для виртуализации вы можете разделить один физический сервер на множество виртуальных серверов. Это обеспечивает конечным пользователям полный контроль над их хостинговым решением, включая административный доступ к их серверам и возможность их перезагрузки. Снижение издержек за счет использования виртуальных ПК происходит благодаря консолидации операций на центральных серверах и

виртуализации, обеспечивающей создание множества рабочих пространств на одном сервере. Так как затраты на управление ОС постоянно растут, компании ищут новые способы доставки приложений своим сотрудникам. Безопасность, скорость и пользовательский спрос – вот основные сложности, связанные с распространением виртуальных ПК. В SaaS-модели контроль и управление основной физической и виртуальной инфраструктурой облака, в том числе сети, серверов, операционных систем, хранения, или даже индивидуальных возможностей приложения (за исключением ограниченного набора пользовательских настроек конфигурации приложения) осуществляется облачным провайдером. Существуют и другие модели организации вычислений в облаке, как например, платформа как услуга (PaaS, Platform-as-a-Service), инфраструктура как услуга (IaaS, IaaS or Infrastructure-as-a-Service), рабочее место как услуга (WaaS, Workplace as a Service) и другие, но распространены они меньше.

Потребители облачных вычислений могут значительно уменьшить расходы на инфраструктуру информационных технологий (в краткосрочном и среднесрочном планах) и гибко реагировать на изменения вычислительных потребностей, используя свойства вычислительной эластичности (elastic computing) облачных услуг.

Для обеспечения согласованной работы узлов вычислительной сети на стороне облачного провайдера используется специализированное промежуточное программное обеспечение, обеспечивающее мониторинг состояния оборудования и программ, балансировку нагрузки, обеспечение ресурсов для решения задачи.

Одним из основных решений для сглаживания неравномерности нагрузки на услуги является размещение слоя серверной виртуализации между слоем программных услуг и аппаратным обеспечением.

По мнению аналитиков, в ближайшие 5-10 лет большая часть ИТ переместится в облака пяти различных типов. Будут проприетарные платформенные облака, предоставляющие различные платформенные услуги, – Google (тип 1), Microsoft (тип 2) и другие крупные ИТ игроки (тип 3), такие как IBM, Apple, HP и Amazon.

Будут облака услуг (тип 4), где мы ожидаем возникновение тысяч облачных провайдеров, предлагающих широкий спектр услуг. В качестве примера можно привести веб-хостинг и хостинг приложений, вертикально интегрированные структуры (правительство, здравоохранение, и т. д.), независимых производителей ПО (стратегическое развитие бизнеса, системы клиентской поддержки и т.д.), телекоммуникационные услуги (голосовая почта, VOIP). И наконец будут облака, управляемые корпоративными ИТ (тип 5), которые будут предоставлять услуги для внутреннего использования и для использования сотрудниками и партнерами.

**А.А. Крупский (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **С.Ф. Маслович**, к.т.н., доцент

## **ВЫСОКОНАГРУЖЕННАЯ СИСТЕМА ПАКЕТНОЙ ОБРАБОТКИ ИНФОРМАЦИИ**

В работе описывается высоконагруженная система с использованием пакетной обработки информации. В настоящее время тема высоконагруженных систем является одной из наиболее популярных областей знаний. Сам же термин “высоконагруженная система” при этом не имеет четкого определения. Но характеризуется двумя основными качествами: большое количество пользователей и данных. В связи с этим высоконагруженное приложение является многопользовательским. Однако не каждое многопользовательское приложения является высоконагруженным.

Пакетная обработка информации – один из видов организации вычислительного процесса на ЭВМ, при котором определенное число задач объединяется в пакет, обрабатываемый как единое целое.

При большом числе пользователей системы возрастает количество ресурсоёмких операций, например, соединение с базой данных, поэтому использование пакетной обработки информации позволяет ускорить работу высоконагруженной системы. Так же есть некоторые требования к приложениям с пакетной обработки данных:

1. Система должна поддерживать согласованность транзакций.
2. Система должна иметь небольшую задержку: обновления на сервере оперативной обработки должны быстро поступать на сервер пакетной обработки.
3. Система должна иметь высокую пропускную способность: она должна обрабатывать репликации большого числа транзакций.
4. Для обработки репликации должен требоваться минимальный объем служебных операций на сервере оперативной обработки.
5. Изменения данных могут передаваться в обоих направлениях: результаты пакетной обработки должны передаваться обратно серверу оперативной обработки.
6. Данные, необходимые на сервере пакетной обработки, могут быть подмножеством данных, доступных на сервере оперативной обработки.

Разработанное в ходе работы приложение можно позиционировать как законченный продукт, позволяющий обрабатывать большое количество запросов пользователей при оплате различных услуг с помощью обращения к банку.

**Е.В. Куликова (УО БТЭУ, Гомель)**  
Науч. рук. **С.В. Карпенко**, к.э.н., доцент

## **РАЗВИТИЕ 3G И 4G МОБИЛЬНОЙ СВЯЗИ**

3G (от англ. *third generation*), – набор услуг, который объединяет как высокоскоростной мобильный доступ с услугами сети Интернет, так и технологию радиосвязи, которая создаёт канал передачи данных. В настоящее время под этим термином чаще всего подразумевается технология UMTS. Сети 3G работают в диапазоне около 2 ГГц, передавая данные со скоростью до 3,6 Мбит/с.

3G включает в себя 5 стандартов семейства IMT-2000: 1) UMTS/WCDMA; 2) CDMA2000/IMT-МС; 3) TD-CDMA/TD-SCDMA (собственный стандарт Китая); 4) DECT; 5) UWC-136.

Наибольшее распространение в мире получили два стандарта: UMTS (или WCDMA) и CDMA2000 (IMT-МС), в основе которых лежит одна и та же технология – CDMA (Code Division Multiple Access – множественный доступ с кодовым разделением каналов).

Технология CDMA2000 обеспечивает эволюционный переход от узкополосных систем с кодовым разделением каналов IS-95 к системам CDMA «третьего поколения».

Технология UMTS разработана для модернизации сетей GSM (европейского стандарта сотовой связи второго поколения).

Работа по стандартизации UMTS координируется международной группой 3GPP (Third Generation Partnership Project), а по стандартизации CDMA2000 – международной группой 3GPP2 (Third Generation Partnership Project 2), созданными и сосуществующими в рамках ITU.

В сетях 3G обеспечивается предоставление двух базовых услуг: передача данных и передача голоса.

Разработка 3G началось еще в 1985 году. Появлялось множество планов и разработок, направленных на поиск наилучшей технологии 3G. В 1998 году организации по стандартизации из Европы, США, Японии и Кореи объединились в союз «Программа партнерства третьего поколения» (3G Partnership Project, 3GPP) для продвижения Wideband CDMA (WCDMA).

Далее версия 3G от 3GPP – UMTS (Universal Mobile Telephone System) – стала доминирующим мировым стандартом. Почти все лицензии 3G, выданные в мире на сегодняшний день, относятся к технологии WCDMA.

01.01.2013, Внешний интернет-шлюз Беларуси расширен до 350 Гбит/с. За полгода интернет-шлюз расширяли трижды: 250 – 280 – 350.



После появления 3G операторы мобильной связи получили возможность предоставлять качественный доступ в Интернет по более низким, чем ранее, ценам. В результате в мире количество абонентов мобильной связи заметно превысило количество абонентов связи стационарной (рисунок. 1).

Проследить развитие 3G-интернета в Беларуси можно путём сравнения характеристик стоимость/скорость с 2008 (до запуска 3G в Беларуси) до нынешнего 2013 года (таблицы 1 и 2).

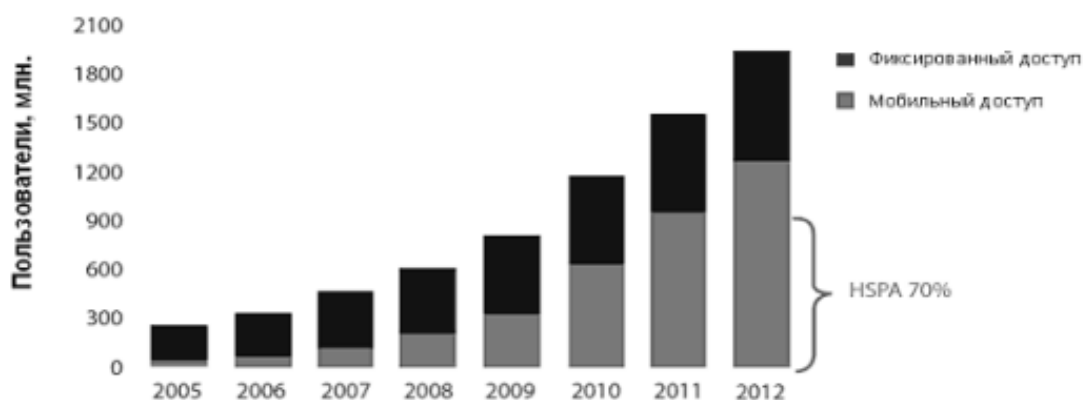


Рисунок 1 – Динамика роста количества пользователей мобильного интернета

Таблица 1 – Динамика стоимости и скорости мобильного интернета в 2008–2013 годах на основе тарифа Life: интернет 100

Год	Стоимость 1 мб оплаченного трафика, руб.	Стоимость 1 мб трафика сверх оплаченного, руб.	Скорость
2008	48	-	356 Кбит/с
2009	100	200	2,4–3,6 мбит/с
2010	100	200	3,6 мбит/с
2011	132	294	3,6–7,2 мбит/с
2012	132	294	7,2 мбит/с
2013	158	350	7,2 мбит/с

Таблица 2 – Динамика стоимости и скорости мобильного интернета в 2008–2013 годах на основе тарифа Velcom: Web 500

Год	Стоимость 1 мб оплаченного трафика, руб.	Стоимость 1 мб трафика сверх оплаченного, руб.	Скорость
2008	45	-	356 Кбит/с
2009	100	208	2,4–3,6 мбит/с
2010	100	208	3,6 мбит/с
2011	130	299	3,6–7,2 мбит/с
2012	130	299	7,2 мбит/с
2013	163	374	7,2 мбит/с

Стоимость 1мб трафика растёт в меньшей степени, чем скорость. С учётом того, что инфляция в РБ за данный период составила около 240 %, можно сделать вывод, что стоимость 1 мб трафика снизилась более чем вдвое, при увеличении скорости передачи данных в 2 раза.

Анализ 96 развитых и развивающихся рынков показал, что переход пользователей с 2G на 3G положительно влияет на экономику: 10 %-ый рост охвата сетей 3G увеличивает ВВП на 0,15 % на душу населения. Об этом говорится в докладе GSMA и Deloitte. Расширение зон покрытия сетей 3G в сочетании с быстрым увеличением числа устройств с функцией ввода данных и подключением к Интернету привело к тому, что объем данных, передаваемых через мобильные устройства, увеличился колоссально. В каждой из 96 стран в период с 2005 по 2010 годы объем данных, передаваемых с мобильных устройств, в среднем ежегодно возрастал более чем вдвое, а в государствах Западной Европы и вовсе на 350 %.

На основе прогноза Cisco о развитии сетевых технологий в 14 странах выведена прямая зависимость между передачей данных с использованием сетей 3G и ростом экономики. Удвоение объема данных, передаваемых через мобильные устройства, привело к увеличению ВВП в расчете на душу населения на 0,5 %.

### Литература

1. Новостной сайт интернет-провайдеров Республики Беларусь / Электронный ресурс: <http://providers.by/flag/3g/> – [Дата доступа 16.01.2013].
2. Официальный сайт Velcom в Республике Беларусь / Электронный ресурс: <http://www.velcom.by> – [Дата доступа 19.01.2013].
3. Официальный сайт Life в Республике Беларусь / Электронный ресурс: <http://life.com.by/> – [Дата доступа 19.01.2013].
4. Интернет без проводов: описание технологий GPRS и 3G / Электронный ресурс: <http://price.od.ua/articles.phtml?id=70> – [Дата доступа 19.01.2013].
5. Беларусь расширила внешний интернет-шлюз до 350 Гбит/с / Электронный ресурс: <http://it.tut.by/328670> – [Дата доступа 22.01.2013].

**А.В. Купреева (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

### ТОП БРАУЗЕРОВ ДЛЯ РАЗЛИЧНЫХ ОС

*Google Chrome* – браузер, разрабатываемый компанией Google на основе браузера с открытым исходным кодом Chromium. Использует

движок WebKit. Самый молодой из всех участников рейтинга. Первый стабильный релиз вышел в декабре 2008 года.

Этот браузер прост в использовании, имеет множество модулей расширения и настроек, отвечает практически всем требованиям современного пользователя, отлично работает с JavaScript, а эту технологию широко используют популярные социальные сети – Twitter, Facebook, YouTube.

К плюсам Google Chrome можно отнести:

- избегает заражение вредоносными программами;
- не нужно перезапускать браузер, чтобы закрыть все зависшие страницы.

Минусами же стали:

- аскетичный дизайн придется по нраву не каждому;
- нет интегрированного механизма чтения каналов RSS.

*Mozilla Firefox* – свободный браузер от компании Mozilla Corporation. Дата первого выпуска — ноябрь 2004 года. Firefox – много платформенный интернет-обозреватель поддерживающий огромное количество расширений. В браузере используется движок Gecko.

Firefox пока лидирует по качеству и числу плагинов. Браузеру уже более десяти лет, но он всё ещё активно развивается и обновляется.

Преимущества Mozilla Firefox:

- большое количество доступных настроек;
- высокий уровень производительности обработки HTML5.

К минусам можно отнести:

- излишнее число настроек замедляет работу браузера;
- довольно медленная загрузка страниц.

*Internet Explorer (IE)* – браузер от компании Microsoft. Поставляется с 1995 года в комплекте с Windows. Долгое время был самым популярным браузером в мире. Особой популярности достиг в 2002–2003 годах с Windows XP (около 95 %). В настоящее время теряет свои позиции, уступая место Chrome и Firefox.

Преимущества Internet Explorer:

- хорошие функции безопасности;
- оперативная обработка HTML-кода.

Недостатки:

- нет автоматической синхронизации закладок;
- довольно сложное управление вкладками;
- нет тем;
- не поддерживает плагины;
- самым главным минусом является некорректное отображение

страниц. Бывает, что создатели сайтов делают отдельную версию дизайна специально для IE.

*Opera* – веб-браузер, выпускаемый компанией Opera Software. Вышел на рынок в 1995 году. Достаточно популярный до недавнего времени.

Вместе с браузером идет большое количество настроек и функций. Причем не смотря на навороченность, он работает довольно плавно и быстро.

Opera имеет ряд преимуществ:

- есть уникальная возможность для расширения функционала браузера с помощью виджетов;
- быстрая загрузка страниц;
- предупреждает об опасных сайтах и поддерживает приватность.

К недостаткам же можно отнести:

- настройки систем безопасности немного хуже, чем у других.

*Safari* – браузер, разработанный компанией Apple, и входящий в состав операционных систем Mac OS и iOS. Работает на свободно распространяемом движке WebKit. Появился после прекращения выпуска Internet Explorer для Mac.

Safari, который установлен по умолчанию на всей продукции компании Apple. Унаследовав эффекты iOS. Safari – понятный и адекватный браузер, но во многом уступает другим.

Преимущества браузера:

- устойчивая сеть безопасности;
- мобильная синхронизация для пользователей на базе iOS.

Недостатки:

- мало функций и плагинов;
- очень медленное открытие страниц на основе JavaScript или HTML5.

Рассмотрим рейтинг браузеров по популярности за конец 2012 года по всему миру (рисунок 1).



Рисунок 1 – Рейтинг браузеров — карта покрытия

New Relic выяснила рейтинг популярности браузеров на различных операционных системах (09.11.2012).

### **Windows**

Самый популярный браузер для Windows, оказался и занял первое место, Google Chrome (все версии – 34 %). Второе место занимает Opera, на её долю пришлось 31 %. Третье место достается Mozilla Firefox (24 %). Microsoft Internet Explorer занимает четвёртое место с 6 %. На пятом месте обосновался Safari с 2 % и некоторые другие браузеры.

### **Mac**

Самыми популярными веб-обозревателями для Mac является – Safari, которому принадлежит 61 %. На втором месте – Google Chrome, с долей 21 %. На третьем – Mozilla Firefox, который занял 18 %. Четвёртое место досталось Opera с 11 %.

### **Linux**

Среди обладателей ОС Linux самыми распространёнными в использовании браузерами являются Mozilla Firefox и Google Chrome.

### **Мобильные браузеры**

Самым популярным мобильным браузером является Opera Mini – самый актуальный браузер для мобильных телефонов, во многом благодаря тому, что специальная технология позволяет сжимать данные, а, следовательно, страницы загружаются быстрее. По данным февральских исследований компании StatCounter встроенный в операционную систему мобильный браузер Android наконец опередил Opera Mini по популярности. Также одним из лидеров является мобильный браузер Safari.

**А.В. Лебедева, Я.В. Михайлич (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **С.В. Карпенко**, доцент, к.э.н.

## **ПРОГРАММНАЯ ПОДДЕРЖКА ДИСТАНЦИОННОГО ОБУЧЕНИЯ**

Одним из условий успешного внедрения дистанционного обучения является правильный выбор программного обеспечения, соответствующего конкретным требованиям, которые определяются потребностями обучаемого, преподавателя и администратора, контролирующего ход, установку и настройку программного обеспечения.

Во всем многообразии средств организации электронного обучения можно обратить внимание на следующие группы:

1) **авторские программные продукты**, разработанные для преодоления затруднений, с которыми сталкиваются преподаватели при использовании языков программирования и позволяют самостоятельно разрабатывать учебный контент;

2) **системы управления обучением (LMS)** предназначены для контроля большого числа обучаемых;

3) **системы управления контентом (CMS)** имеют возможность создавать каталоги графических, звуковых, видео- и текстовых файлов и манипулировать ими;

4) **системы управления обучением и учебным контентом (LCMS)** следят за обучением большого количества людей, создают учебные материалы, а также хранят и находят отдельные элементы контента.

При выборе соответствующего программного продукта необходимо учитывать 5 потребительских характеристик, важных для достижения эффективности обучения.: надежность в эксплуатации, совместимость, удобство использования, модульность и обеспечение доступа.

На основе анализа можно выделить несколько наиболее распространенных систем дистанционного обучения, которые представляют собой коммерческие разработки, ориентированные на использование в рамках учебного заведения. Основные критерии отбора – степень поддержки системы и многоязыковое сопровождение.

«**Доцент**». Система ДО «Доцент» (сокращение от Дистанционный Образовательный ЦЕНТр) была разработана компанией «УНИАР». Для клиентов система может поставляться в трех основных комплектациях, которые различаются количеством дополнительных модулей и средств разработки. Минимальная комплектация включает в себя основной блок, состоящий из автоматизированных рабочих мест: «Администратор», «Куратор», «Преподаватель» и «Слушатель», а также конструктор тестов и обучающего курса. В стандартной комплектации поставляется конструктор курсов, необходимый набор Java-классов, а также дополнительный обучающий курс и документация.

Цена установки стандартного комплекта на 1 сервер составляет 6000-12000 долл. для корпоративных заказчиков, также действует специальная система скидок для государственных вузов. Успешный опыт внедрения СДО в ВУЗах РФ – Российский химико-технологический университет им. Д.И. Менделеева, Московский инженерно-физический институт, Ивановский химико-технологический университет и др.

Особенность этой системы заключается в том, что конструктор курсов и тестов «Униар Билдер 2002» позволяет создавать индивидуальные и групповые тесты заданной сложности с использованием широкого

перечня типовых тестов (16 шаблонов). Однако выделяют и недостатки – довольно сложный интерфейс для непрофессиональных пользователей, ограниченная масштабируемость и недостаточная функциональность системы, привязка к продуктам Microsoft.

**«Прометей».** Выпускается негосударственным образовательным учреждением «Институт виртуальных технологий в образовании» с 1997 г. и имеет модульную архитектуру, что позволяет расширять и модернизировать систему по мере необходимости, управлять деятельностью виртуального учебного заведения.

Для корпоративных заказчиков СДО поставляется по цене в 8400 долл., для тренинг-центров предусмотрены скидки. Для учебных заведений (школ, вузов) предусмотрена специальная льготная цена в 4800 долл. Успешный опыт внедрения СДО "Прометей" в МЭСИ, МГТУ имени Н.Э. Баумана, ОмГТУ, МАИ и др. Значительным минусом в использовании является привязка к продуктам Microsoft и недостаточная масштабируемость.

**WebTutor.** Компания WebSoft предлагает свой продукт как средство корпоративного обучения персонала, в котором используется система управления контентом Web Builder. В качестве платформы используется Lotus Domino R5, который имеет в своем составе веб-сервер и сохраняет данные во встроенном формате Lotus NS F. Система поддерживает международные стандарты обмена учебными материалами (SCORM, AICC), имеет возможность построения на основе ПО WebTutor распределенной системы дистанционного обучения для компаний с филиальной сетью любого масштаба.

В отличие от большинства подобных систем, стоимость программного продукта WebTutor не зависит от количества пользователей и приобрести его можно по цене в 5500 долл. Есть опыт более 350 внедрений системы WebTutor в крупнейших компаниях, организациях и ВУЗах России (Финансовая Академия при Правительстве РФ) и СНГ. К существенному недостатку относится функционирование системы только при использовании платформы Windows.

**eLearning Server.** Первая версия программного продукта eLearning Server была выпущена компанией «ГиперМетод» в 2001 г. Система позволяет создавать собственные учебные центры в Интернете, интранете и организовывать полный цикл дистанционного обучения (управление расписанием, сертификацией знаний учащихся, электронной ведомостью успеваемости и электронной библиотекой).

Пакет eLearning состоит из двух основных продуктов: eLearning Office и eLearning Server с ценами 1799 долл. для ВУЗов и 2099 долл.

для корпоративных заказчиков. Имеется успешный опыт внедрения СДО в 32 ВУЗах РФ (СПбГУЭФ, ВГПУ, МГИМО, НГПУ и др.). Версия для ВУЗов, помимо функциональности ядра системы, поддерживает дополнительные свойства – например, формирование планов специальностей и направлений, групповой перевод, зачисление, отчисление и т. п. Недостаток – относительная сложность доработки программного средства.

Таким образом, сетевое образование продолжает развиваться – компании, производящие программные продукты для дистанционного образования, активно конкурируют на рынке, демонстрируя преимущества технологий новых и перспективных разработок экспериментальных систем. Однако, несмотря на имеющийся потенциал, в современных условиях сетевое обучение имеет немаловажную проблему – не представляется возможным массовое использование данных систем отечественными вузами из-за их высокой стоимости и необходимости продления лицензии на каждый учебный год; коммерческие же системы предоставляют ограниченное количество образовательных лицензий и недостаток расширения и масштабирования возможностей.

Выгода состоит в том, что сетевое программное обеспечение, один раз установленное и обслуживаемое в одном месте, может использоваться по всему миру тысячами учащихся, имеющих компьютер и доступ в Интернет. Отметим, что системы с открытым кодом, такие как ATutor, LAMS, Moodle, OpenACS, Sakai позволяют решать те же задачи, что и коммерческие системы, но при этом у пользователей есть возможность доработки и адаптации конкретной системы к своим потребностям с существенно меньшими затратами и большей эффективностью. Многообразие различных систем дистанционного образования крайне важно, так как одна система, разработанная для одного класса пользователей (например, с одинаковым складом ума) может совсем не подойти другим обучаемым.

**С.В. Леванцов (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

## **ПЛЮСЫ И МИНУСЫ ТЕХНОЛОГИИ IPoE**

Существует множество технологий доступа в Интернет для конечных абонентов. На нашей территории особенно популярны две: РРТР и РРРоЕ. В обоих случаях создается РРР-туннель, производится аутентификация, и внутри туннеля ходит абонентский IP-трафик. Основное



отличие этих протоколов – они работают на разных уровнях сетевой модели OSI. PPPoE работает на втором (канальном) уровне, добавляя специальные теги, идентифицирующие конкретный туннель, в Ethernet-фреймы. PPTP работает на третьем (сетевом) уровне, упаковывая IP-пакеты в GRE.

Сейчас провайдеры начали продвигать новую технологию под названием IPoE.

IPoE принципиально отличается от PPTP и PPPoE. Сам термин придуман, скорее всего, в России и является абстрактным. Означает он следующее: IP over Ethernet. Смысл именно такой, как и расшифровка – IP-трафик поверх Ethernet. Контроль доступа может осуществляться при помощи привязок IP-MAC на коммутаторах доступа или на BRAS или выделения VLAN на каждого абонента.

Для настройки IPoE нужен минимум действий – все настройки на сетевой карте компьютера или роутере должны находиться в положении «получить автоматически». Доступ в сеть Интернет появляется на компьютере или любом другом сетевом устройстве сразу же после получения им IP-адреса от DHCP-сервера.

На рисунке 1 представлена схема работы данной технологии.

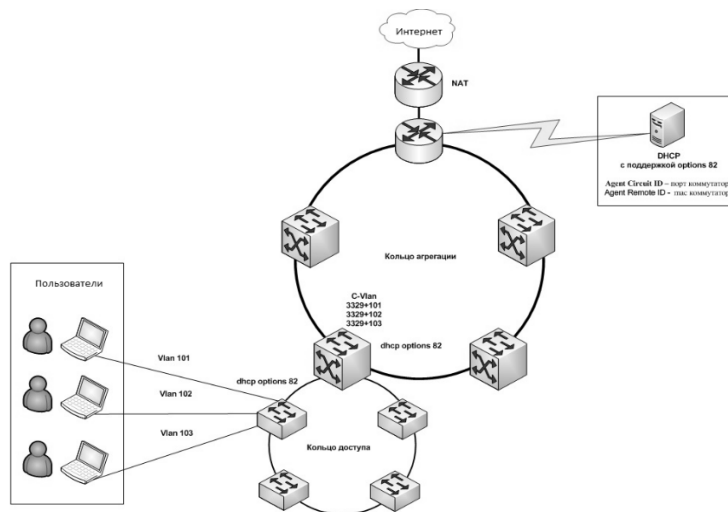


Рисунок 1 – Схема работы технологии IPoE

Рассмотрим настройку интерфейсов маршрутизатора на примере ISP «Гарант».

Для того, чтобы изменить тип подключения, Вам необходимо зайти в «Личный кабинет» и нажать кнопку «перейти на IPoE». Далее, вместо Russia PPTP (Dual Access) (варианты «PPTP», «L2TP») нужно выбрать тип подключения «Dynamic IP». Осталось лишь сохранить настройки.

Минусы технологии:

На данную технологию нет RFC.

Многое дешевое оборудование, использующееся на доступе, не всегда корректно работает с options 82.

И самый главный минус – безопасность конечных пользователей. Она не обеспечивается.

**А.В. Левченко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Д. Левчук**, к.т.н., доцент

## **РАЗРАБОТКА ПЛАГИНА НА ПЛАТФОРМЕ GOOGLE ДЛЯ ИНДИВИДУАЛИЗАЦИИ ЗАДАНИЙ**

Платформы социальных сетей были разработаны специально с целью удовлетворения потребностей исследователей для сотрудничества и доступа к информации и знаниям. Каждая платформа используется определенным образом и предлагает разнообразные функции. Инструменты Google заняли достойное место в оптимизации взаимодействия ученых-исследователей. В этом аспекте рассматривается задача по автоматизации процессов индивидуализации заданий (исследовательских и учебных) при помощи Google Apps Script.

Для решения данной задачи рассматривается группа в Google Groups, для каждого члена которой надо создать структуру папок с корневой папкой в GDrive. Каждый член группы сможет получить доступ только к своей папке. Также выделяется отдельный список пользователей, которые будут иметь доступ ко всем папкам созданной структуры. Для размещения скрипта используется Google Sites, с которого размещена ссылка на скрипт.

Пользовательский интерфейс представляет собой набор полей, которые необходимо заполнить:

1. Название новой корневой папки,
2. Доступ (Через ","),
3. Название вложенной папки,
4. Имя группы.

Пользовательский интерфейс создается с помощью вызова метода `UiApp.createApplication().setTitle('Folder Utils')` в `doGet` функции.

Обработка происходит в `folderUtilsServerHandler` (функция, которой передаются значения пользовательского ввода).

Создание папок происходит с помощью Google Doc API.

Таким образом, Google Apps Script – это инструмент, который предоставляет быстрый и удобный интерфейс интеграции всех инструментов платформы Google посредством JavaScript.

**А.М. Леоненко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

## **ОПИСАНИЕ ERP-СИСТЕМЫ SAP R/3 И ПРИМЕРЫ ЕЁ ПРИМЕНЕНИЯ НА БЕЛОРУССКИХ ПРЕДПРИЯТИЯХ**

Сначала следует сказать несколько слов о разработчике данной ERP-системы – компании SAP AG. SAP AG — немецкая компания, производитель программного обеспечения для организаций. Штаб-квартира расположена в Вальдорфе. Самый известный продукт компании — ERP-система SAP R/3, ориентированная на крупные и средние предприятия, разрабатываемая и продаваемая компанией с начала 1990-х годов. R/3 создана в продолжение линеек RF (позднее идентифицированной как R/1) и R/2. Начиная с выпусков середины 2000-х годов название R/3 не используется, ядро ERP-системы, созданной в продолжение линейки, производитель называет SAP ERP ECC.

Теперь рассмотрим общие концепции системы SAP R/3.

Решения – это подход к программному обеспечению, направленный на внешний мир, ориентированный на бизнес и на пользователя. Компоненты – это подход к программному обеспечению, направленный внутрь, технически ориентированный и относящийся к организационным представлениям SAP. Компоненты являются не целыми бизнес – решениями, а их составными частями. Деловой опыт, стратегия и ноу-хау являются нематериальными активами, которые в сочетании с огромным потенциалом гибкости, интеграции и настройки, присущим программному обеспечению SAP, становятся мощными отраслевыми и межотраслевыми решениями электронного бизнеса.

Платформа электронного бизнеса mySAP.com включает полное семейство решений (рисунок 1), сервисов и технологий электронного бизнеса, объединенных потребностями основных областей:

### **Межотраслевые решения:**

Наиболее комплексный из доступных в настоящее время на рынке набор бизнес-решений, позволяющий бизнесу получить реальное конкурентное преимущество.

### **Отраслевые решения:**

Отраслевые решения, построенные на базе межотраслевых, предоставляют специфическую функциональность для 21 отрасли.

### **Инфраструктура и сервисы:**

Технология и сервисы поддержки межотраслевых и отраслевых решений, обеспечивающие их быстрое внедрение и бесперебойное использование.

<p><b>Межотраслевые решения</b></p> <ul style="list-style-type: none"> <li>● mySAP Enterprise Portals</li> <li>● mySAP CRM</li> <li>● mySAP SCM</li> <li>● mySAP Exchanges*</li> <li>● mySAP SRM</li> <li>● mySAP BI</li> <li>● mySAP PLM</li> <li>● mySAP HR</li> <li>● mySAP Financials</li> <li>● mySAP Mobile Business</li> </ul>	<p><b>Отраслевые решения</b></p> <ul style="list-style-type: none"> <li>● mySAP Aerospace &amp; Defense</li> <li>● mySAP Automotive</li> <li>● mySAP Banking</li> <li>● mySAP Chemicals</li> <li>● mySAP Consumer Products</li> <li>● mySAP Engineering &amp; Construction</li> <li>● mySAP Financial Services</li> <li>● mySAP Healthcare</li> <li>● mySAP High Tech</li> <li>● mySAP Higher Education &amp; Research</li> <li>● mySAP Insurance</li> <li>● mySAP Media</li> <li>● mySAP Mill Products</li> <li>● mySAP Mining</li> <li>● mySAP Oil &amp; Gas</li> <li>● mySAP Pharmaceuticals</li> <li>● mySAP Public Sector</li> <li>● mySAP Retail</li> <li>● mySAP Service Providers</li> <li>● mySAP Telecommunications</li> <li>● mySAP Utilities</li> </ul>
<p><b>Инфраструктура и сервисы</b></p> <ul style="list-style-type: none"> <li>● mySAP Technology</li> <li>● mySAP Services</li> <li>● mySAP Hosted Solutions</li> </ul>	

Рисунок 1 – Решения платформы электронного бизнеса meSAP.com

Структура предприятия отображается в приложениях SAP посредством организационных элементов (рисунок 2). Организационные элементы представляют собой механизм определения структуры предприятия в SAP-системе в юридических и/или коммерческих целях. Организационные элементы включают юридические лица компании, заводы, склады, офисы продаж, а также места возникновения прибыли и могут быть присвоены одному приложению (например, сбытовая организация присваивается сбыту) или нескольким приложениям (например, завод присваивается управлению материальными потоками и планированию производства).

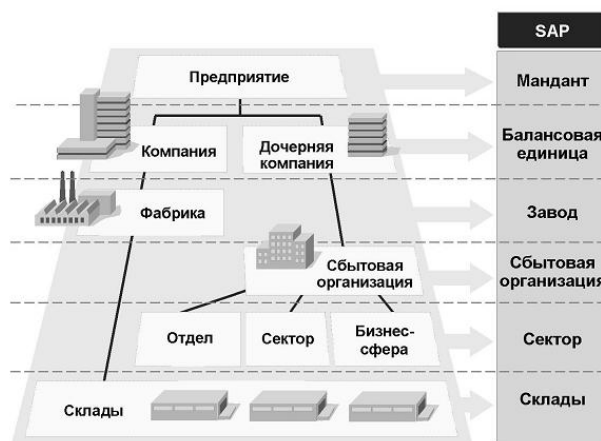


Рисунок 2 – Организационная структура: терминология

Платформа mySAP.com предоставляет компаниям решения и услуги, необходимые для упразднения границ компаний и выхода на глобальные торговые площадки – от порталов, основанных на ролях, до решений для мобильного бизнеса. Открытые и гибкие решения mySAP.com поддерживают базы данных, приложения, операционные системы и аппаратное обеспечение практически всех основных поставщиков.

В 1992 году был открыт офис SAP AG в городе Москве, Россия, а затем создана компания ООО «САП СНГ», представительства которой открыты в крупных городах России, а также в Алматы (Казахстан), Киеве (Украина) и Минске (Беларусь).

Рынок Беларуси стратегически важен для бизнеса SAP. Первый проект SAP с одним из лидеров белорусской промышленности – Белорусским металлургическим заводом (БМЗ) – был реализован еще в 90-ые годы и стал одним из первых проектов на территории стран СНГ. Важным фактором является также то, что в Беларуси сосредоточены значительные ресурсы разработчиков и консультантов SAP-интеграторов.

Белорусские ресурсы активно привлекаются для работы в проектах SAP по всему миру. Сегодня партнерская сеть в Беларуси включает шесть активно работающих партнеров SAP, среди которых две компании «Группы ИВА». Беларусь входит в Топ-3 крупнейших потребителей лицензий программного обеспечения SAP в странах СНГ по результатам 2010 года.

Ключевыми клиентами SAP в Беларуси являются Атлант-М, Белаз, Беларусбанк, Беларуснефть, БелЖД, БелТрансГаз, БелЭнерго, БМЗ, Мозырский НПЗ, Государственный таможенный комитет РБ, Velcom и другие предприятия, являющиеся основой экономики страны. «Это очень показательно в том плане, насколько высоко стремление белорусских компаний использовать в своей работе мировые практики и модернизировать свои системы управления и принятия решений, чтобы сделать собственную экономику конкурентноспособной и эффективной», – отмечает Илья Юрьев, назначенный в мае 2011 года директором представительства «САП СНГ» в Республике Беларусь.

**К.С. Марченко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Ю.С. Митченко**, ассистент

## **ЯЗЫК ЗАПРОСОВ В 1С ПРЕДПРИЯТИЕ 8.2**

Программирование 1С состоит не только из написания программы. 1С это слиток действий пользователя и данных с которыми он работает.

Данные хранятся в базе данных. Запросы 1С – это способ доставать данные из базы данных для того, чтобы показать пользователю в форме или чтобы обработать их.

Основополагающая часть отчета – это запрос 1С. В случае отчета СКД – это большая часть отчета.

Для выполнения запроса необходимо составить текст на специальном языке запросов, который сильно отличается от обычного языка программирования по синтаксису и назначению.

В тексте запроса присутствуют следующие секции:

- объявление переменных запроса (ссылки на атрибуты или реквизиты объектов);
- описание функций агрегирования, значения которых определяются на основании переменных запроса;
- описание группировок запроса (по каким переменным необходимо сгруппировать данные и как их упорядочить).

В листинге 1 приведен образец запроса.

Листинг 1 – Пример построения запроса

```
Запрос = Новый Запрос();
```

```
Запрос.Текст = "ВЫБРАТЬ
```

```
  | Номенклатура.Ссылка
```

```
  | ИЗ
```

```
  | Справочник.Номенклатура КАК Номенклатура
```

```
  | ГДЕ
```

```
  | Номенклатура.Услуга";
```

```
Выборка = Запрос.Выполнить().Выбрать();
```

```
Пока Выборка.Следующий() Цикл
```

```
  Сообщить(Выборка.Ссылка);
```

После выполнения запроса 1С приходит результат и нужно его обработать.

Результат можно выгрузить в обычную таблицу. На листинге 2 представлен пример выгрузки результата в таблицу :

Листинг 2 – Выгрузка базы в таблицу

```
Выборка = Запрос.Выполнить().Выгрузить(); //Результат – таблица значений
```

Чтобы не писать текст запроса руками – существует конструктор запросов 1С. Для этого нужно нажать правой кнопкой мыши в любом месте модуля и выбрать Конструктор запроса 1С. Выбрать в конструкторе запросов 1С нужную таблицу слева и перетащить правее. Выбрать в конструкторе запросов 1С из таблицы нужные поля и перетащить правее.

На закладке Условия в конструкторе запросов 1С можно выбрать нужные отборы таким же способом (перетаскивая поля, по которым нужно делать отбор) указать верное условие. На закладке Порядок – указывается сортировка. На закладке Итоги – суммирование итогов.

С помощью конструктора запросов 1С Вы можно изучить любой существующий запрос. Для этого требуется нажать правой кнопкой на текст существующего запроса и выбрать также конструктор запросов 1С – и запрос будет открыт в конструкторе запросов 1С.

**О.И. Микулинская (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Н.А. Алешкевич**, к. ф.-м. н., доцент

## **О РАЗРАБОТКЕ СИСТЕМЫ МЕТРОЛОГИЧЕСКОГО УЧЕТА ТНПА И СРЕДСТВ ИЗМЕРЕНИЙ**

В настоящее время актуальными являются задачи метрологического учета и систематизации, имеющих на предприятиях и в организациях технических нормативных правовых актов (ТНПА), а так же осуществление метрологического надзора за эксплуатацией средств измерений (СИ). Для обеспечения эффективного функционирования службы стандартизации и метрологического обеспечения университета возникла необходимость создания автоматизированной системы учета ТНПА и СИ университета.

На отечественном рынке уже имеется ряд программных продуктов, позволяющих оказать существенную помощь в автоматизации учета, контроля ТНПА и СИ. Однако они адаптированы под большие предприятия, достаточно дорогие и требуют периодической актуализации. Поэтому разработка своего программного продукта для автоматизации метрологических работ является весьма актуальной задачей.

Компьютерная система учета будет предначинана в первую очередь для оптимизации работы кабинета стандартизации. В настоящее время в кабинете стандартизации имеется более 3000 наименований ТНПА учебной и справочной литературы и периодических изданий в области метрологии, стандартизации, сертификации и управления качеством. Кроме того, в университете насчитывается порядка 2000 единиц измерительных приборов и оборудования, списки которого должны ежегодно обновляться, а СИ, используемые в научных целях, поверяются в соответствии с графиком. Автоматизация процессов учета СИ существенно упростит работу службы стандартизации и метрологического учета.

Разработку автоматизированной системы метрологического учета ТНПА и средств измерений целесообразно вести в СУБД MS Access, так как данная СУБД является общедоступной. Данные можно импортировать из разных источников, таких как MS Excel, Postgresql, DB2 и т.д.

В рамках разработки автоматизированной системы учета ТНПА и средств измерений была представлена схема данных, которая отображает структурную зависимость данных и обеспечивает быстрое действие работы БД. Все данные хранятся в таблицах, в которых имеются поля с определенными типами данных, поиск осуществляется с помощью запросов SQL (Structure query language) и выводится в виде отчета на форму. Так же данные можно экспортировать в MS Excel, MS Word и др.

Ю.С. Митченко (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. В.Д. Левчук, к.т.н., доцент

## ОРГАНИЗАЦИЯ ЕДИНОВРЕМЕННЫХ ВЫПЛАТ СТУДЕНТАМ В ПОДСИСТЕМЕ «1С:СТИПЕНДИЯ»

В подсистеме «Стипендия» разработан набор документов, которые позволяют начислять (удерживать) студентам определенные суммы с расчетом на длительный или краткосрочный период. С развитием различных технологий ручной ввод информации должен занимать минимально время, что в свою очередь позволит повысить удобство работы и качество труда.

Для организации ввода краткосрочных выплат был создан документ «Разовые начисления (удержания)» (рисунок 1). Здесь в шапке документа бухгалтер выбирает вид начисления (удержания), номер и дата документа ставятся автоматически. Бухгалтер может также заполнить такие поля как общая сумма, общий источник финансирования, счет. При автоматическом заполнении табличной части или при интерактивном вводе новой строки в табличную часть документа значения этих общих полей будут автоматически проставляться в табличную часть документа, что освобождает бухгалтера от однотипной работы по вводу одинаковых данных.

N	Студент	УчГр	Назн...	Сумма к ...	Источ. фин...	Счет
1	МАКАРЕВИЧ АНАСТАСИЯ ОЛЕГОВИ	ПС-31	дневн	500,000	ФМП (С-1)	240
2	ЛИТВИНОВА ЕКАТЕРИНА ВАСИЛЬЕ	П-33	дневн	500,000	ФМП (С-1)	240
3	ЖАРИТОНЕНКО ОЛЬГА ВАСИЛЬЕВН	А-22	дневн	500,000	ФМП (С-1)	240
4	ДИВИЗИОННЫЙ КИРИЛЛ ЛЕОНИД	А-52	дневн	500,000	ФМП (С-1)	240
5	ГАВРИЛЕНКО ДАРЬЯ ВЛАДИМИРОВ	БФ-51	дневн	500,000	ФМП (С-1)	240
6	ШВЕДОВА АННА СЕРГЕЕВНА	БФЖ-52	дневн	1,000,000	ФМП (С-1)	240
7	ТИХОНОВА АНАСТАСИЯ МИХАЙЛОВ	РФ-51	дневн	1,000,000	ФМП (С-1)	240

Рисунок 1 – Вид документа «Разовые начисления (удержания)»

Так как наша задача состоит в том, чтобы бухгалтер затрачивал меньше времени на ввод данных, то в данном документе были созданы ряд возможностей для автоматического ввода информации. По кнопке



«Заполнить» пользователь имеет ряд возможностей для заполнения документа. Нужно отметить, что при всех способах заполнения документа, реквизиты учебная группа и назначение проставляются автоматически при выборе студента и не доступны для редактирования. По реквизиту назначение можно определить обучается ли студент на контрактной или на бюджетной основе. Рассмотрим каждый из способов автоматического заполнения.

Во-первых, бухгалтер может заполнить табличную часть документа только студентами определенной учебной группы. После автоматического заполнения табличной части документа студентами, если не были заполнены общие реквизиты шапки документа, бухгалтер вводит необходимую сумму выплаты, выбирает источник финансирования и счет, причем счет проставляется автоматически при выборе источника, однако пользователь имеет возможность его корректировать.

Во-вторых, данный документ позволяет осуществить выплату студентам по определенным видам расчета. Рассмотрим этот механизм на примере. Бухгалтер начислил некоторым студентам определенную сумму по виду расчета «Детское пособие на 1-го ребенка» документом «Пособия». После проведения этих документов, данные попали в журнал расчетов. После сделанного начисления через пару дней бухгалтер хочет сформировать списки на банк и выплатить эти детские пособия. Для этого необходимо создать документ «Разовые начисления (удержания)», нажать на кнопку «Заполнить»→«Выбрать по виду расчета». В появившемся окне выбираем наш вид расчета «Детское пособие на 1-го ребенка» и нажимаем «ОК». После этого система выберет все записи в журнале расчета, в которых присутствует выбранный вид расчета и заполнит табличную часть документа, а в шапке в реквизите «Начисление/удержание» автоматически появится вид расчета «Межрасчетный период». Т.е. бухгалтеру не нужно помнить и выбирать каждого студента, которому была начислена данная выплата.

В-третьих, с помощью этого документа выполняется выплата всей стипендии студентам. Для этого необходимо нажать на кнопку «Заполнить»→«Заполнить на выплату стипендии». В данном действии заложен сложный механизм, который проводит подробный анализ записей журнала расчетов, осуществляет их выборку, анализирует данные по начислениям и удержаниям за общежития. После заполнения в шапке документа автоматически появится вид расчета «Выплата стипендии». Бухгалтеру надо лишь проверить суммы и провести документ.

Как видно из рисунка 1 документ имеет реквизит «Комментарий», в который бухгалтер может записывать свои пометки либо какую-либо

другую информацию, связанную с данным начислением (удержанием). Также на форме присутствует подсчет итоговой суммы к выплате.

В подсистеме «стипендия» присутствуют и такие виды расчетов, которые облагаются налогами и дополнительными удержаниями. При выплате таких начислений в нашем документе появляется еще одна колонка с суммой. В нее заносится та сумма, которая была назначена бухгалтером при начислении. А в колонке «Сумма к выплате» содержится сумма, которая получилась с учетом всех удержаний (рисунок 2).

N	Студент	УчГр	Назн...	Сумма к выпл...	Источ. фин...	Счет	Сумма
1	ЛАПУЖИН АРТЕМ АЛЕКСАНДРИ	ФК-12		528,000	ФМП (С-1)		600,000
2	КУХАРЕВ АРСЕНИЙ ОЛЕГОВИЧ	ФК-11		528,000	ФМП (С-1)		600,000
3	ЗАЙЧИКОВА ОЛЬГА ГЕННАДЬЕ	ФК-34	дневн	440,000	ФМП (С-1)		500,000

Комментарий: Выплата: Премия (с удерж.)

Всего сумма к выплате: 8 888 000    Всего сумма: 10 100 000

Рисунок 2 – Вид документа «Разовые начисления (удержания)» с двумя суммами

В данном документе также учтены возможности исправления ошибочно введенных данных (рисунок 3). Так пользователь может изменить сумму для всех выбранных студентов, либо источник финансирования, либо заново пересчитать сумму выплаты, получить итоги по источникам финансирования и т. д.

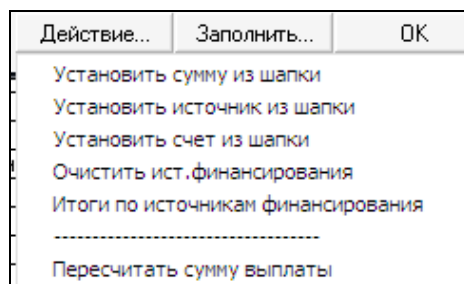


Рисунок 3 – Дополнительные возможности документа «Разовые начисления (удержания)»

**Ю.С. Митченко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Д. Левчук**, к.т.н., доцент

## НАЗНАЧЕНИЕ ОСНОВНЫХ НАЧИСЛЕНИЙ И УДЕРЖАНИЙ В ПОДСИСТЕМЕ «СТИПЕНДИЯ»

С развитием различных технологий ручной ввод информации на сегодняшний день должен занимать минимально время и не отвлекать

бухгалтера от его основных обязанностей, что в свою очередь позволит повысить удобство и качество труда.

В подсистеме «Стипендия» разработан набор документов, которые позволяют начислять (удерживать) студентам определенные суммы с расчетом на длительный или краткосрочный период.

Документ «Приказы длительного действия» (рисунок 1) позволяют назначить студенту персонально определенную выплату по конкретному виду расчета на длительный срок (либо удерживать определенную сумму на протяжении длительного периода). К таким начислениям можно отнести персональные надбавки, доплату к стипендии, дополнительные именные стипендии, расходы по найму жилья и т.д. В данном документе необходимо указать период выплат, студента, вид начисления (удержания), источник финансирования, счет. Здесь осуществлена возможность указывать как определенную сумму выплаты (удержания), так и процент от какой-либо суммы. Для удобства бухгалтеру автоматически при выборе студента показывается дополнительная информация по обучению выбранного студента в университете. Также бухгалтер имеет возможность делать свои пометки в специальное поле комментариев.

Постоянные начисления и удержания-Новый

№ док-та: 6630 от: 06.03.13 Выплачивать

Начисл./Удерж.: Надбавка к стипендии с: 01.03.13 по: 30.06.13

Студент: МИТЧЕНКО ЮЛИЯ СТАНИСЛАВОВНА Учеб. группа: МГ-ПМ(э)-21

Обучается с 01.09.11 по 30.06.13

Постоянный процент оплаты или же фикс. сумма

Колво (%): 40.00 Фикс. сумма(от от. времени): (0р.)

Источник финансирования: бюд. п.213 Счет: 200

Комментарий: распоряжение № 274

OK Закрыть

Рисунок 1 – Внешний вид документа «Приказы длительного действия»

При проведении данного документа данные попадают в Журнал расчета «Стипендия» и после этого появляется возможность осуществить выплату по данному виду расчета. Так как в документе указывается период выплаты в несколько месяцев, то и в Журнал расчетов попадают записи, у которых указан период равный началу и окончанию каждого месяца, входящего в выбранный период. Также записи попадают в справочник «Приказы длительного действия», который подчинен справочнику «Обучение». Это обеспечивает более быстрый доступ к необходимым данным. Благодаря этому бухгалтер может в любой момент увидеть, какие долгосрочные выплаты (удержания) имеет студент, и имеет возможность корректировать дату окончания приказа, не используя журнал документов.

Бывают ситуации, когда длительные выплаты назначаются многим людям, и чтобы создать документ на каждого студента персонально уйдет много времени. Для того чтобы оптимизировать этот процесс, был разработан документ «Групповые начисления (удержания)», в котором мы можем выбрать множество студентов и назначить им постоянную выплату или удержание. Как и в предыдущем документе, мы должны сначала указать вид начисления (удержания), период выплаты, а также указать факультет. Так как наша задача состоит в том, чтобы бухгалтер затрачивал меньше времени на ввод данных, то в данном документе были созданы ряд возможностей для автоматического ввода информации. По кнопке «Заполнить» пользователь выбирает в разрезе каких данных ему нужно выбрать студентов: либо по определенному факультету, либо по учебной группе. После автоматического заполнения табличной части документа студентами, бухгалтер вводит либо процент начислений, либо определенную сумму. Здесь также присутствует возможность автоматического заполнения реквизитов документа. После того, как все основные данные введены, пользователь должен нажать на кнопку «Действия»→«Создать документы постоянного действия». Мы увидим в табличной части нашего документа в реквизите «Документ» ссылку на только что созданный документ постоянного действия (рисунок 2).

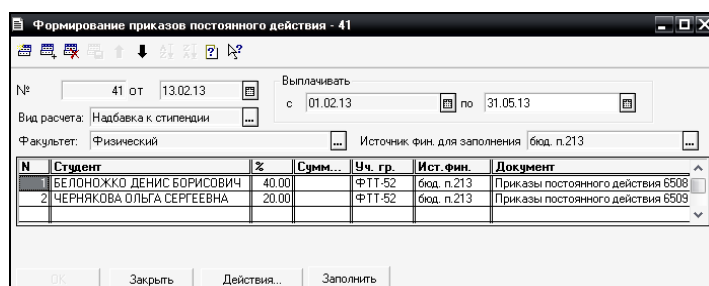


Рисунок 2 – Внешний вид документа «Групповые начисления (удержания)»

**А.В. Мостовцов (УО «ГГУ им. Ф. Скорины», Гомель)**  
 Науч. рук. **В.Н. Леванцов**, старший преподаватель

## **ПРОЕКТИРОВАНИЕ СИСТЕМЫ АВТОМАТИЗАЦИИ СОСТАВЛЕНИЯ РАСПИСАНИЯ ТЕЛЕВЕЩАНИЯ**

В процессе работы современных телекомпаний часто встает проблема составления расписания вещания. При этом необходимо учитывать

множество факторов, таких как класс телепрограммы, возрастная категория, длительность и многие другие. Несмотря на то, что задачи составления расписаний достаточно глубоко рассматривались отечественными и зарубежными учеными, и результаты их исследований достаточно полно изложены в литературе, на практике расписание составляют, как правило, вручную. Эффективность составления расписания зависит от большого количества факторов, а известные методы предлагают решение лишь частных задач, общее решение задач теории расписания отсутствует.

В свое время был сформулирован класс так называемых слабо формализованных задач (задачи, которые нельзя решить без участия эксперта в данной области) теории расписаний, для решения которых не найдено эффективных алгоритмов решения. Именно к таким задачам можно отнести большинство реальных задач производственного планирования.

При этом нужно учесть, что в последнее время появилось достаточно много мощных программных и аппаратных средств (языки программирования высокого уровня, современные персональные компьютеры), резко расширяющих возможности разработчиков методов решения задач календарного планирования. Для реализации автоматизированной системы была применена трёхуровневая архитектура (браузер, Web-сервер IIS, СУБД MS SQL Server) сбора и анализа данных. Применение данной архитектуры позволило обеспечить быстрое внедрение системы, а также, при необходимости, позволит быстро внедрять новые возможности и расширять существующие компоненты без необходимости каких-либо действий со стороны пользователей системы.

Разработанная автоматизированная система построена по модульной архитектуре, что позволяет не только расширять существующие компоненты (модули), но и внедрять новые. Данная возможность является очень важной для данной системы в силу специфики и непостоянства данных рассматриваемой предметной области.

Разработанная автоматизированная система позволяет быстро решать задачи импорта, анализа, составления, совмещения и экспорта расписания вещания, что, несомненно, повышает экономическую эффективность работы соответствующих отделов телекомпании. Данная автоматизированная система может быть применена в соответствующих отделах телекомпании любого уровня.

**Д.А. Навныко (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **Шаповалова Н.А.**, старший преподаватель

## **МИКРОКОНТРОЛЛЕРЫ**

Микроконтроллер – микросхема, предназначенная для управления электронными устройствами. Типичный микроконтроллер сочетает на одном кристалле функции процессора и периферийных устройств, содержит ОЗУ или ПЗУ. По сути, это однокристалльный компьютер, способный выполнять простые задачи.

С появлением однокристалльных микро-ЭВМ связывают начало эры массового применения компьютерной автоматизации в области управления. По-видимому, это обстоятельство и определило термин «контроллер».

В связи со спадом отечественного производства и возросшим импортом техники, в том числе вычислительной, термин «микроконтроллер» (МК) вытеснил из употребления ранее использовавшийся термин «однокристалльная микро-ЭВМ».

Первый патент на однокристалльную микро-ЭВМ был выдан в 1971 году инженерам М. Кочрену и Г. Буну, сотрудникам американской Texas Instruments. Именно они предложили на одном кристалле разместить не только процессор, но и память с устройствами ввода-вывода.

В 1976 году американская фирма Intel выпускает микроконтроллер i8048. В 1978 году фирма Motorola выпустила свой первый микроконтроллер MC6801, совместимый по системе команд с выпущенным ранее микропроцессором MC6800. Через 4 года, в 1980 году, Intel выпускает следующий микроконтроллер: i8051. Удачный набор периферийных устройств, возможность гибкого выбора внешней или внутренней программной памяти и приемлемая цена обеспечили этому микроконтроллеру успех на рынке. С точки зрения технологии микроконтроллер i8051 являлся для своего времени очень сложным изделием – в кристалле было использовано 128 тыс. транзисторов, что в 4 раза превышало количество транзисторов в 16-разрядном микропроцессоре i8086.

На сегодняшний день существует более 200 модификаций микроконтроллеров, совместимых с i8051, выпускаемых двумя десятками компаний, и большое количество микроконтроллеров других типов. Популярностью у разработчиков пользуются 8-битные микроконтроллеры PIC фирмы Microchip Technology и AVR фирмы Atmel, 16-битные MSP430 фирмы TI, а также 32-битные микроконтроллеры, архитектуры ARM, которую разрабатывает фирма ARM Limited и продаёт лицензии другим фирмам для их производства. Несмотря на

популярность в России микроконтроллеров упомянутых выше, по данным Gartner Group от 2009 года мировой рейтинг по объёму продаж выглядит иначе: первое место с большим отрывом занимает Renesas Electronics на втором Freescale, на третьем Samsung, затем идут Microchip и TI, далее все остальные.

Неполный список периферии, которая может присутствовать в микроконтроллерах, включает в себя:

- универсальные цифровые порты, которые можно настраивать как на ввод, так и на вывод;
- различные интерфейсы ввода-вывода, такие как UART, I<sup>2</sup>C, SPI, CAN, USB, IEEE 1394, Ethernet;
- аналого-цифровые и цифро-аналоговые преобразователи;
- компараторы;
- широтно-импульсные модуляторы;
- таймеры;
- контроллеры бесколлекторных двигателей;
- контроллеры дисплеев и клавиатур;
- радиочастотные приемники и передатчики;
- массивы встроенной флеш-памяти;
- встроенный тактовый генератор и сторожевой таймер;

Использование в современном микроконтроллере мощного вычислительного устройства с широкими возможностями, построенного на одной микросхеме вместо целого набора, снижает размеры, энергопотребление и стоимость построенных на его базе устройств.

В то время как 8-разрядные процессоры общего назначения полностью вытеснены более производительными моделями, 8-разрядные микроконтроллеры продолжают широко использоваться. Это объясняется тем, что существует большое количество применений, в которых не требуется высокая производительность, но важна низкая стоимость. В то же время, есть микроконтроллеры, обладающие большими вычислительными возможностями, например цифровые сигнальные процессоры.

### Литература

1. Бродин, В. Б. Системы на микроконтроллерах и БИС программируемой логики / В. Б. Бродин, А. В. Калинин. – М.: ЭКОМ, – 2002.
2. Микушин, А. В. Занимательно о микроконтроллерах / А. В. Микушин. – М.: БХВ-Петербург, – 2006.

**Н.А. Навныко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Н. Леванцов**, старший преподаватель

## **АВТОМАТИЗАЦИЯ РАБОТЫ ФИРМЫ ПО РЕМОНТУ БЫТОВОЙ ТЕХНИКИ**

В настоящее время автоматизируется практически каждая деятельность человека, начиная от добывающей промышленности и заканчивая управлением школьного учебного процесса. В данной работе предложена автоматизация работы фирмы, занимающейся ремонтом бытовой техники. Раньше, как и во многих других рабочих процессах, использовались различные журналы учета, электронные таблицы Excel, обновление которых требовало соответствующих знаний. Также ими совершенно неудобно пользоваться.

С помощью данного проекта появится возможность удобного просмотра услуг фирмы, списка сотрудников, изменение должностей, занесение новой информации в базу данных. Создание такой базы данных позволит вести удобное управление рабочим процессом фирмы.

Основные возможности разработанной базы данных представлены ниже:

- редактирование любой введенной информации с автоматическим изменением связанных данных;
- возможность ввода большого количества информации;
- удобное составление заказов на ремонт;
- внешняя поисковая система;

Для реализации проекта была использована система управления базами данных Microsoft Access 2003. Данная среда поставляется вместе с пакетом Microsoft Office, что делает ее общедоступной. А также удобный интерфейс Access позволяет реализовать поставленные задачи. Различные подсчеты и поиск информации осуществляется с помощью использования SQL-запросов. Для удобства ввода, изменения и просмотра данных использовались формы. Итоговая база данных включает 6 таблиц, 8 запросов, 7 форм, включая главную кнопочную форму, разработанную для удобства управления приложением.

В заключении стоит отметить, что разработанный проект может быть применен для автоматизации любой фирмы по ремонту бытовой техники, в том числе и узкоспециализированных фирм, таких, как фирмы по ремонту компьютеров, холодильников, телевизоров и т. п.



**И.А. Небышинец (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Н.А. Шаповалова**, старший преподаватель

## **ЭЛЕКТРОННАЯ ТОРГОВАЯ ПЛОЩАДКА**

Электронная торговая площадка (ЭТП) – комплекс информационных и технических решений, обеспечивающий взаимодействие покупателя (заказчика) с продавцом (поставщиком) через электронные каналы связи на всех этапах заключения сделки.

Электронная Торговая Площадка (ЭТП) позволяет объединить в одном информационном и торговом пространстве поставщиков и потребителей различных товаров и услуг и предоставляет участникам ЭТП ряд сервисов, повышающих эффективность их бизнеса. Электронной торговой площадкой сегодня можно назвать любой интернет-ресурс, посредством которого заключаются сделки купли-продажи между предприятиями – покупателями и продавцами. Заказчики получают возможность проводить электронные торги – тендеры, аукционы, запросы цен и предложений, – оптимизируя затраты, а поставщики – участвовать в проводимых закупках, размещать информацию о предлагаемой продукции и услугах. Иногда размещением торговых процедур занимаются специализированные компании, которые помимо размещения информации на торговой площадке обрабатывают полученный результат и даже, возможно, определяют победителя процедуры.

ЭТП для коммерческих заказчиков – это Электронные Торговые Площадки, на которых электронные торги проводят не государственные компании (коммерческие заказчики). Таких ЭТП значительно больше, чем для государственных торгов и регламент проведения электронных аукционов более гибкий (в некоторых аукционах не требуется даже электронная цифровая подпись). Например, Электронная торговая площадка Росатома – Аукционный Конкурсный Дом.

ЭТП для коммерческих поставщиков делятся на четыре типа:

Специализированные ЭТП, созданные под нужды определенного предприятия. Например: Электронная торговая площадка Газпрома по продаже нефтепродуктов

Многопрофильные Электронные Торговые Площадки, на которых представлен более широкий спектр продукции и услуг, чем на специализированных ЭТП. На таких площадках любая компания может выступать как в качестве заказчика, так и в качестве поставщика товаров и услуг без ограничений по номенклатуре.

Supplier-driven – торговые площадки, создаваемые и поддерживаемые продавцами. Этот вид ЭТП формируется крупными компаниями,

корпорациями и их объединениями, заинтересованных в каналах сбыта своей продукции современными способами через механизмы электронных торговых площадок.

Third-party-driven – торговые площадки, создаваемые и поддерживаемые третьей стороной. Это наиболее значительная категория посреднических площадок, призванных свести вместе покупателей и продавцов.

*Функции ЭТП.* Работая на Электронной Торговой Площадке, заказчик или поставщик может успешно решать различные вопросы, возникающие в повседневной деловой практике, потому что данные системы выполняют следующие важные функции: информационная функция позволяет ознакомиться с перечнем организаций, работающих на ЭТП, получить информацию по интересующей организации; функция маркетинга, которая позволяет осуществлять поиск покупателей и потребителей интересующих работ и услуг, а также получать информацию о потребностях и предложениях работ и услуг, которые размещают на площадке другие организации; рекламная функция. Разместив информацию о своей организации на ЭТП, сразу попасть в единое информационное пространство; торговая функция, которая позволяет в качестве организатора торгов осуществлять полный комплекс разнообразных торгово-закупочных мероприятий по приобретению товаров и услуг. В качестве участника торгов осуществлять комплекс действий для эффективной продажи собственных товаров и услуг; аналитическая функция позволяет проводить сравнительный анализ различных показателей деятельности организаций. Правильно выбрать контрагентов для выполнения поставок, работ и услуг по интересующей тематике; функция защиты информации, осуществляет безопасный электронный документооборот, построенный с использованием сертифицированных средств криптографической защиты информации (ЭЦП).

*Преимущества работы на ЭТП.* Любой руководитель отдела закупок или менеджер по закупкам сталкивался с тем, что быстро найти хорошего поставщика товара или услуги очень не просто. Даже не столь сложно найти поставщика, сколько определить, который предложит самые выгодные условия поставок. Преимущества работы на ЭТП для заказчика в решении этого вопроса очевидны:

- значительная экономия рабочего времени;
- экономия денежных средств на организации и проведении закупок;
- прозрачность и открытость процесса закупок;
- честная конкуренция, исключая работу недобросовестных сотрудников со «своими» фирмами-поставщиками;
- участие в торгах возможно из любой точки мира, не выходя из своего офиса;

– доступность для представителей любого бизнеса – цена и условия лота ничем не ограничены.

Преимущества работы на ЭТП для поставщика состоит в том, что каждый руководитель компании, руководитель отдела сбыта или продаж, всегда заинтересован в расширении клиентской базы и увеличении продаж компании. На рекламу тратятся колоссальные деньги, работают многочисленные call-центры, но в итоге, чем больше продает компания, тем больше она тратит лишних денег. При работе с ЭТП компания получает ряд преимуществ:

- быстрый поиск интересующих торгов;
- экономия средств на рекламной кампании;
- прозрачность и открытость процесса продаж;
- честная конкуренция, исключая неценовые методы ведения борьбы;
- равные права всех поставщиков товаров, работ и услуг;
- участие в торгах возможно из любой точки мира, не выходя из своего офиса.

**А.О. Пименов (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Е.А. Левчук**, к.т.н., доцент

## **ПОДКЛЮЧЕНИЕ И РАБОТА С HRM УСТРОЙСТВАМИ GARMIN**

Разработанная подсистема для анализа GPS и HRM данных поможет подвести итоги тренировочной деятельности, используя анализ тренировочных зон, времени, набора высоты, скорости прохождения, изменения сердечного ритма. Данная подсистема была разработана для сайта [www.olt-dagbok.net](http://www.olt-dagbok.net). Веб-приложение обеспечивает чтение данных с устройств Garmin и обработку данных: построение на интерактивной карте пройденного пути, построение графиков по различным параметрам.

Созданное приложение поддерживает TCX и FIT форматы хранения данных. TCX – файл базы данных Garmin Training Center (XML документ) для хранения GPS и фитнес-данных, их обмена между устройствами. В отличие от TCX формата, FIT формат является бинарным. Назначение и содержание файла аналогично TCX.

Для обработки данных FIT формата был использован набор средств для разработки FIT SDK. Для удобства дальнейшей разработки был создан класс Gate. Он позволяет обрабатывать в бинарном файле необходимые FIT сообщения (наборы данных определенного типа), проверять

целостность файла и корректность хранимых данных. После обработки всего бинарного файла, полученные данные сортируются по дате и объединяются по виду содержимого. Создание и проектирование классов выполнялось при помощи подробной спецификации FIT формата. Для правильной обработки TCX файла была использована XSD схема, скачанная с официального сайта Garmin. Исходными данными для обработчика FIT формата должен являться массив байтов, для TCX – строка в кодировке UTF-8.

Определение подключенных устройств и обмен данными с устройствами в разработанной подсистеме осуществляется при помощи Garmin Communicator Plugin API. API представляет собой Javascript-библиотеку, которая может обмениваться информацией между ПК и устройством. API состоит из трех уровней: DeviceDisplay, DeviceControl, DevicePlugin. В приложении был сконфигурирован в соответствии с требованиями модуль DeviceDisplay, были созданы вспомогательные функции для корректной работы с устройствами и удобного представления данных.

**А.О. Пименов (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Д. Левчук**, к.т.н., доцент

## **ОБРАБОТКА ДАННЫХ И ВЗАИМОДЕЙСТВИЕ С GPS-УСТРОЙСТВАМИ**

Существует много устройств, которые используют спутниковую систему навигации и позволяют измерять пульс человека (heart rate monitor – сокращенно HRM, монитор сердечного ритма), давление, количество израсходованных калорий и т.п. Индивидуальные портативные навигаторы предназначены для спортсменов-профессионалов и сторонников здорового образа жизни, которые хотят получать больше информации о том, как проходят их тренировки.

Разработанная подсистема для анализа GPS и HRM данных поможет подвести итоги тренировочной деятельности, используя анализ тренировочных зон ЧСС, времени, набора высоты, скорости прохождения, изменения сердечного ритма. Интерактивная карта и различные графики помогут наглядно увидеть результаты тренировок спортсмена. Данная подсистема была разработана для сайта [www.olt-dagbok.net](http://www.olt-dagbok.net). Веб-приложение обеспечивает чтение данных с устройств Garmin и обработку данных: построение на интерактивной карте пройденного пути, построение графиков по различным параметрам (скорость, пульс, высота над уровнем моря и т. д.).

Пользователь может загружать данные тренировок с устройств компании Garmin, как с помощью USB-интерфейса, так и при помощи технологии ANT+. Приложение использует протоколы ANT+ и FIT.

ANT является запатентованной беспроводной технологией сенсорных сетей, принцип действия основан на использовании радиоволн. Радиосвязь ANT осуществляется в ISM-диапазоне. Протокол ANT+ разработан на базе протокола ANT. Приложения, разработанные с использованием ANT, характеризуются периодическими передачами небольшого количества информации между несколькими взаимосвязанными устройствами (до нескольких десятков).

The Flexible and Interoperable Data Transfer (FIT) Protocol – протокол, разработанный специально для хранения данных и обмена данными между HRM-устройствами (или аналогичными). Протокол является компактным и расширяемым.

Созданное приложение поддерживает TCX и FIT форматы хранения данных. TCX – файл базы данных Garmin Training Center (XML документ), разработанный компанией Garmin для хранения GPS и фитнес-данных, их обмена между устройствами. Данный формат файла поддерживается различными устройствами компании Garmin. Файл может включать в себя данные о различных характеристиках в каждой точке (температура воздуха, пройденный путь, скорость и т. д.), суммарную информацию о тренировке, информацию об устройстве (наименование, серийный номер).

В отличие от TCX формата, FIT формат является бинарным. Назначение и содержание файла аналогично TCX. FIT формат используется в программном обеспечении и устройствах Garmin.

Для обработки данных FIT формата был использован набор средств для разработки FIT SDK. Он позволяет генерировать библиотеки, отвечающие конкретным требованиям разработчика; эффективно использовать формат; значительно сократить усилия по разработке и развитию продукта, использующего FIT формат.

Были сгенерированы необходимые классы на языке C++, т.к. FIT SDK не поддерживает генерирование C# классов (только на C++, C, Java). После этого все классы были переписаны на язык C#. Основная часть работы заключалась в переводе небезопасного кода (прямая работа с памятью и т. п.) в безопасный (выполняемый средой CLR). Для удобства дальнейшей разработки был создан класс Gate. Он позволяет обрабатывать в бинарном файле необходимые FIT сообщения (наборы данных определенного типа), проверять целостность файла и корректность хранимых данных.

В данном случае при нахождении корректных данных генерируется событие, аргументами которого являются различные наборы данных. Эти события обрабатываются на другом уровне приложения. После обработки всего бинарного файла, полученные данные сортируются по дате (т. к. возможна непоследовательная запись данных с устройства) и объединяются по виду содержимого (информация об устройстве, точках пути, суммарная информация). Это осуществляется классами, реализующими соответствующие интерфейсы. Создание и проектирование классов выполнялось при помощи подробной спецификации FIT формата, в которой также содержались рекомендации к разработке, используемые единицы измерения и краткие описания полей. Была создана библиотека `Fit.Kit.dll`, которая была включена в проект для дальнейшего использования.

TCX представляет собой XML-документ. Для правильной обработки была использована XSD схема, скачанная с официального сайта Garmin. XSD (XML Schema) – язык описания структуры XML-документа. Спецификация XML Schema является рекомендацией W3C. Таким образом был создан обработчик TCX формата. Исходными данными для обработчика FIT формата должен являться массив байтов, для TCX – строка в кодировке UTF-8.

Определение подключенных устройств и обмен данными с устройствами в разработанной подсистеме осуществляется при помощи интерфейса прикладного программирования `Garmin Communicator Plugin API`. `Garmin Communicator Plugin` – бесплатный плагин для браузера, который обеспечивает доступ веб-приложения к GPS-устройствам Garmin. Данный плагин позволит быстро и без проблем пересылать данные между устройством и компьютером. Программный интерфейс плагина (API) позволяет разработчикам с лёгкостью интегрировать в приложение поддержку GPS-устройств. API представляет собой Javascript-библиотеку, которая может обмениваться информацией между ПК и устройством.

API состоит из трех уровней. Первый уровень (`DeviceDisplay`) самый простой в использовании, он может быть добавлен на веб-сайт и сконфигурирован с помощью нескольких строк кода. Следующий уровень – `DeviceControl`, он предназначен для программистов, которым необходимо больше контроля над моделью событий. Наконец, `DevicePlugin` – является низкоуровневой оболочкой плагина для разработчиков, которым нужен полный доступ. Все что нужно для интеграции в приложение большинства функций API – это исходные JavaScript файлы.

В приложении был сконфигурирован в соответствии с требованиями модуль `DeviceDisplay`, были созданы вспомогательные функции для

корректной работы с устройствами и удобного представления данных.

Запись и чтение осуществляется асинхронно, например, данные о подключенных устройствах можно получить после обработки события `onFinishFindDevices`.

```
onFinishFindDevices: function (json) {
    GarminControl.Callbacks.searchFinish();
    var devices = json.controller.getDevices();
    if (json.controller.numDevices != 0) {
        GarminControl.Callbacks.getDevices(json.controller.getDevices());
    } else {
        GarminControl.Callbacks.getDevices(null);
    }
}
```

Для корректной работы с приложением необходимо установить 2.9.3.0 или более позднюю версию плагина. Если плагин не установлен, не поддерживается браузером, или версия ниже 2.9.3.0 – приложение выводит сообщение об ошибке.

Таким образом, разработанная подсистема может обрабатывать данные от подключенных к ПК HRM-устройств.

**Барскар Пиюш (УО «ГГТУ им. П.О. Сухого», Гомель)**

Науч. рук. **И.А. Мурашко**, д.т.н., доцент

## **МИНИМИЗАЦИЯ ЭНЕРГОПОТРЕБЛЕНИЯ МНОГОВХОДОВЫХ СУММАТОРОВ ПО МОДУЛЮ ДВА**

Многовходовые сумматоры по модулю два нашли широкое применение в различных цифровых устройствах: генераторах псевдослучайных тестовых последовательностей, сигнатурных анализаторах, коммуникационных устройствах и т. п. Реализация произвольной логической функции на элементах «Исключающее ИЛИ» (*XOR*) часто является более эффективной с точки зрения аппаратных затрат (площади кристалла СБИС) и/или потребления энергии [1]. Особенно актуальна в настоящее время задача синтеза многовходовых логических элементов с минимальным энергопотреблением [2]. В [1] представлена методика оценки средней переключательной активности схем на основе элементов *XOR*. Это позволяет получить среднюю оценку переключательной активности. В настоящей работе рассмотрен случай, когда смена логических состояний на входах сумматора может происходить только в различные моменты времени, что позволяет получить максимально возможную оценку переключательной активности

и, соответственно, максимальную потребляемую мощность, что гарантирует надежную работу схемы (реальная мощность никогда не превысит расчетную). Для этого случая в работе предложена методика декомпозиции многовходовых сумматоров по модулю два с минимальным энергопотреблением. Причем, в отличие от работы [3], декомпозиция может проводиться на двухвходовые, трехвходовые и т. п. элементы. Получены оценки минимальной и максимальной переключательной активности. Приведены примеры минимальной реализации многовходового сумматора.

Энергопотребление цифровых КМОП схем вызвано следующими четырьмя основными источниками: токами утечки, обратными токами *pn*-переходов, сквозными токами при переключении элемента и токами заряда-разряда паразитной емкости. Причем основной вклад в энергопотребление (90–99 %) вносят два последних фактора (динамическая составляющая). Как показано в [3], потребляемая схемой энергия может быть найдена следующим образом:

$$E = WSA \cdot E_0, \quad (1)$$

где  $E_0$  – энергия одного переключения (определяется переключаемым напряжением и нормализованной емкостной нагрузкой одного входа),  $WSA$  – переключательная активность схемы (среднее число переключений в течение такта работы).

Для анализа переключательной активности будем использовать модель, предложенную в [2], которая позволяет получить максимально возможную оценку переключательной активности (и, соответственно, максимальную оценку энергопотребления). Будем считать, что смена состояний в узлах схемы происходит в непересекающиеся моменты времени, а на входы поданы сигналы с максимальной переключательной активностью  $WSA_i=0,5$ . В этом случае переключательная активность  $d$ -входового сумматора по модулю два, реализованного на  $b$ -входовых элементах *XOR*, будет:

$$WSA = \sum_{i=1}^n WSA_i, \quad (2)$$

где  $WSA_i$  – переключательная активность  $i$ -го узла,  
 $n$  – количество внутренних узлов.

Рассмотрим вывод выражений для оценки максимального ( $WSA_{max}$ ) и минимального ( $WSA_{min}$ ) значений переключательной активности  $d$ -входового сумматора по модулю два, реализованного на  $b$ -входовых элементах *XOR*. Минимальная реализация  $d$ -входового сумматора может быть выполнена на  $k$  элементах, где



$$k = \left\lceil \frac{d-1}{b-1} \right\rceil. \quad (3)$$

Исследования показали, что максимальную переключательную активность имеет схема с последовательным расположением элементов (рис. 1), которую можно определить как  $((x_1 \oplus x_2 \oplus x_3) \oplus x_4 \oplus x_5) \oplus x_6 \oplus x_7$ .

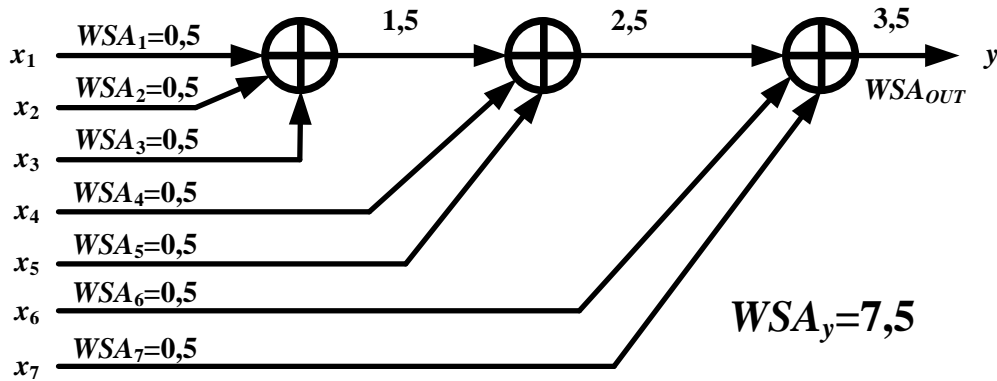


Рисунок 1 – Пример реализации семивходового сумматора по модулю два на трехвходовых элементах

В общем случае, максимальную переключательную активность можно найти из следующего выражения:

$$WSA_{\max} = x \left( \sum_{i=1}^k (b-1)i + 1 \right) - N, \quad (4)$$

где

$$N = (b-1)k + 1 - d. \quad (5)$$

Для минимальной переключательной активности выражения получаются не столь регулярными и имеют вид:

$$WSA_{\min} = \begin{cases} x(d + 3(k-1) - N) & \text{для } 0 < k \leq 4, \\ x(d + 3(k-1) + 2(k-4) - N) & \text{для } 4 < k \leq 13, \\ \dots & \end{cases} \quad (6)$$

При  $d=7$  ( $k=3$ ) получим  $WSA_{\min}=6,5$ . В компактной форме схема декомпозиции запишется как  $((x_1 \oplus x_2 \oplus x_3) \oplus (x_4 \oplus x_5 \oplus x_6) \oplus x_7)$ .

На рисунке 2 представлены графики зависимости минимальной и максимальной переключательной активности от числа используемых вентилей.

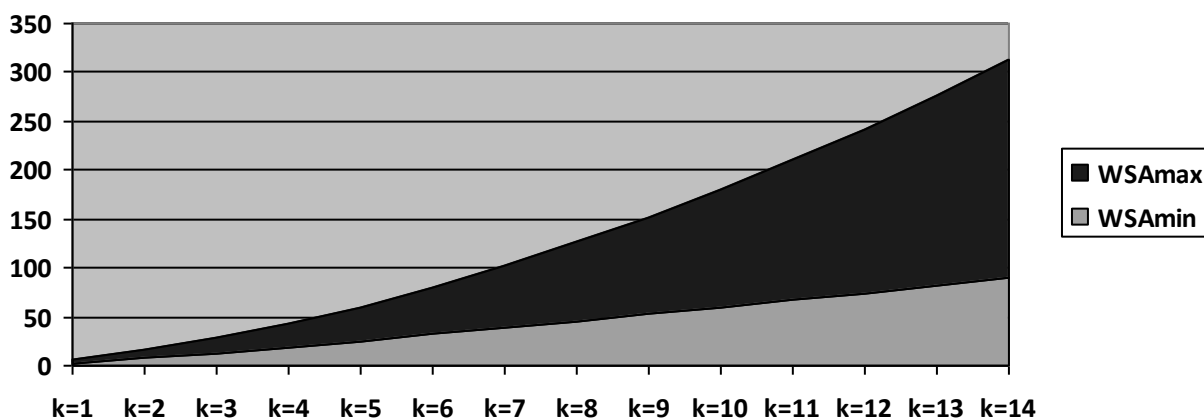


Рисунок 2 – Сравнение между максимальной и минимальной переключательной активности

Анализ графика показывает, что уже при  $k > 4$  минимальная переключательная активность практически в два раза меньше, чем максимальная.

### Литература

1. Roy, K. Low Power CMOS VLSI Circuit Design / K. Roy, S.C. Prasad. – New York: John Wiley and Sons, Inc., 2000. – 376 p.
2. Мурашко, И.А. Встроенное самотестирование. Методы минимизации энергопотребления (монография) / И.А. Мурашко, В.Н. Ярмолик. – Saarbrücken: LAP LAMBERT Academic Publishing, 2012. – 339 с.
3. Мурашко, И.А. Анализ энергопотребления многовходового сумматора по модулю два / И.А. Мурашко // Информатика. – 2006. – № 1 (9). – С. 97–103.

**А.В. Потапенко, О.А. Кузьменко (УО «ГГУ им. Ф. Скорины», Гомель)**  
 Науч. рук. **С.В. Карпенко**, к.э.н., доцент

## ВЛИЯНИЕ СОВРЕМЕННОГО ЭКОНОМИЧЕСКОГО КРИЗИСА НА ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Информационные технологии (ИТ) и экономика тесно связаны и оказывают мощное взаимное влияние друг на друга.

ИТ в том виде, в котором они существуют сейчас, сформировались благодаря тому, что представители сфер бизнеса и управления поняли: информация и информационные технологии являются мощным ресурсом для развития. Благодаря такой востребованности ИТ в последнее десятилетие стали одной из самых быстроразвивающихся отраслей,

обогнавшей по темпам развития даже космические технологии. С периодичностью в несколько месяцев происходит значительный шаг вперед и связанная с этим полная смена стандартов.

Развитие ИТ, в свою очередь, оказывает влияние на современную экономику. Информация вышла на лидирующие позиции в списке жизненно необходимых и важных составляющих экономики. Наиболее развитыми странами считаются страны, имеющие высокую степень информационного развития [1].

Возможности, предоставляемые современными ИТ, способствуют повышению динамичности экономики, повышению скорости оборота капитала на рынке. Повышение скорости и виртуализация выполнения операций (заключения сделок, исполнения заказов, перевода денег и т.д.) способствуют интенсификации и развитию экономики.

Повышение темпов развития экономики, усиление конкурентной борьбы, связанное с развитием ИТ, заставило искать новые структуры управления. Эти теории получили название TQM (Total Quality Management) и BPR (Business Process Reengineering). Под влиянием этих теорий произошли децентрализация управления и разукрупнение организаций. В результате смещены права управления и ответственности за процесс на нижние уровни. Такой способ построения организации обеспечивает гибкость и необходимую динамичность управления вне зависимости от размера организации.

Важным аспектом влияния ИТ на экономику явилось широкое распространение понятия «интеллектуальная собственность» и возрастание её роли в экономических процессах. Необходимо отметить, что значительную долю оборота средств в области ИТ занимает именно продажа интеллектуальной собственности (лицензии и т. д.) [2].

Настоящим «зеркалом кризиса» в ИТ-индустрии являются большие и всем известные корпорации. Например: темпы роста рынка таких компаний, как Intel, Applied Materials и National (производителей полупроводников), Google, Nokia заметно снизились, в результате чего эти компании понесли большие потери, измеряемые в сотнях миллионов долларов в период с 2006 по 2009 год.

Компания Microsoft также понесла убытки из-за сокращения продаж серверов и персональных компьютеров [3].

В последнем отчетном квартале доложила о худших в своей истории финансовых показателях сразу вся топовая обойма компьютерной индустрии – Microsoft, Google, IBM, HP.

Причина – очередная волна острой фазы кризиса (идеальный шторм, как метафора новой Великой Депрессии, с легкой руки профессора

Нью-Йоркского университета Нуриэля Рубина ставший обозначением того, что ожидается в предстоящие 2013–2014 годах). Немаловажными считаются события, происходящие на стоп-торгах у таких крупных фирм как Google, Apple, когда общая капитализация компании Google рухнула сразу на 12 %, после чего торговля её акциями была временно заблокирована. После успокоения рынка торговля возобновилась с позиции – 9 %. Столь же драматичные стоп-торги не так давно случились и у второго вышеуказанного лидера мира ИТ.

В 2013 году мировые расходы ИТ-индустрии составят 3,74 трлн долларов. Основное влияние на бизнес компаний окажут корпоративное программное обеспечение, социальные системы и мобильные коммуникации, однако, к 2014 году фокус на рынке сместится в пользу больших данных и технологий информационного менеджмента, таких как управление корпоративным контентом, средства интеграции данных и подобных.

По прогнозу рост рынка корпоративного программного обеспечения составит 6,4 %, объем достигнет 296 млрд долларов. Этот сегмент будет зависеть от следующих ключевых рынков: безопасность, управление данными, управление связями с клиентами.

Наименьший рост – 2,4 %, покажет сегмент телекоммуникационных сервисов.

Крупнейшим по объему останется рынок ИТ-сервисов: в 2013 году – 927 млрд долларов, а в 2014 году может достичь 1 трлн долларов.

Международная ИТ-ассоциация CompTIA проанализировала экономическое состояние ИТ-индустрии и сделала несколько прогнозов на ближайшие годы. Согласно оценкам CompTIA, ИТ в будущем ожидает сдержанный, но стабильный рост. Так, в 2013 г. по всему миру рост ИТ-индустрии составит 3 %. Локомотивом развития станет США: ИТ-отрасль в Америке в 2013 г. прибавит почти столько же процентов, сколько и в мире – 2,9 % . Если отчетливо проявятся кризисные явления, то можно предположить падение рынка, аналогичное тому, которое наблюдалось в 2008–2009 гг. Рынок геоинформационных технологий в 2013 г. снизит темпы, несмотря на привычный рост в 20–25 %. Прежде всего, это опасения связано с тем, что сейчас происходит сокращение бюджетов на ИТ-проекты в ряде государственных ведомств.

В 2013 году рынок персональных компьютеров так же сможет вернуться к росту. Благодаря выходу новых моделей под управлением ОС Windows 8 и снижению цен на ультрабуки, мировые поставки ПК достигнут 332 млн устройств (127 млн десктопов и 205 млн) и по сравнению с нынешним годом увеличатся на 3,2 %. Рост рынка обеспечат

ноутбуки, отгрузки которых увеличатся на 5,9 %. В сегменте настольных ПК падение показателей хоть и не прекратится, но замедлится (-0,9 %). [4].

ИТ играют важную стратегическую роль, которая быстро возрастает, т. к. ИТ активизируют и повышают эффективность использования информационных ресурсов, обеспечивают экономию сырья, энергии, полезных ископаемых, материалов и оборудования, людских ресурсов, социального времени, реализуют наиболее важные и интеллектуальные функции социальных процессов, обеспечивают информационное взаимодействие людей, способствуют распространению массовой информации и т. д. [5].

### Литература

1. Роль информационных технологий в развитии экономики [Электронный ресурс]. – Режим доступа: <http://uztelecom.uz/ru/press/news/2012/859/>. Дата доступа: 13.03.2013 г.

2. Роль ИТ в развитии экономики и общества [Электронный ресурс]. – Режим доступа: <http://admp.ru/?p=7>. Дата доступа: 13.03.2013 г.

3. ИТ-гиганты на фоне кризиса [Электронный ресурс]. – Режим доступа: <http://www.nestor.minsk.by/sr/2008/11/sr81107.html>. Дата доступа: 15.03.2013 г.

4. ИТ-экономика: в ожидании идеального шторма [Электронный ресурс]. – Режим доступа: <http://dev.by/blogs/tags/кризис>. Дата доступа: 19.03.2013 г.

5. Мировой кризис тормозит развитие ИТ [Электронный ресурс]. – Режим доступа: [http://www.sostav.ru/news/2013/01/09/ktizis\\_it/](http://www.sostav.ru/news/2013/01/09/ktizis_it/). Дата доступа: 20.03.2013 г.

**В.И. Рагин (УО «ГГУ им. Ф. Скорины»)**

Науч. рук. **В.Н. Кулинченко**, старший преподаватель

### **ЗАМЕНА ДВОИЧНОЙ ЛОГИКИ, УВЕЛИЧЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ**

Двоичная система не является единственной используемой в цифровой технике, на заре компьютерной эры существовали машины, которые использовали десятичную и троичную систему счисления (СС).

Десятичная система удобна человеку, но ее достаточно сложно реализовать на существующей элементной базе. Кроме того, десятичная система подвержена ошибкам в результате искажения сигнала при передаче. Троичную систему реализовать не на много сложнее двоичной, но она способна дать как минимум три преимущества:

– представление отрицательных чисел становится естественным. Так как минимальная единица информации в троичной системе (будем называть ее трит) способна принимать три возможных значения, появляется возможность по-разному представлять ноль и положительное число;

– трайт (совокупность из восьми трит) способен принимать значения в диапазоне  $\pm 3 \cdot 280$ , включая ноль, что приблизительно в тридцать раз больше числа возможных значений байта в диапазоне только положительных чисел (байт принципиально не может принимать отрицательные значения без введения специальных условностей);

– трит способен принимать значение «не определено», что ближе к модели человеческого мышления, чем совокупность двоичных значений «истина» и «ложь»;

На сегодняшний день существуют лишь экспериментальные проекты вычислительной техники построенной на троичной логике, раньше, массовому запуску в производство мешало недостаточная развитость электронной промышленности и сопутствующие сложности при проектировании систем троичного исчисления. Но сегодня эти проблемы не столь существенны и переход на троичную систему позволит сделать новый шаг в развитии компьютерной техники.

Среди плюсов троичной СС над двоичной можно выделить:

– меньше разрядов. Возьмем число 10 в десятичной СС и переведем его в двоичную СС, получим 1010, переведем в троичную симметричную СС, получим +0+, ну а если в троичную несимметричную СС, то получим 101.

– емкость. Троичная СС вмещает больший диапазон чисел, т. к.  $3^n > 2^n$  (где n-натуральное число). Например, если  $n=9$ , то  $3^9=19683 > 2^9=512$ .

– экономичность системы счисления, т.е. запас чисел, который можно записать в данной системе с помощью определенного количества знаков. Чем больше запас, тем экономичнее система.

**А.Ю. Речицкий (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Е.А. Левчук**, к.т.н., доцент

## **АВТОМАТИЗАЦИЯ ДОКУМЕНТООБОРОТА РЕГИСТРАТУРЫ В УЧРЕЖДЕНИИ «МОЗЫРСКИЙ ГОРОДСКОЙ РОДИЛЬНЫЙ ДОМ»**

Внедрение информационных технологий в различные сферы жизни дает огромные преимущества, оптимизируя нашу деятельность и делая

ее более эффективной. Не является исключением и медицина. Более того, именно в данной области в последние годы много усилий и денежных средств вкладывается в автоматизацию работы медицинских учреждений.

По принципу работы все системы делятся на две категории – это офлайн-системы и онлайн-системы. При использовании офлайн-системы программа устанавливается на компьютер пользователя. Соответственно, формирование отчетности происходит на жестком диске компьютера.

Автоматизация регистратуры поликлиники предоставляет следующие возможности:

- возможность автоматизировать рутинные операции, осуществляемые сотрудниками регистратуры поликлиники, позволяя им выполнять свою работу более быстро и эффективно;
- возможность более эффективно и оперативно организовать процедуру приема пациентов и запись их к врачам-специалистам;
- возможность более оптимально составлять расписание приема врачей, исключая возможность совершения ошибок.

База данных предназначена для работников регистратуры поликлиники и позволяет им повысить эффективность своей работы за счет систематизации и быстрого поиска нужной им информации. Это гораздо упрощает работу, т. к. отпадает необходимость просматривать горы бумаг в поисках нужной информации, писать справки каждому пациенту и отчеты о работе.

В БД должны храниться сведения о больных: ФИО, адрес, диагноз, дата заболевания; сведения о врачах: ФИО, № кабинета, № участка, дни и часы приема; описание болезней: название (диагноз), симптомы, лекарства.

Работнику регистратуры поликлиники может понадобиться информация о больных, врачах, известных заболеваниях, а также справка о заболеваниях пациента и отчет о работе поликлиники.

Необходимо, чтобы в БД хранились сведения о больных, врачах, диагнозах каждого больного, описания всех известных диагнозов (симптомы и рекомендуемое лекарство). Также должны храниться сведения об участках (за какими адресами закреплены и какими врачами обслуживаются).

Такое представление повышает удобство использования базы данных, в данном случае ввод информации сведется к выбору необходимых сведений из списка, где это возможно, что, безусловно, повысит скорость ввода информации и поможет избежать неверного ввода параметров.

Перед созданием базы данных необходимо располагать описанием выбранной предметной области, которое должно охватывать реальные объекты и процессы, определить все необходимые источники информации для удовлетворения предполагаемых запросов пользователя и определить потребности в обработке данных.

На основе такого описания на этапе проектирования базы данных определяется состав и структура данных предметной области, которые должны находиться в БД и обеспечивать выполнение необходимых запросов и задач пользователя. Структура данных предметной области может отображаться информационно-логической моделью. На основе этой модели легко создается реляционная база данных.

В базе данных сведения из каждого источника сохраняются в отдельной таблице. При работе с данными из нескольких таблиц устанавливаются связи между таблицами. Для поиска и отбора данных, удовлетворяющих определённым условиям, создаётся запрос. Запросы также позволяют обновить или удалить одновременно несколько записей, выполнить встроенные и специальные сообщения.

Для просмотра, ввода или изменения данных прямо в таблице применяются формы. Форма позволяет отобразить данные из одной или нескольких таблиц и вывести их на экран, используя стандартный или созданный пользователем макет.

В окне базы данных можно работать со всеми её объектами. Для просмотра объектов определённого типа следует выбрать соответствующую вкладку. С помощью кнопок можно открывать и изменять существующие объекты и создавать новые.

С распространением безбумажных технологий представления отчетности у сотрудников появится больше времени на приложение усилий направленных на улучшение эффективности работы и развитие организации.

**А.М. Ролич (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Н. Леванцов**, старший преподаватель

## **ОРГАНИЗАЦИЯ СИСТЕМЫ ДЕЛОПРОИЗВОДСТВА**

Рыночные условия предъявляют жесткие требования к процессу принятия управленческих решений. Эффективность управления обеспечивается оперативностью, достоверностью и достаточностью информации, на основе которой принимается конкретное решение. Одним из основных способов представления информации является



документ. Таким образом, документ можно рассматривать как средство принятия управленческих решений, поэтому правильно организованная система делопроизводства является значимой частью системы управления и залогом успеха фирмы.

Одной из проблем традиционной системы делопроизводства, основным объектом которой является бумажный документ, является экспоненциальный рост количества документов. Кроме того, возрастает доля электронных документов.

Средством повышения эффективности делопроизводства является использование компьютерных технологий, в частности, внедрение систем автоматизации делопроизводства. Именно компьютерные системы способны управлять растущим объемом информации, они создают принципиально новую технологическую среду для эффективного управления документооборотом.

Компьютеры, являются основными инструментами работы с документами. Распространение и совершенствование систем разработки электронных документов и документов на традиционных носителях привело к росту объема электронных документов и способствовало появлению класса программных продуктов, называемых системами автоматизации делопроизводства.

В последние годы наряду с термином «делопроизводство» употребляется и синонимичный ему термин «документационное обеспечение деятельности организации». Возникновение его связано с изменением организационно-технической основы делопроизводства и подходов к его совершенствованию, ставшее возможным благодаря активному внедрению в сферу работы с документами средств вычислительной техники и современных информационных технологий создания, сбора, обработки, накопления, хранения, поиска и использования информации.

Одна из важнейших функций систем документационного обеспечения – функция учета документов. Учет документов обеспечивается их регистрацией-записью учетных данных о документе по установленной форме, фиксирующей факт создания документа, его отправки или получения. Наряду с функцией учета документов регистрация позволяет осуществлять контроль исполнения документов, вести поиск документов по запросам подразделений и работников учреждения.

Таким образом, правильно организованная система делопроизводства ускорит и облегчит работу секретаря в приемной по учету деловой документации.

**О.Ю. Романова, И.С. Нестерович (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **С.В. Карпенко**, доцент, к.э.н.

## **СОСТОЯНИЕ И ПЕРСПЕКТИВЫ ИТ-РЫНКА**

Экономическая неопределенность продолжает оказывать негативное влияние на рынок информационных технологий (ИТ), в том числе на рынок ИТ-сервисов как в США, так и Европе. В 2012 году мировой рынок ИТ-услуг показал худшие результаты с 2002 года [1]. Показатели 4-го квартала 2012 года относительно сопоставимы с итогами 4-го квартала 2011 года, но результаты всех 12 месяцев неутешительны: по объему сделок и совокупной стоимости контрактов (total contract value, TCV) рынком были достигнуты худшие параметры за десятилетие. TCV в четвертом квартале 2012 года [1] составил \$20,8 млрд, что на 34 % ниже, чем годом ранее. Количество заключенных сделок за сопоставимый период снизилось на 17 %, существенно уменьшилось число мега-сделок (со стоимостью контрактов от \$1 млрд и выше). Показатель TCV по 2012 году в целом показал снижение за год как в публичном, так и частном секторе. При этом в частном секторе показатель TCV оказался худшим с 1998 года.

Самая важная тенденция — это рост затрат компаний на ИТ. По данным Gartner, 2002 год был годом самых низких ИТ-бюджетов. В 2003 году зафиксирован небольшой их прирост — 5 %. И хотя увеличение затрат компаний на ИТ невелико, это уже достаточно важная тенденция — во-первых, положительная динамика, а во-вторых, настроение. Этот прирост вызвал пробуждение прежде всего рынка ИТ-услуг. При падении ИТ-бюджетов компании в основном экономят на таких услугах, как разработка ИТ-стратегии и обучение. При росе ИТ-бюджетов компании снова стали интересоваться этими услугами [2].

Мировые расходы на ИТ достигли в 2012 году \$3,6 трлн, что на 3 % выше 2011 года, когда этот показатель составил \$3,5 трлн. [3]. Они восстанавливаются, но не во всех секторах и странах.

Сегменты ПО и ИТ-сервисов в 2011 году восстановились практически на всех рынках. Существует большая разница в развитии конкретных стран [3]. В целом 2012 год был трудным для отрасли ИТ.

Климат ИТ-рынка зависит от экономической ситуации в регионе. В странах, испытывающих экономические трудности (Греция, Италия, Португалия) организации краткосрочно сокращают ИТ-расходы. В Австралии, США и странах БРИК они более прибыльны и эффективны,

Финансовая отрасль, особенно инвестиционный банкинг, и агентства публичного сектора демонстрируют дефицит средств. Ситуация

в производстве остается сопоставимо позитивной. В одной отрасли на ИТ-расходы влияет экономическая ситуация в стране.

Число поставщиков ИТ-продуктов продолжает сокращаться. Сохраняется высокий спрос на аутсорсинговые услуги. На первом месте ИТ-проекты, на 100 % необходимые, сокращающие издержки или обеспечивающие быстрый ROI.

Мировые расходы на ИТ, по прогнозу Gartner от января 2013 года, в 2013 году достигнут \$3,7 трлн, что на 4,2 % больше показателя 2012 года. Выраженные в постоянной валюте (доллар) расходы на ИТ в 2013 году увеличатся на 3,9 %. В сегменте ПК, планшеты, мобильные телефоны и принтеры мировые расходы на устройства, по прогнозам, достигнут \$666 млрд в 2013 году, что на 6,3 % больше 2012 года. Долгосрочный прогноз для рынка устройств понижен по состоянию на январь 2013 года: с 2012 по 2016 годы он составит порядка 4,5 % ежегодно в долларовом выражении (ранее давался показатель в 6,4 %) и порядка 5,1 % ежегодно в постоянной валюте (ранее – 7,4 %).

Таблица 1 – Прогноз мировых ИТ-расходов 2012–2014, \$ млрд

	Расходы 2012	Рост 2012	Расходы 2013	Рост 2013	Расходы 2014	Рост 2014
Устройства	627	2,9 %	666	6,3 %	694	4,2 %
Дата-центры	141	2,3 %	147	4,5 %	154	4,2 %
Корпоративное ПО	278	3,3 %	296	6,4 %	316	6,8 %
ИТ-сервисы	881	1,8 %	927	5,2 %	974	5,1 %
Телеком-сервисы	1661	-0,1 %	1701	2,4 %	1742	2,4 %
Всего	3588	1,2 %	3737	4,2 %	3881	3,8 %

Источник; Gather, январь 2013

Корпоративные расходы на ПО вырастут в 2013 году по отношению к 2012 году на 6,4 % до \$296 млрд. Ключевыми драйверами станут такие сегменты как безопасность, ПО управления хранилищами и CRM. В 2014 г. акцент сместится в пользу баз данных и технологий информационного менеджмента, таких как управление документооборотом [2]. Для достижения стратегических целей инвестиции в ИТ должны сопровождаться модернизацией системы управления. Необходимы изменения в политиках, процессах, оргструктуре, корпоративной культуре.

Мировые ИТ-расходы предприятий в 2013 году достигнут \$2.679 трлн. Это на 2,5 % больше прогнозов 2012 г. [3]. Максимальный рост расходов до 2016 года продемонстрируют финансовые, коммуникационные компании, сферы медиа и сервисов, производственные предприятия.

Год оценивают оптимистично, т.к. США и Европе удалось избежать глобальных экономических потрясений. Лидеры вертикальных рынков по объему ИТ-расходов [3]: сектор производства и природных ресурсов. Основные расходы: автоматизация и оптимизация процесса продаж, облачные, социальные, мобильные технологии и big data.

Банки и сектор безопасности: в 2013 г.: затраты на ИТ составят \$460 млрд. Это на 3,5 % больше, чем в 2012 г. (\$445 млрд). Инвестиции в ИТ в 3 раза больше в % от собственной выручки, чем в других отраслях. Области инвестиций: автоматизация займов, платежей и управление рисками [2].

Центральная и Восточная Европа: из \$158 млрд ИТ-расходов в 2012 году, 48,2 % приходится на корпоративный сектор. Доля России – 45 %, 2-е место у Польши – 11,8 %, Чехия – 7,7 %, Венгрия – 3,7 %. Большинство отраслей требуют масштабного развития в 2013–2015 годы. Объем иностранных инвестиций в белорусскую ИТ-отрасль за 2012 год вырос в 1,7 раза.

10 наивысших бизнес-приоритетов: 1 – Анализ и бизнес-анализ; 2 – Мобильные технологии; 3 – Облачные вычисления (SaaS, IaaS, PaaS); 4 – Технологии совместной работы (документооборот); 5 – Виртуализация; 6 – Модернизация; 7 – ИТ-управление; 8 – CRM; 9 – Приложения ERP; 10 – Безопасность.

10 главных трендов мирового ИТ рынка: 1) 2016 г. вовлечение руководства в 80 % новых проектов в области ИТ; 2) 2015 г. – 90 % ИТ-инвестиций будут оцениваться на соответствие стратегическим целям компаний; 3) взаимное влияние должностей «ИТ-директор» и «финансовый директор»; 4) в 2016 г. облачные решения возглавят стратегии ИТ; 5) BI и Big Data превзойдут по актуальности все другие ИТ-проекты; 6) 2018 г. 50 % топ-менеджеров будут воспринимать ИТ-директоров как бизнес-инноваторов [4].

### Литература

1. Шматалюк, О. Три важнейшие тенденции мирового ИТ-рынка [Электронный ресурс]. – <http://www.iemag.ru/analytics/detail.php> – Дата доступа: 20.03.2013.

2. ИТ (мировой рынок) // TADVISER Государство. Бизнес. ИТ – 2013. – 6 февр. – URL: <http://www.tadviser.ru/index.php>. – Дата доступа: 18.03.2013.

3. Gartner Says Worldwide Enterprise IT Spending is Forecast to Grow 2.5 Percent in 2013 [Electronic resource] // Gartner / STAMFORD, Conn., November 15, 2012. – Mode of access: <http://www.gartner.com/> – Data of access: 20.03.2013.

4. Тенденции мирового ИТ-рынка – URL: <http://www.tadviser.ru/> – Дата доступа: 18.03.2013.

## **РАЗРАБОТКА ВЕБ-СЕРВИСА «КАЛЕНДАРЬ СОБЫТИЙ»**

Активный отдых является одной из важнейших сфер повседневной жизни человека, который, вместе с образованием, способен очень серьезно влиять на состояние человека в частном и общества в целом. Но, к сожалению, вовремя узнать обо всех предстоящих событиях (кинотеатры, клубы, рестораны, кафе, театры, спортивные мероприятия и многое-многое другое) человек зачастую не может. Ведь оно и понятно – у каждого много дел и забот, и уследить за информационным потоком, сыплющемся на нас из афиш, очень трудно. А если событие не сильно рекламируется – то зачастую о нем узнать о нем можно лишь по счастливой случайности.

Частично данную проблему решают интернет-афиши, но и тут есть существенная проблема: слишком много ненужной информации (излишняя перегруженность пользователя информацией).

Для решения этих проблем было решено разработать веб-сервис, предоставляющий удобный и «легкий» интерфейс для поиска событий из сферы развлечений. С помощью этого сервиса любой желающий сможет найти информацию о предстоящих событиях и соответствующим образом спланировать свое время. Ниже перечислены основные функции разрабатываемого сервиса:

- при входе на сайт, по IP-пользователя определяется его город и его местное время;

- на основании этой информации пользователю показываются ближайшие события на сегодняшний день, разделенные по группам (кино, театр, клуб и др.);

- если пользователя заинтересовало какое-либо событие, он может перейти на страницу этого события и посмотреть более детальную информацию;

- на странице события показывать так же место проведения, для того, чтобы пользователь мог просмотреть события только для данного заведения;

- ввести календарь, для поиска не только на сегодняшний день, но и на другие возможные даты.

Разработка данного сервиса реализуется на языке PHP с использованием фреймворка Yii.

**Ф.С. Рябиков (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

## **МОДИНГ BIOS ВИДЕОКАРТЫ**

BIOS видеокарты – это выделенный BIOS видеокарты компьютера. Поскольку видеокарта обладает собственным процессором, оперативной памятью и системой питания, ей необходим выделенный BIOS для управления.

Так же, как и системный BIOS, BIOS видеокарты предоставляет набор функций и инструкций, необходимых приложению для доступа к видеокарте. Видео-BIOS выступает в роли посредника между приложением (в основном драйвером) и аппаратной частью видеокарты. BIOS графической карты также содержит данные о рабочих частотах и напряжениях графического процессора и видеопамати (и ее таймингов), режимы работы системы охлаждения и прочее. На современных видеоадаптерах BIOS прошит в чип перезаписываемой памяти, поэтому допускается редактирование вышеописанных параметров с помощью специальных утилит и последующая прошивка уже отредактированного BIOS взамен стандартного, что часто применяется среди любителей разгона.

В отличие от некоторых компонентов компьютера, видеокарта должна быть задействована еще во время загрузки компьютера, чтобы пользователь мог видеть на мониторе, что происходит. Для этого видеокарте требуется быть задействованной задолго до загрузки операционной системы; значит, она должна быть активирована BIOS'ом – единственной микропрограммой, представленной на начальном этапе загрузки. После запуска компьютера системный BIOS отдает команду на запуск видео-BIOS'a, который, в свою очередь, инициализирует работу всей видеокарты.

После загрузки операционной системы в роль посредника вступает видеодрайвер, оснащенный большим функционалом и работающий по принципу "приложение-драйвер-VideoBIOS-аппаратная часть" (или напрямую "приложение-драйвер-аппаратная часть").

Условно процесс модинга можно условно разбить на три шага.

Шаг 1: Сохранение текущего BIOS'a

Открываем программу GPU-Z, находим строку "BIOS Version", сразу после номера версии BIOS'a находится кнопка с изображением зеленой стрелочки, указывающей на BIOS, нажимаем на нее и выбираем пункт "Save to file...", в появившемся окне указываем куда должен сохраниться полученный файл. Как итог мы получаем файл вида

bios\_name.bin. Можно сразу заменить его в формат, понятный программе NVFLASH, с помощью программы NiBiTOR, в которой открываем наш файл \*.bin и сохраняем его как файл \*.rom. Этот шаг существует не только для сохранения оригинального BIOS'a видеокарты для возможного последующего восстановления, но он так же просто необходим при работе с x64 битными системами семейства Windows, так как уже упоминалось, что NiBiTOR не считывает информацию о BIOS'e в x64 битных системах, а потому сохранить BIOS можно будет только данным способом.

#### Шаг 2: Редактирование BIOS видеокарты

Запускаем программу NiBiTOR и поступаем следующим образом: либо открываем с помощью нее ранее сохраненный файл с BIOS, либо воспользовавшись средствами программы (в случае если у нас x32 битная система) считываем информацию о BIOS'e и начинаем работу с ним. В первом случае пункты 2 и 3 можно пропустить, во втором же делаем все по порядку:

- 1 Запускаем программу NiBiTor.
- 2 Жмем кнопку Tools , выбираем Read BIOS > Select Device.
- 3 Снова жмём Tools , выбираем Read into NiBiTor.
- 4 Выставляем желаемые параметры
- 5 Сохраняем – жмем кнопку File > Save bios (задаём имя сохраняемому биосу).

#### Шаг 3: Прошивка BIOS'a

После всех приготовлений приступаем непосредственно к прошивке BIOS'a. Прошивать мы будем с помощью программы NVFLASH, которая существует в двух версиях: NVFLASH WIN для работы из под Windows, и NVFLASH DOS версия для работы в среде DOS. Желательно прошиваться именно NVFLASH DOS, так как во время работы Windows возможны проблемы связанные как с работой самой операционной системой, так и со сторонним софтом. Однако мы рассмотрим обе версии.

3.1 Прошивка с помощью NVFLASH WIN представляет собой следующий процесс:

Загруженный файл NVFLASH WIN распаковываем, перемещаем все три файла (nvflash ,nvflash32 ,nvflash64) в корень диска C:, сюда же помещаем файл BIOS'a, который мы будем записывать, далее жмем кнопку Win+R > вводим команду cmd > ОК или Enter > в командной строке набираем команду “nvflash –protectoff” (без кавычек).

Далее должно появиться сообщение о том что: "nvflash не является внутренней или внешней программой исполняемой программой или

пакетным файлом". По этому набираем "cd .." (без кавычек) чтобы выйти в корень C: (возможно процедуру придется повторить), затем из положения C:\ набираем выше описанные команды по порядку

- 1 nvflash --protectoff (отключаем защиту)
- 2 nvflash -b reserved.rom (сохранение оригинального BIOS'a)
- 3 nvflash -4 -5 -6 bios\_name.rom (старт прошивки)

После 3 команды в консоли потребуется подтвердить уверенность в собственных действиях, появится предупреждение о моргании экрана, и затем сообщение о том, что все прошло успешно.

3.2 От предыдущего варианта работа с NVFLASH DOS отличается лишь необходимостью создать загрузочную флешку либо дискету, при этом команды используются одни и те же в обеих версиях программы.

Технически процедура моддинга заключается в модификации или замене обнаруженных файлов, настройке закрытых опций и некоторых других операций. Программа, предлагающая удобный интерфейс для проведения этих операций представлена на рисунке 1.

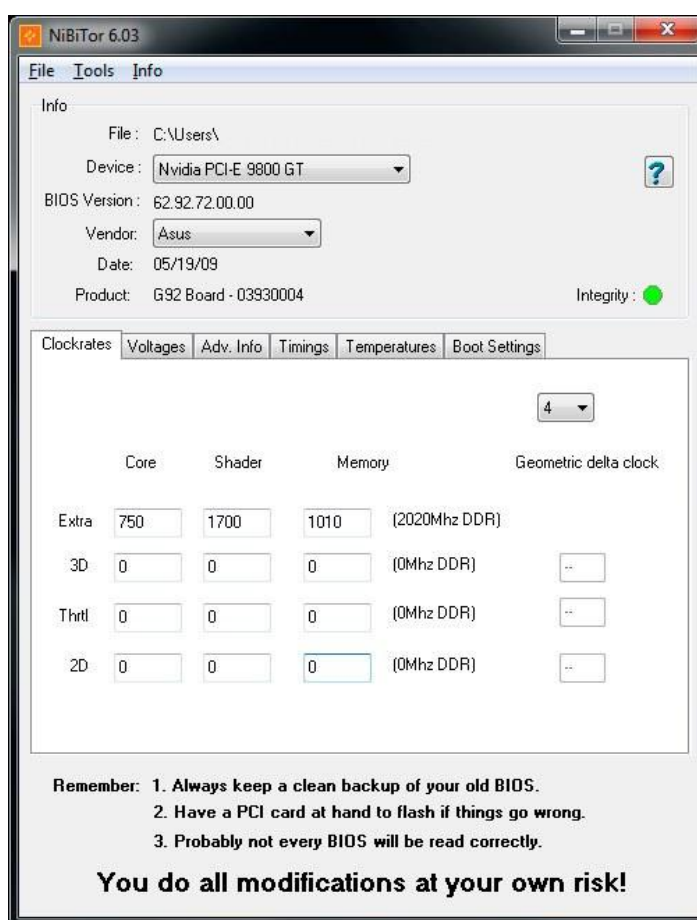


Рисунок 1 – Результат работы утилиты NiBiTOR



## УПРАВЛЕНИЕ СПИСКАМИ ЗАДАЧ НА ПЛАТФОРМЕ iOS

У современного человека всегда есть большое количество планов на будущее, текущих дел на ближайшее время и мелких повседневных задач. И порой все они накапливаются и в них уже становится трудно разобраться. Трудно определить, что наиболее приоритетнее в данный момент. Про часть из них вообще можно забыть. Но среди всей этих задач может быть что-то интересное и важное. Именно эту проблему и решает данное приложение: позволяет структурировать все списки, напоминает о предстоящих задачах, позволяет планировать предстоящие дела. Для мобильной платформы iOS и, впоследствии, для настольной платформы OS X было создано приложение, которое позволяет пользователю создать аккаунт и начать наслаждаться порядком в своих задачах. Приложение дает возможность создавать списки задач, которые в свою очередь содержат задачи. После выполнения задачи ее можно отменить как завершенную. Добавляя задачу в список нужно указать текст задачи. Кроме этого можно добавить дополнительное описание. Чтобы не пропустить важные задачи можно добавить напоминание, которое уведомит пользователя о важных делах в заданное время. Для личного удобства задачам можно присваивать приоритет. Кроме этого имеются умные списки. Они создаются автоматически и содержат в себе выборку из всех созданных пользователем задачам по определенному критерию: задачи на сегодня, на эту неделю, выполненные и т. д. Кроме этого есть список, который называется Входящие. Он создается также как и умные списки автоматически. В это список обычно добавляются задачи, которые пользователь пока не может связать ни с одним существующим списком, но не хочет забыть про эту задачу. А уже после добавления через какое-то время пользователь можно перенести задачу в существующий список или же создать новый. Общие списки – это специальная категория списков, когда они принадлежат нескольким пользователям. Они имеют равные права в создании, редактировании и удалении задач из этого списка. Данный тип списков бывает довольно полезен, когда ведется какая-то совместная работа или небольшой проект. Сделать общим можно любой уже существующий список даже при наличии в нем задач.

А.А. Савченко (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. П.Л. Чечет, к.т.н., доцент.

## СТРУКТУРА СОВРЕМЕННЫХ СИСТЕМ КОНТРОЛЯ И УЧЕТА ЭНЕРГОРЕСУРСОВ

Решение проблем энергоучета на предприятии требует создания автоматизированных систем контроля и учета энергоресурсов (АСКУЭ), в структуре которых в общем случае можно выделить четыре уровня [1, 2], приведённых на рисунке 1.

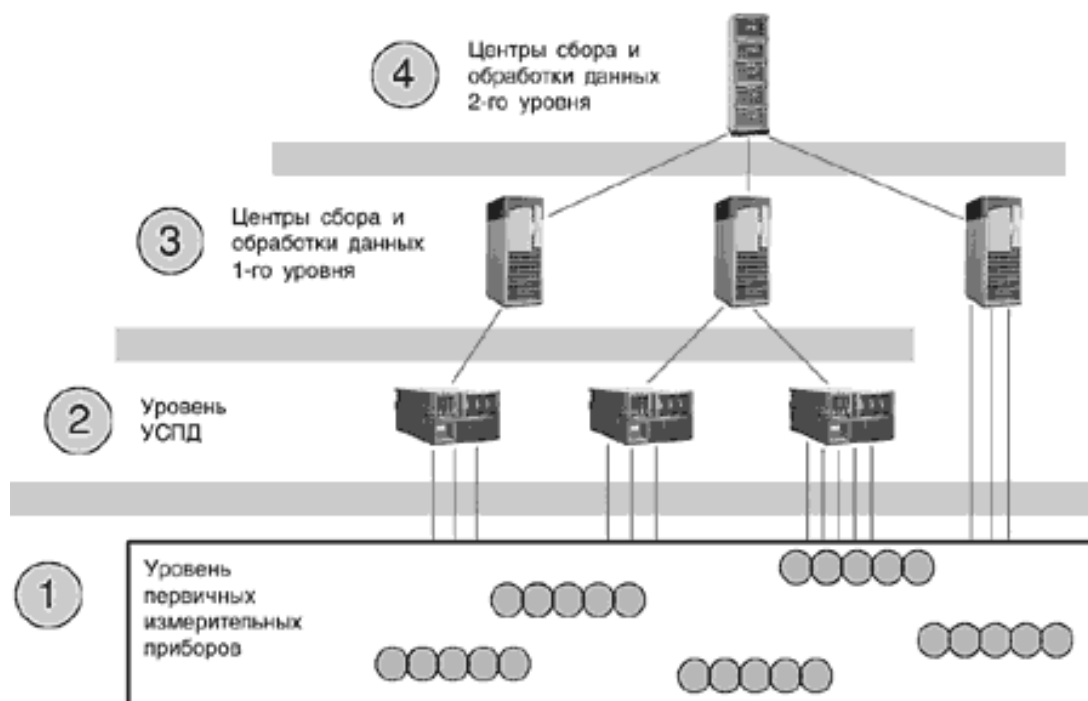


Рисунок 1 – Структура автоматизированных систем контроля

Первый уровень – первичные измерительные приборы (ПИП) с телеметрическими или цифровыми выходами, осуществляющие непрерывно или с минимальным интервалом усреднения измерение параметров энергоучета потребителей (потребление электроэнергии, мощность, давление, температуру, количество энергоносителя, количество теплоты с энергоносителем) по точкам учета (фидер, труба и т. п.).

Второй уровень – устройства сбора и подготовки данных (УСПД), специализированные измерительные системы или multifunctional программируемые преобразователи со встроенным программным обеспечением энергоучета, осуществляющие в заданном цикле интервала усреднения круглосуточный сбор измерительных данных с территориально

распределенных ПИП, накопление, обработку и передачу этих данных на верхние уровни.

Третий уровень – персональный компьютер (ПК) или сервер центра сбора и обработки данных со специализированным программным обеспечением АСКУЭ, осуществляющий сбор информации с УСПД (или группы УСПД), итоговую обработку этой информации как по точкам учета, так и по их группам – по подразделениям и объектам предприятия, документирование и отображение данных учета в виде, удобном для анализа и принятия решений (управления) оперативным персоналом службы главного энергетика и руководством предприятия.

Четвертый уровень – сервер центра сбора и обработки данных со специализированным программным обеспечением АСКУЭ, осуществляющий сбор информации с ПК и/или группы серверов центров сбора и обработки данных третьего уровня, дополнительное агрегирование и структурирование информации по группам объектов учета, документирование и отображение данных учета в виде, удобном для анализа и принятия решений персоналом службы главного энергетика и руководством территориально распределенных средних и крупных предприятий или энергосистем, ведение договоров на поставку энергоресурсов и формирование платежных документов для расчетов за энергоресурсы;

Все уровни АСКУЭ связаны между собой каналами связи. Для связи уровней ПИП и УСПД или центров сбора данных, как правило, используется прямое соединение по стандартным интерфейсам [3] (типа RS-485, ИРПС и т.п.). УСПД с центрами сбора данных 3-го уровня, центры сбора данных 3-го и 4-го уровней могут быть соединены по выделенным, коммутируемым каналам связи или по локальной сети.

### Литература

1. Гуртовцев, А. Комплексная автоматизация энергоучета на промышленных предприятиях и хозяйственных объектах / А. Гуртовцев // журнал "СТА". – № 3, 1999 г. – С. 44–45.
2. Быценко, С.Г. Инструментальное обеспечение рынка электроэнергии. Концепция создания автоматизированной системы контроля и управления энергопотреблением / С.Г. Быценко // Промышленная энергетика. – №№ 1, 2, 3, 4, 1998 г.
3. Касьян, В.Я. Организация проектирования автоматизированных систем контроля и учета энергии в Энергосбыте АО "Челябэнерго" / В.Я. Касьян, П.Л. Самсонов, П.А. Синютин. // Промышленная энергетика. – № 7, 1997 г.

**А.В. Сапанович (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.И. Кучеров**, м.т.н., старший преподаватель

## **РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗАЦИИ РАБОТЫ С РАСПИСАНИЕМ ЗАНЯТИЙ ФАКУЛЬТЕТА**

Составление расписания занятий и последующая работа с ним – это всегда важная задача факультета. Когда составлен первый вариант расписания, иногда сложно увидеть все ошибки и накладки в расписании. Является также проблемой оперативно предоставить каждому преподавателю его расписание: для этого ему обычно требуется просмотреть занятия всего факультета. В связи с этим весьма полезным является создание приложения для управления расписанием занятий, с помощью которого можно будет извлекать нужную информацию как для преподавателей, так для студентов, и которое позволит формировать отчёты по расписанию преподавателей, групп, занятости кабинетов, а также отслеживать в расписании накладки по преподавателям и кабинетам.

Учитывая, что в университете уже существует база данных всех дисциплин университета, преподавателей, групп, специальностей, кафедр, факультетов, а главное, учебных планов, был смысл разрабатывать такое приложение, которое будет взаимодействовать с уже существующей в университете базой данных и будет выводить расписание в приемлемом виде, а также будет в точности отвечать всем потребностям факультета.

Приложение основано на базе данных. В ней хранятся все необходимые данные для работы. Структура базы имеет достаточно сложную схему (рисунок 1). База данных была приведена в третью нормальную форму, что полностью избавило ее от избыточной информации. Все таблицы связаны между собой связью один-ко-многим. Она также полностью синхронизирована с базой данных факультета. Причем все эти данные берутся в соответствии с учебным планом. Так, например, приложение не позволит выставить для групп первого курса предметы, которые ведутся на пятом, также система контролирует и соответствие предметов, кабинетов выбранному факультету, выбрать дисциплину, предназначенную для других факультета и специальности будет нельзя. Но при этом функциональность по работе с данными учебных планов ограничена лишь чтением. Сделано это для исключения человеческого фактора при работе.

Основной задачей системы является максимально удобный и быстрый ввод информации в базу данных из уже составленного

расписания или проектирование расписания непосредственно в системе. Для этого в приложении были добавлены несколько ролей пользователей – составителя расписания и простого пользователя. Для них различается набор функциональности. Простой пользователь может лишь просматривать расписание и сохранять его на своем компьютере или мобильном устройстве. Для работы с расписанием такому пользователю нет необходимости в авторизации – весь функционал доступен при входе на сайте. Это дает возможность любому желающему увидеть расписание факультета, например, родители смогут контролировать график дня своего ребенка. Для составителя, вдобавок к вышеперечисленному, существуют возможности добавлять занятия в сетку, редактировать их либо удалять вообще. Для каждого пользователя с правами на внесение изменений в расписание также существуют свои ограничения в рамках системы. Составитель назначается для определенного факультета: он может заполнять расписание только для назначенного факультета. Расписание других пользователей он может только просматривать. Эти настройки производятся глобальным администратором приложения, вдобавок число пользователей с расширенными правами строго ограничено. Все это сделано для исключения возможности постороннего вмешательства в работу с приложением.

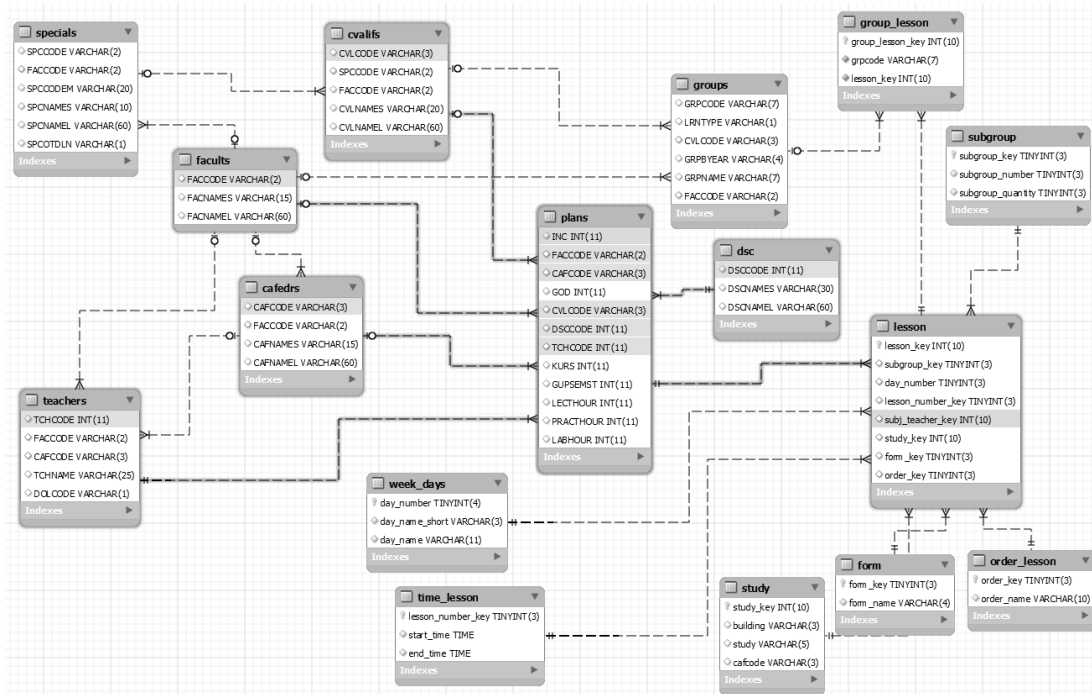


Рисунок 1 – Схема базы данных приложения

Фактически все компоненты делятся на два вида – на стороне клиента и на стороне сервера. На стороне сервера работают функции по

проверке данных и работе с базой данных. Проверка входных данных является одной из важнейших операций, поскольку именно через этот механизм системе можно нанести вред: например, с целью создать путаницу и нарушить нормальную работу факультета. Поэтому все данные перед обработкой проходят через специальные «фильтры»: регулярные выражения и предвыборку данных. Предвыборка данных представляет собой следующий механизм: сразу происходит поиск в таблицах ключа, который приходит от клиентской части приложения и если этот ключ существует, тогда по нему происходит выборка данных, в противном же случае система возвращает сообщение об ошибке. Также все входящие данные экранируются для предотвращения SQL-инъекций.

На стороне клиента разработка направлена была на разработку удобных форм работы с расписанием. Наиболее сложной из них является форма добавления занятий, которая спроектирована быть понятной, удобной и позволяющей быстро формировать занятия. Функция автоматического заполнения для основных полей ввода позволяет, например, по нескольким буквам найти необходимые названия предметов или фамилии преподавателей. Набор переключателей позволяет также разделить занятие на несколько подгрупп, выбрать тип занятия – лекция, практическое занятие, лабораторная работа. Но наиболее интересной функцией при добавлении занятий является поиск накладок в реальном режиме времени. Эта функция является не только функцией клиентской части – тут активно задействуется и серверная часть. Приложение отслеживает любые изменения и, когда они происходят, параллельно запускает запросы на проверку дубликатов – проверяет не назначено ли еще одно занятие у этого преподавателя на тоже самое время или не занята ли аудитория. Если дубликат найдет, то все ячейки с ошибкой подсвечиваются. В них указывается как ошибка в расписании, а дальше ее из режима редактирования можно исправить. Для удобства использования системы поиск накладок можно вызвать и при помощи отдельной кнопки. При каждом ее нажатии система будет перемещать курсор по ячейкам с дубликатами

Также для удобства просмотра расписания созданы все возможные фильтры, которые позволяют фильтровать занятия по группам, курсу, преподавателю или конкретному занятию, например, можно получить какие занятия проводятся на факультете первой парой в среду и т. д. Данный функционал также доступен и для обычных пользователей.

Отдельного внимания заслуживает и механизм создания отчетов. Поддерживаются два типа отчетов в формате Excel. Документы Excel добавляют удобную возможность редактирования. Также табличные

документы имеют фиксированную схему, что делает их предпочтительными для сохранения, если необходима последующая обработка отчета в другом приложении. Также большим плюсом к использованию отчета в табличном формате является и то, что некоторые мобильные устройства на платформе J2ME хорошо поддерживают формат Excel.

**С.В. Семенцов (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **С.Ф. Маслович**, к.т.н., доцент

## **ОБРАБОТКА РЕЗУЛЬТАТОВ ГЕОФИЗИЧЕСКИХ ИССЛЕДОВАНИЙ ПРИ ПРОЕКТИРОВАНИИ НЕФТЯНЫХ СКВАЖИН**

В рамках решаемой задачи необходимо было при проектировании нефтяных скважин автоматизировать расчет комплексов геофизических исследований скважины (ГИС) и виды исследований, входящие в эти комплексы. Каждое исследование должно быть привязано к конкретной колонне и к конкретному интервалу бурения.

Автоматизированный расчет проведение рациональных комплексов ГИС позволит сэкономить время проектировщикам скважин из института нефти БелНИПИ, а также позволит учитывать разнообразные факторы, которые могут повлиять на наличие того или иного комплекса в скважине.

Для оценки важности ГИС опишем основные задачи, решаемые при выполнении геофизических исследований в бурящихся скважинах:

- стратиграфическое и литологическое расчленение разрезов скважин. При решении данной задачи основной целью является геологическая увязка положения забоя скважины;
- контроль процесса цементации обсадных колонн и текущее состояние цементного камня;
- определение характера насыщения коллекторов в разрезе скважины; контроль траектории ствола и технического состояния скважин.

Чтобы выполнить поставленную задачу, необходимо было определить порядок выполнения всех комплексов с интервалами проведения исследований. Для чего использовались методические указания из института БелНИПИ, на основе которых я сформировал алгоритм расчета рационального проведения комплексов. Был создан справочник, который содержит формулы для описания интервалов и

критериев наличия различных исследований, в который и был занесен разработанный алгоритм.

Для реализации задачи использовался объектно-ориентированный язык программирования C# 2.0. В качестве базы данных использовалась объектно-реляционная система управления базами данных компании Oracle версии 11g Release 2, а так же язык программирования PL/SQL, процедурное расширение языка SQL.

В качестве анализатора выражений, который рассчитывал значения в формулах интервалов и формулах наличия использовался Evaluator 3.0.

**А.А. Семионов (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.И. Кучеров**, старший преподаватель

## **ЦВЕТКОРРЕКЦИЯ С ПОМОЩЬЮ КРИВЫХ**

Всем нравятся цвета в фильмах и на пленочной фотографии. Но на цифровых фотографиях чаще всего цвета совсем другие, более жесткие, переходы светотеней более резкие и так далее. А ведь так хочется картинки как в кино. Я постараюсь сделать универсальный фильтр с помощью кривых в программе Photoshop, который бы подходил для большинства фотографий.

Наиболее предпочтительной будет фотография, которая была снята широким светосильным объективом с размытием фона, так как она будет захватывать достаточно много и выглядеть более эффектно. Желателен мягкий рассеянный свет.

Если все эти условия соблюдены, можно рассчитывать на достойный результат. Исходная фотография показана на рисунке 1.



Рисунок 1 – Исходная фотография

Лучше начинать с синего канала, так как там происходят самые значительные изменения. Мы осветляем черную точку и глубокие тени, с



помощью дополнительной точки снижаем контраст в тенях, чтобы смягчить картинку, делаем тени синее, что в принципе является одной из основ цветовосприятия и киношного цвета: света должны быть желто-оранжевыми, тени – синими. Результат на рисунке 2.

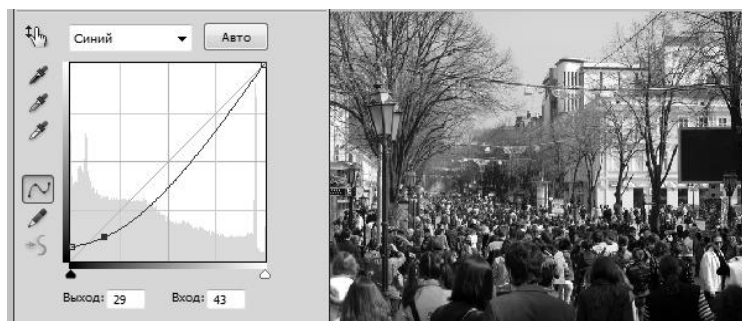


Рисунок 2 – Синий канал

Открываем зеленый канал и осветляем черную точку и тени, уменьшаем контраст в тенях, увеличиваем его немного в средних тонах и немного прижимаем белую точку, чтобы сделать света не такими зелеными. Результат на рисунке 3.

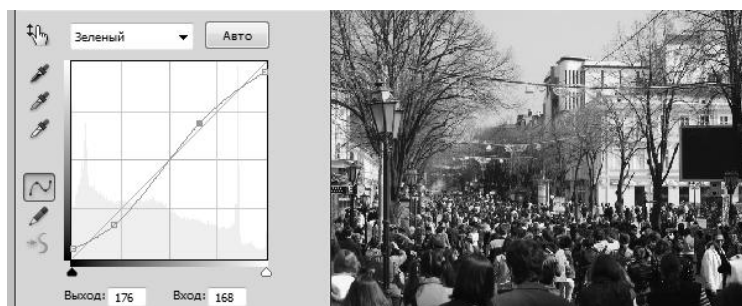


Рисунок 3 – Зеленый канал

В красном канале можно оставить почти все без изменений, разве что избавить фотографию от излишней теплоты, сместив кривую чуть вниз. Результат на рисунке 4.

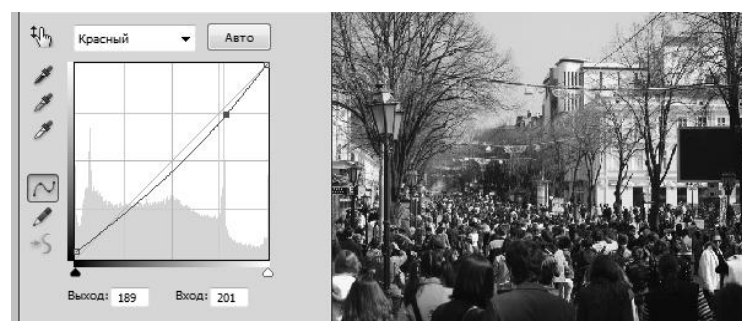


Рисунок 4 – Красный канал

Далее поднимаем яркость и уменьшим осветление светов. Также слегка усилим контраст, сдвинув черную точку правее.

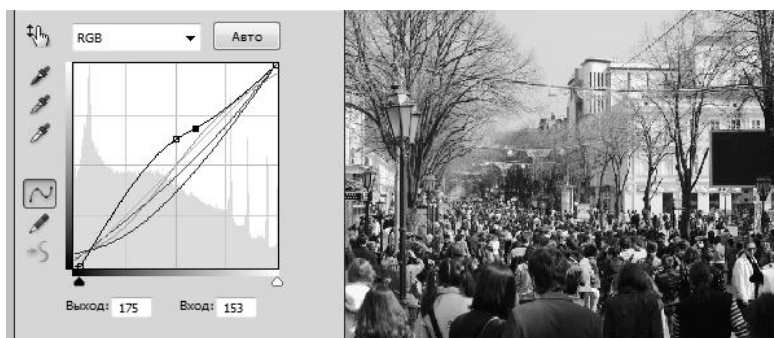


Рисунок 5 – RGB кривая

Используя описанный выше подход можно получить множество вариантов, и выбрать один, который нравится именно вам. Если мало контраста, можно сделать кривую более крутой. Если фотография кажется слишком желтой, можно поднять синюю кривую на этом диапазоне. Возможно, в самом конце будет не лишним подправить насыщенность, так как цифра выдает очень яркие цвета.

### Литература

1. Кирьянов, Д.А. Самоучитель Adobe Photoshop CS3 / Е.А. Кирьянова – Петербург: Издательство БХВ–Петербург, 2008. – 54 с.
2. Черников, С.В. Учимся Photoshop с нуля / К.С. Иваницкий – Москва: Издательство Лучшие книги, 2008. – 201 с.

**А.А. Семионов (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **А.И. Кучеров**, старший преподаватель

### ПРИМЕНЕНИЕ ЭФФЕКТА РЕВЕРБЕРАЦИИ ПРИ ОБРАБОТКЕ ЗВУКА

В этой статье рассказывается об интересном и полезном эффекте “RVerb” из пакета Waves – эффект реверберации. Он применяется практически при любой обработке звука. Его основное достоинство – придание объема и глубины звуку.

Реверберация – эффект, который вызывает большинство сложностей. С одной стороны вроде бы все просто – можно выбрать готовый пресет, например Hall, и все отлично. Но вот здесь-то и кроется отличие профессиональных записей от любительских. Профессионалы используют реверберацию не только для придания объема звуку, но и для

расстановки инструментов по глубине. Плюс к этому есть еще несколько секретов.

Чтобы понять, как правильно настроить эффект, рассмотрим его основные настройки на примере Waves RVerb.

*Predelay.* Предварительная задержка – один из самых главных параметров. Задаёт время, через которое к слушателю приходят первые отражения от стен помещения. Чем больше это время, тем больше расстояние до источника звука и тем больше помещение, в котором находится слушатель.

*Time.* Длительность реверберации (время затухания звука приблизительно на 60 дБ).

*Size.* Размер или объем имитируемого помещения.

*Diffusion.* Плотность – параметр, характеризующий, насколько плотно звучит реверберация. Более высокое значение диффузии увеличивает отражения, что приводит к более тяжелому звуку. Уменьшение диффузии приводит к тому, что звук становится более легким и прозрачным. На практике более высокое значение плотности подходит для ударных и перкуссии, меньшее – для клавишных и гитарных соло.

*Decay.* Время затухания. Настройка полностью аналогична настройке Time. В данном ревербераторе ее изменении, никакой разницы в звук не внесло. Как правило, в других ревербераторах присутствует одна из этих двух настроек.

*Early reflections.* Ранние отражения. Реверберация представляет собой серию отражений от поверхностей помещения, следующих друг за другом столь плотно, что мы не слышим их по отдельности. В этой серии самыми первыми появляются так называемые ранние отражения. Они приходят от поверхностей, которые расположены ближе всего к источнику звука. На слух они представляют короткий, плотный, задержанный на относительно небольшое время хвост.

*Reverb.* Реверберация – звук, который продолжает звучать в помещении после того, как затих исходный сигнал. Отличается от эхо, которое просто повторяет звук с некоторой задержкой, принося его издалека.

*Wet/Dry.* Настройка представляет собой управление отношением оригинального звука к звуку реверберации.

*Gain.* Настройка уменьшает или увеличивает громкость после обработки звука.

В качестве примера работы с эффектом, на рисунке 1 приведено окно с настройками Waves “RVerb”.

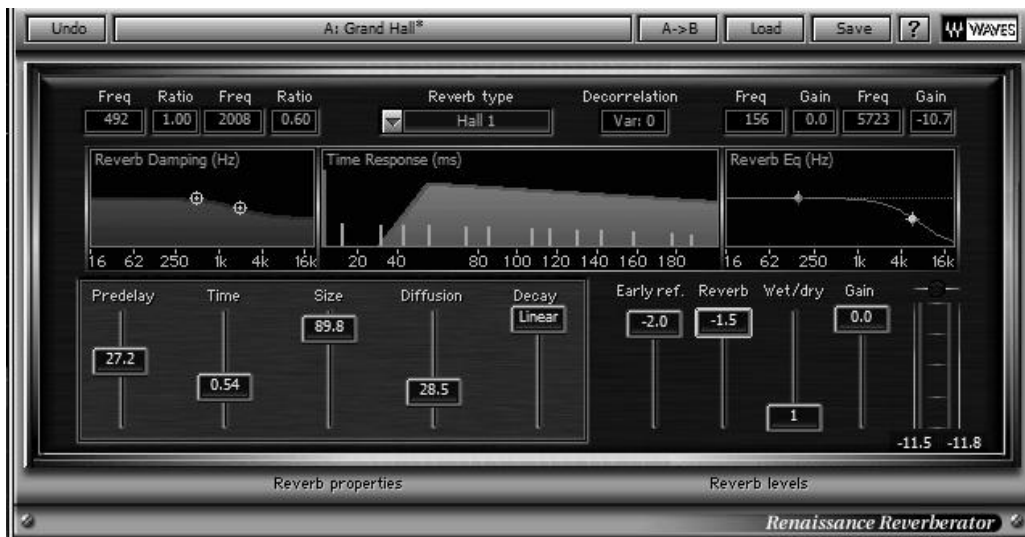


Рисунок 1 – Ревербератор

В заключение, хочу привести ряд советов касательно практического применения реверберации.

Чтобы достичь чувства общности пространства, рекомендуется использовать единый тип реверберации для всех инструментов в миксе.

Реверберация подходит для моделирования глубины сцены. Расположение инструментов с более глубокой реверберацией кажется более далеким от слушателя. И наоборот, инструменты или вокал, не имеющие реверберации вообще, ощущаются находящимися вблизи.

Интересный эффект можно получить, используя реверсивную реверберацию. Эффект реверсивной реверберации достигается путем обработки звука с помощью ревербератора и обращения по времени уже обработанного результата функцией Reverse в звуковом редакторе. Таким образом, создается ощущение плавного нарастания/приближения звука еще до начала его звучания, что довольно необычно.

Для улучшения звучания бас-бочки, баса и других низкочастотных инструментов не следует применять большое количество реверберации, более того, ее и вовсе не стоит использовать для этих инструментов.

К реверберации следует подходить с осторожностью – переборщив с ее количеством, можно полностью испортить микс.

### Литература

1. Музыкальная фабрика Fruity Loops Studio / Р.И. Петелин – Петербург: Издательство БХВ–Петербург, 2008. – 184 с.

2. Медведев, Е.А. Nuendo для музыкантов / Трусова В.И. – Петербург: Издательство БХВ–Петербург, 2009. – 87 с.

**С.В. Слука (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **В.Н. Леванцов**, старший преподаватель

## **НОВОЕ В ВЕРСИИ 8.2 ПЛАТФОРМЫ 1С:ПРЕДПРИЯТИЯ 8**

Версия 1С:Предприятие 8.2 – это принципиальное изменение архитектуры платформы версии 8, наиболее существенное с момента ее выпуска. 1С:Предприятие 8.2 полностью меняет весь слой работы с интерфейсом. Сюда относится и командный интерфейс, и формы, и оконная система. При этом не только меняется модель разработки пользовательского интерфейса в конфигурации, но и предлагается новая архитектура разделения функциональности между клиентским приложением и сервером.

В версии 8.2 принципиально переработан механизм установки, запуска и обновления платформы. Также выполнено значительное развитие кластера серверов 1С:Предприятия сразу по нескольким направлениям: масштабируемость, отказоустойчивость и динамическое распределение нагрузки.

### **Работа в режиме тонкого клиента и веб-клиента:**

Тонкий клиент и веб-клиент – это два новых клиентских приложения. Кроме привычного файлового доступа и подключения к серверу по локальной сети они позволяют подключаться к информационной базе по протоколу HTTP через специально настроенный веб-сервер. Тонкий клиент и веб-клиент обеспечивают работу пользователей в новом режиме – режиме управляемого приложения.

Тонкий клиент устанавливается на компьютер пользователя. При этом он имеет значительно меньший объем дистрибутива, чем старое клиентское приложение, и использует меньше аппаратных ресурсов. Тонкий клиент поставляется как в составе полного дистрибутива платформы, так и отдельным дистрибутивом.

Веб-клиент не требует предварительной установки. Он выполняется не в среде операционной системы, а в среде интернет-браузера (Microsoft Internet Explorer или Mozilla Firefox).

### **Работа пользователей через Интернет, в том числе по низкоскоростным каналам связи:**

Тонкий клиент и веб-клиент позволяют работать с информационной базой через Интернет в реальном режиме времени. Подключение осуществляется через специально настроенный веб-сервер (Apache или IIS).

Если такое подключение осуществляется через низкоскоростное интернет-соединение (например, GPRS), то эти клиентские приложения могут быть запущены в специальном режиме – режиме низкой скорости

соединения. При этом уменьшаются издержки за счет меньшего количества обращений к серверу: несколько запросов к серверу объединяются в один пакет, выполняется упреждающий сбор данных и т. д.

### **Режим управляемого приложения:**

В режиме управляемого приложения интерфейс не «рисует», а «описывается». Разработчик определяет только общую схему командного интерфейса и общую схему форм. Это описание платформа использует при построении интерфейса для конкретного пользователя с учетом различных факторов: прав пользователя, особенностей конкретного внедрения, настроек, сделанных самим пользователем.

Управляемое приложение значительно облегчает внедрение прикладных решений. Теперь можно не дорабатывать, а настраивать решение под специфику компании, не меняя при этом само прикладное решение.

### **Отказоустойчивый кластер серверов с динамической балансировкой нагрузки:**

В новой версии платформы реализована отказоустойчивость кластера серверов "1С:Предприятия" – основного компонента, обеспечивающего взаимодействие между пользователем и СУБД.

Теперь выход из строя или плановое отключение любого компьютера, входящего в состав кластера, не приведет к прекращению работы пользователей с информационной базой. Это достигается за счет горячего резервирования самого кластера серверов и за счет резервирования рабочих процессов, функционирующих внутри кластера.

Также кардинально повышена устойчивость системы к обрыву канала связи между пользователем и кластером серверов. Теперь в подавляющем большинстве случаев после восстановления соединения пользователь сможет продолжить работу с того «места», на котором она была прервана. При этом не потребуется повторное подключение к информационной базе.

### **Поддержка СУБД Oracle Database:**

Реализована поддержка новой СУБД – Oracle Database. Разработка велась при взаимодействии со специалистами корпорации Oracle. Теперь платформа "1С:Предприятия 8" поддерживает работу с пятью СУБД:

- Файловая;
- Microsoft SQL Server;
- PostgreSQL;
- IBM DB2;
- Oracle Database.

## **Механизм агрегатов для оптимизации построения аналитических отчетов:**

Использование механизма агрегатов позволяет значительно сократить время формирования отчетов. Это особенно важно для больших информационных баз, содержащих сотни тысяч и миллионы записей в регистрах.

Агрегаты используются для оборотных регистров накопления. Для любого регистра может быть создано несколько специализированных хранилищ, содержащих агрегированные данные регистра в различных разрезах, удобных для формирования отчетов в данной информационной базе. Система автоматически оценивает интенсивность работы пользователей с тем или иными разрезами информации и на основе накопленной статистики выбирает оптимальный состав поддерживаемых агрегатов.

### **Новый дизайн интерфейса:**

В режиме управляемого приложения используется новый дизайн интерфейса, ориентированный на комфортную и эффективную работу. Он соответствует современным тенденциям и в то же время учитывает сильные стороны прежнего интерфейса.

Работа с программой начинается с рабочего стола и панели разделов, расположенных в основном окне приложения. Рабочий стол содержит часто используемые документы, отчеты, справочники и т. п. Это своеобразный «помощник» пользователя. Рабочий стол вводит пользователя в курс дел, отвечает на его вопросы.

Панель разделов представляет основное разделение функциональности прикладного решения.

При активизации раздела вся его функциональность представляется в виде команд в двух панелях: панели навигации (слева) и панели действий (сверху).

## **Литература**

1. Хрусталева, Е.Ю. Разработка сложных отчетов в 1С: Предприятие 8. Система компоновки данных / Е.Ю. Хрусталева. – М.: ООО «1С-Публишинг», 2009 – 513 с.
2. Бойко, В.В. Проектирование баз данных информационных систем / В.В. Бойко, В.М. Савинков – М.: Финансы и статистика, 1989. – 351 с.
3. Радченко, М.Г. 1С: Предприятие. Практическое пособие разработчика. Примеры и типовые приемы / М.Г. Радченко, Е. Ю. Хрусталева – М.: ООО «1С-Публишинг», 2009 – 874 с.
4. Томас, К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / К. Томас, Б. Каролин – М.: Вильямс, 2004. – 143 с.

**М.А. Слюсарева (УО БрГТУ, Брест)**  
Науч. рук. **В.И. Хвещук**, к.т.н., доцент

## **ВОПРОСЫ АВТОМАТИЗАЦИИ ПЛАНИРОВАНИЯ ПРОИЗВОДСТВА АВТОМАТИЗИРОВАННЫХ СИСТЕМ**

В общем случае, процесс планирования производства автоматизированных систем (АС) предполагает построения набора планов с разным уровнем детализации. Это вызвано тем, что на разных стадиях жизненного цикла АС менеджеру проекта доступна информация с разным уровнем точности. В работе предложен набор средств и алгоритмов для планирования реализации АС на основе результатов проектирования архитектуры АС, т. е. для разработчика известна системная архитектура АС, которая представляется виде спецификации системных элементов системы (технический проект).

Процесс планирования представляет собой итеративную процедуру, которая заключается в построении плана работ по реализации АС и ее элементов в рамках заданных ресурсов на проект АС.

Предложенный комплекс построения и оценки плана реализации АС включает набор средств для описания:

1 Системная архитектура АС. Описываются информационные, программные, технические элементы, а также рабочие места пользователей:

– информационные элементы: файлы архивных и текущих документов, базы данных (локальные общие), архивы. Задается размещение информационных элементов по рабочим станциям АС;

– программные элементы: системные, инструментальные и прикладные программы. Для прикладных программ указываются взаимосвязи с информационными элементами (базами данных), а также возможные связи между прикладными программами. Задается размещение программных элементов по рабочим местам и станциям АС;

– технические элементы: рабочие станции АС: перечень устройств и ПЭВМ;

– рабочие места АС. Для каждого рабочего места определяются используемые информационные, программные и технические элементы АС.

2 Способы реализации отдельных элементов (повторное использование, закупка и поставка, создание нового элемента).

3 Группы системных элементов – совокупность элементов АС, реализуемая как единое целое (рабочее место, группа рабочих мест, очередь АС, вид средств АС и другие).



4 Способы реализации групп системных элементов.

5 Временные, финансовые и людские ресурсы на реализацию АС.

Для комплекса разработан набор алгоритмов планирования (включая диалоговое взаимодействие пользователя), позволяющий создавать и документировать план-график работ для организации процесса реализации АС и ее компонентов. Полученные результаты можно использовать для сборки элементов.

Предложенный комплекс реализуется в виде набора приложений в клиент-серверной архитектуре, обеспечивающего пользователям диалоговый интерфейс. Планируется использование предложенного приложения в рамках лабораторных работ и курсового проектирования по дисциплине «Проектирование автоматизированных систем».

**Е.С. Судакова (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Н.А. Аксёнова**, ассистент

## **ОСНОВНЫЕ ФУНКЦИИ И ОСОБЕННОСТИ LOTUS NOTES**

Одним из основных достоинств Lotus является система защиты информации. Защита данных и проверка прав доступа проводится многократно и постоянно. Существует несколько уровней, которые обеспечивают контроль доступа к данным, иначе неудача на любом из этих уровней защиты приведёт к несанкционированному чтению и/или разрушению информации.

При регистрации пользователя, сервер создает для него идентификационный файл, в котором хранятся публичный и приватный ключи, а также электронная подпись этого пользователя. Для обращения к серверу пользователь должен авторизоваться, используя этот файл. Все изменения, которые пользователь вносит в документы, подписываются им. Кроме того, Lotus позволяет сохранять электронные подписи всех когда-либо сохранявших документ пользователей.

При установке сервера указывается, кто имеет доступ к серверу, кто не имеет, кто может создавать на нем новые базы, реплики баз, запускать агентов (персональных, обычных и с внешними вызовами).

Также существует список управления доступом к базе данных (рисунок 3) (access control list, ACL) где определяется, кто имеет доступ к каждому из ресурсов и какими правами по их использованию он обладает (например, составление документов, чтение, запись, удаление). Ресурсы в списках управления доступом включают в себя серверы, отдельные базы данных, документы и поля в документах.

Профессиональные разработчики получают в свое распоряжение полный набор средств разработки Notes, позволяющих создавать стратегические системы. Сюда входят основные элементы для проектирования баз данных, полный набор встроенных средств программирования и полный набор средств интеграции с базами данных. Кроме того, Notes обеспечивает интеграцию с различными популярными инструментальными средствами других поставщиков.

Основные элементы проектирования Notes включают в себя формы, поля, виды, агенты и папки.

*Поля.* Формы строятся с помощью полей, таких как автор, организация, должность и т. д. Разработчик может назначить полям определенные типы данных: текстовый, числовой, формат времени, форматированный текст. Пользователи могут создавать новые поля и изменять атрибуты и свойства имеющихся полей с помощью ряда диалоговых окон.

*Формы.* Основным элементом при проектировании приложений Notes является форма. Она представляет собой средство для ввода и индикации информации. В Notes входит набор предопределенных форм.

*Виды.* Поля имеют очень важное значение для управления базами данных Notes, так как документы представляются пользователю отсортированными по значению в определенных полях. Виды разрабатываются пользователем для вывода списка документов по различным критериям (по состоянию, по фамилии, по организациям и т. д.). При представлении вида пользователю имена полей выводятся в качестве заголовков столбцов.

*Папки.* Разработчики и пользователи имеют возможность создавать папки, как личные, так и коллективного пользования, которые представляют собой "контейнеры" с иерархической структурой, позволяющие организовать документы в любой базе данных. Пользователи могут перемещать документы из видов в папки, где с ними удобнее будет работать, по принципу «перетащи и отпусти».

*Агенты.* Агенты представляют собой логическую часть приложений Notes, позволяя разработчикам автоматизировать выполнение прикладных задач на клиентах и серверах. Эти задачи включают в себя обработку данных, репликацию и службы передачи сообщений. Кроме того, пользователи могут поручить агентам управление информацией в базах данных Notes. Например, не написав не единой строчки кода, конечный пользователь может создать агента, который будет автоматически просматривать поступающую почту в поиске слов «urgent» («срочно») или

«important» («важно») в полях темы сообщений, и копировать отобранные послания в папку «hot issues» («важные вопросы»).

В распоряжении разработчиков приложений Notes имеется множество различных конструктивных элементов для их построения. В дополнение к описанным выше шаблонам, Notes включает в себя LotusScript, агенты и язык формул Notes.

*LotusScript.* LotusScript представляет собой совместимый с BASIC язык программирования. Это полностью объектно-ориентированная среда программирования с классами, методами и свойствами, предоставляющая разработчикам согласованный доступ ко всем средствам Notes. Как и любое мощное средство программирования, LotusScript включает в себя интегрированные средства просмотра объектов и отладчик.

*Язык формул Notes.* В Notes также входит собственный язык формул. Формулы аналогичны математическим формулам и состоят из переменных, констант и операторов, хотя результаты могут коренным образом отличаться. Например, разработчик может написать формулу, которая будет форматировать текст, превращая все первые буквы слов в заглавные, или формулу для объединения содержимого двух независимых полей в одном поле, которое и будет выведено пользователю. Notes поставляется с набором встроенных функций @functions, позволяющих проводить самые различные вычисления с данными различных типов.

*Навигаторы.* Иерархические папки и виды в панели поиска Notes предоставляют пользователю современный и интуитивно понятный способ доступа к информации, но разработчик приложений может пожелать настроить интерфейс Notes в целях еще большего упрощения метода просмотра приложения. Навигаторы, которые представляют собой новые конструктивные элементы в Notes R4, состоят из графических изображений и "горячих зон", которые позволяют разработчику управлять ходом исполнения приложения Notes. Навигаторы – это наглядный интерфейс, с помощью которого разработчики могут руководить действиями пользователя приложения Notes. Например, можно создать навигатор, отображающий карту территорий, на которой компания осуществляет продажи. При щелчке мышкой на горячей зоне, такой как, например, город на карте, разработчик может предусмотреть вывод документа и/или вида Notes. С помощью навигаторов разработчики имеют возможность создавать приложения Notes, согласующиеся с понятиями и представлениями конечного пользователя.

**Е.С. Судакова (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Н.А. Аксёнова**, ассистент

## **РАЗРАБОТКА XPages-СТРАНИЦ С ИСПОЛЬЗОВАНИЕМ JAVASCRIPT В LOTUS NOTES**

Разработка XPages-страниц, в сущности, базируется на двух традиционных элементах дизайна: формах (form) для определения данных и представлениях (view) для вывода данных. Однако это не является абсолютным требованием. Для перемещения данных между элементами управления и хранилищем данных Lotus Domino можно использовать JavaScript, и это несложно.

Возможно, некоторым не хочется использовать JavaScript по различным причинам:

- формы и представления позволяют визуально определять сложные структуры;

- после объединения необходимых форм и представлений процесс разработки протекает довольно легко. Например, связывание формы с элементом управления на XPage состоит просто в перемещении поля на элемент управления;

- если XPages помещается в существующее приложение IBM Lotus Domino Designer, то, наверное, захочется полностью использовать возможности существующих форм и представлений.

Но новичкам в Lotus Domino Designer для изучения форм и представлений требуется время. И даже если есть опыт работы с Lotus Domino Designer, формы и представления потребуют выхода за рамки парадигмы XML. При использовании JavaScript в XPages в вашем распоряжении обширная библиотека инструментов для облегчения управления пользовательской средой и серверным хранилищем данных. Если вы знакомы с LotusScript, вам известны NotesSession, NotesDatabase, NotesDocument и многие другие объекты Domino Objects. Такие же объекты доступны и в JavaScript, но работать с ними в исходном коде проще.

JavaScript можно использовать, например, для возврата значения вычисляемого поля. Можно подключить JavaScript к событию onclick кнопки. Можно использовать JavaScript для определения значения (true или false) свойства Visual элемента управления.

Элемент дизайна XPages теперь инкапсулирует весь пользовательский интерфейс. Для определения данных можно связать элементы управления XPages с полями форм. В качестве альтернативы можно обращаться к данным посредством JavaScript и объектов Domino Objects.

Аналогично можно использовать представления для отображения и доступа к наборам документов, но также можно использовать таблицы данных и JavaScript.

**Д.А. Тишко (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **В.Н. Леванцов**, старший преподаватель

## **РАЗРАБОТКА БАЗЫ ДАННЫХ АВТОМАТИЗАЦИИ УЧЁТА ОБОРУДОВАНИЯ ДЛЯ РУП «ГОМЕЛЬЭНЕРГО»**

В настоящее время, несмотря на повышение компьютеризации общества, в сфере образования до сих пор нет средств, позволяющих в достаточной мере автоматизировать процесс ведения документации и отчетности.

Одной из составных задач можно рассматривать проблему составления графика проверок оборудования, а так же оперативную корректировку графика при возникновении необходимости в этом.

О своевременности и актуальности рассматриваемой проблемы говорит тот факт, что большую часть своего времени технологи предприятий тратят на оформление различной документации и отчетов. Огромное количество подразделений и отсутствие предложений в данной сфере гарантируют высокую потребность в данном продукте.

Базы данных (БД) составляют в настоящее время основу компьютерного обеспечения информационных процессов, входящих практически во все сферы человеческой деятельности.

Действительно, процессы обработки информации имеют общую природу и опираются на описание фрагментов реальности, выраженное в виде совокупности взаимосвязанных данных. Базы данных являются эффективным средством представления структур данных и манипулирования ими. Концепция баз данных предполагает использование интегрированных средств хранения информации, позволяющих обеспечить централизованное управление данными и обслуживание ими многих пользователей. При этом БД должна поддерживаться в среде ЭВМ единым программным обеспечением, называемым системой управления базами данных (СУБД). СУБД вместе с прикладными программами называют банком данных.

Одно из основных назначений СУБД – поддержка программными средствами представления, соответствующего реальности.

Предметной областью называется фрагмент реальности, который описывается или моделируется с помощью БД и ее приложений. В предметной области выделяются информационные объекты –

идентифицируемые объекты реального мира, процессы, системы, понятия и т. д., сведения о которых хранятся в БД.

Этапам реализации баз данных соответствуют уровни описания предметной области: реальность в том виде, как она существует; концептуальное описание реальности; представление описания в виде формального текста и физическая реализация БД на машинных носителях.

Для ввода в ПК полученное описание должно быть представлено в терминах специального языка описания данных, который входит в комплекс средств СУБД.

Простое (элементарное) данное – это наименьшая семантически значимая поименованная единица данных (например, серийный номер, модель, место установки и т. д.). Значения простого данного описывает представленную им характеристику объекта для каждого экземпляра объекта. Имена простых данных хранятся в описании БД, в то время как их значения запоминаются в самой БД.

Совокупность простых данных можно объединить в составное данное двумя способами. Во-первых, можно соединить несколько разнотипных данных. Например, данные ПРОВЕРКИ состоят из данных Код\_проверки, Служба, Подстанция, Оборудование, ДатаПроверки, ПериодПроверки, СрокПроверки. По этому принципу образуется структурное данное или данное типа структура. Описание структуры состоит из перечисления ее составных частей, значение – из значений составляющих ее данных. Во-вторых, составное данное может объединять совокупность однотипных данных (список оборудования, список подстанций и т.п.). Составные данные этого типа называется массивом. В описании массива достаточно указать описание одного элемента, значение массива представляется однородным списком значений его элементов.

В общем случае составные данные представляют собой объединенную под одним именем совокупность данных любых типов, в том числе структур и массивов, с произвольной глубиной вложенности составных данных.

Элементы массива могут идентифицироваться ключом – данным, значения которого взаимно однозначно определяют экземпляры элементов.

Составные типы данных представляют иерархические связи значений данных в БД. Представление предметной области в виде структур данных с иерархическими связями носит название иерархической модели данных. В общем случае в базе могут быть определены также сетевые связи, позволяющие описать сеть – ориентированный граф произвольного вида. Представление предметной области в виде сетевых структур данных общего вида называется сетевой моделью данных. Сетевые связи реализуются путем отождествления отдельных данных БД.

## Литература

1. Симонович, С.В. Информатика. Базовый курс / С.В. Симонович. – М.: Питер, 2005. – 640 с.
2. Бекаревич, Ю.Б. Самоучитель Microsoft Access 2002 / Ю.Б. Бекаревич. – М.: БХВ – Петербург, – 2002. – 720 с.
3. Савельев, В.А. Персональный компьютер для всех (кн.3), Создание и использование баз данных / В.А. Бекаревич. – М.: Высшая школа, – 1991. – 160 с.

**А.А. Трибилустов (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Н. Леванцов**, старший преподаватель

## СИСТЕМА УПРАВЛЕНИЯ ПРОЕКТАМИ И ЗАДАЧАМИ JIRA

Представляя продукт JIRA, нужно сказать, что жизни любой компании бывает момент, когда количество дел, которое вынуждены контролировать сотрудники и особенно руководство становится таким что превосходят возможности человеческой памяти.

При решении многосложных задач бывает момент, когда сотрудники и управляющие не могут видеть проект в целом, теряется из памяти необходимость сделать те или иные работы.

Бывает и другая ситуация подобная этой – отдел, состоящий более чем из 15–20 сотрудников работающих одновременно над одной задачей – сложная структура для управления, и без специальных средств проконтролировать их работу чрезвычайно сложно. Устаревший метод – бесконечные совещания и доклады – только усугубляет проблему, так как отвлекает от ведения главного дела.

Менеджеры всего мира ищут способы как не забыть что-то в потоке дел, при этом вовремя сообщить команде важные новости, поставить всем задачи, проследить за выполнением, отыскать и узкие места и принять меры, и, в конце концов, успешно завершить работу в срок. Все давно поняли – для этого не достаточно записей на квадратиках из желтой бумаги! Не помогут и блокноты-органайзеры – нужно интеллектуальное, удобное программное обеспечение, доступное всем, не отнимающее много усилий, простое и понятное и вместе с тем очень гибкое и легко настраиваемое.

Представленное здесь программное обеспечение JIRA, разработанное австралийской компанией Atlassian – это web-базированное средство для управления проектами и задачами. JIRA может применяться во

всех случаях, когда необходимо организовать работу сотрудников, эффективно назначать им задачи, иметь мгновенные средства контроля выполнения,

JIRA это программа, претендующая на звание стандарта де-факто в этой области, и ее применение очень широко.

JIRA это продукт, предназначенный для организации процесса контроля запросов и задач, имеющий часть функциональности обычно присущей большим и дорогим системам управления проектами.

Ключевыми понятиями в JIRA являются проекты и задачи. Задачи создаются в проектах, для выполнения задач назначаются исполнители. Задачи могут быть разного типа и иметь подзадачи, задачи могут быть связанными с другими задачами. Статус задач меняется в процессе их выполнения.

Вы можете организовать контроль разработки проектов, раздав задачи исполнителям, вы можете определить свой собственный метод движения заданий – от создания к исполнению и контролю результатов, сконфигурировать правила уведомления о событиях всех участников процесса, управлять правами доступа пользователей и делать многое другое.

JIRA приносит большой эффект любой организации, деятельность которой можно интерпретировать как выполнение каких-либо проектов и задач, имеющих тематические и временные рамки.

Главное преимущество этого продукта в его ни с чем не сравнимой способности настройки под ваши нужды.

Например, в финансовой сфере, вы можете организовать процесс оформления кредита, от заявки, к вводу необходимых данных, к принятию решения и так далее.

В сфере государственного управления – вы можете создать задания, определить сроки выполнения, присоединить документы, организовать процесс прохождения задания между сотрудниками и проконтролировать результат.

Ключевыми понятиями в JIRA являются проекты и задачи. Проекты служат для группирования задач. Задачи создаются в проектах, для выполнения задач назначаются исполнители. Задачи могут быть разного типа и иметь подзадачи, задачи могут быть связанными с другими задачами. Статус задач меняется в процессе их выполнения.

В JIRA многие понятия связаны с определенными проектами. Проект связанного с ним лидера проекта, URL сайта проекта, схему рассылки уведомлений, схему контроля доступа – очень гибкий механизм контроля доступа пользователя к задачам проекта.



Каждому проекту можно сопоставить свою схему движения задачи (документооборот). Каждому проекту можно сопоставить свой вид экрана – "собрать" требуемый экран из возможных компонентов.

Также, JIRA формирует отчеты по каждому проекту.

Задачи создаются в проектах. Задачи имеют типы, например: Задание, Ошибка, Новая идея. Можно создавать и свои типы задач. При описании каждого типа задачи имеется возможность управления набором полей.

JIRA позволяет отыскивать задания по всем критериям и по пользовательским полям, создавать фильтры, которые можно сохранить и использовать вновь, а также сделать общедоступными и организовать автоматическую рассылку результатов работы фильтров членам рабочей группы.

Для организации работы с пользователями JIRA имеет группы пользователей и роли.

JIRA имеет систему контроля доступа пользователей к проектам, задачам и функциям, основанную на членстве пользователей в группах и ролях.

Так, для каждого проекта, есть возможность управления доступом каждой группы пользователей к каждому действию. Также, есть возможность сформировать набор допусков в "роль".

Типичное простейшее разделение ролей в JIRA включает в себя роли:

- администратора;
- руководителя проекта;
- сотрудника работающего над проектом;
- других сотрудников.

Но возможности JIRA этим не ограничиваются, возможно создание специальных ролей, например таких, которым возможно только чтение задач одного типа но невозможно другого.

Задачи JIRA в каждый момент времени имеют определенный статус. Возможные действия с задачами, имеющими тот или иной статус, определяется встроенной системой управления движением задач. Здесь приведена схема простейшего описания движения задачи.

В JIRA имеется возможность создания столь сложной схемы движения задачи, какая нужна. Схема движения задачи может быть своя для каждого отдела, проекта, типа задачи.

Схема движения редактируется встроенным редактором. Редактируя движение задач, создавая новые статусы задач (события) и определяя возможные действия, можно организовать любую работу.

Движение задачи можно сделать зависимым от условий, применять

логику И/ИЛИ, выполнять определенные действия на каждом этапе движения задачи.

Задачи JIRA в каждый момент времени имеют определенный статус. Пользователи информируются по e-mail в случае любых действий с заданиями, для этого служит настраиваемая система нотификации пользователей. Совместно с системой управления движением задачи и настраиваемыми рассылаемыми фильтрами это позволяет очень эффективно информировать всех заинтересованных лиц о ходе выполнения задачи.

**Т.Ф. Умаров (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. П.Л. Чечет, к.т.н., доцент

## **ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ НАКОПИТЕЛЕЙ С ИСПОЛЬЗОВАНИЕМ RAID-МАССИВОВ**

RAID (избыточный массив независимых жёстких дисков) – массив из нескольких дисков, управляемых контроллером, воспринимаемых внешней системой как единое целое. Служит для повышения надёжности хранения данных и/или для повышения скорости чтения/записи информации.

Существует некоторое количество конфигураций системы RAID. Конфигурации этой системы называют уровнями. Перечислим самые распространенные из них.

RAID 0 (Striping – «чередование») – массив нулевого уровня, обеспечивает повышенную производительность, однако не обеспечивает отказоустойчивость. Массив собирается из двух и более дисков, информация разбивается на равные блоки данных и записывается на несколько дисков одновременно. За счет того, что запись/чтение происходит одновременно на несколько дисков, достигается повышенная производительность.

Данный уровень не обеспечивает отказоустойчивости, т.к. массив не создает копии блоков данных на разные диски, следовательно, все данные в данном массиве являются уникальными. В случае выхода из строя одного из дисков, данные будут утеряны. На практике RAID нулевого уровня применяется, когда необходимо обеспечить максимальную производительность дискового массива и информация, размещенная в массиве, не представляет важности (например, операционная система, файл подкачки и др.).

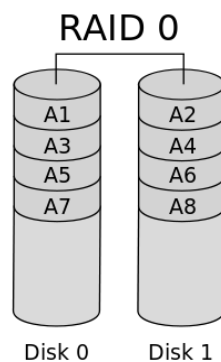


Рисунок 1 – Схема размещения данных нулевого уровня

RAID 1 (Mirroring – «зеркалирование») – зеркальный дисковый массив, состоящий из двух дисков. Информация дублируется на оба диска одновременно, оба диска являются полными копиями друг друга. Данная конфигурация не увеличивает быстродействие системы в целом. Первый уровень имеет высокую надежность, в случае выхода из строя одного из дисков, информация в полном объеме остается на другом диске, система остается в рабочем режиме.

Для систем RAID уровня 1 и выше, применяют диски горячего резерва (Hot-spare). В случае отказа одного из дисков, контроллер автоматически переходит к использованию резервного диска вместо вышедшего из строя диска.

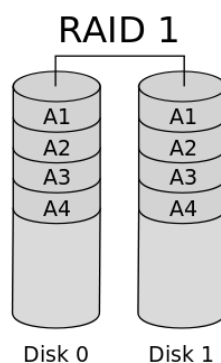


Рисунок 2 – Схема размещения данных первого уровня

Различают несколько видов дисков hot-spare:

- локальные: диск принадлежит к конкретному массиву и используется для подмены вышедшего из строя диска только в заданном массиве, если в системе несколько массивов и диск выходит из строя в соседнем массиве, то локальный для другого массива диск не используется для подмены;
- глобальные: диск не принадлежит ни к одному массиву и может быть использован для подмены вышедшего из строя диска в любом из массивов;

– групповые: в этом случае некоторые массивы объединяются в группу, в пределах которой может использоваться резервный диск.

Недостатком первого уровня является стоимость, устанавливается минимум два диска, при этом объем дискового массива равен только одному.

RAID 5 – в отличии от других систем отличается экономичностью. Уровень применим минимум при трех дисках в наличии. Дисковый объем памяти пятого уровня вычисляется следующим: образом объем диска умножается на  $n-1$ , где  $n$  – общее количество дисков. Блоки данных и контрольные суммы циклически записываются на все диски массива. Под контрольными суммами подразумевается результат операции XOR (исключающее или). Плюсом системы пятого уровня является финансовая часть, отказоустойчивость системы обеспечивается всего одним диском. Так же скорость чтения с массива увеличивается, т.к. потоки данных с нескольких дисков могут обрабатываться параллельно.

Минусы уровня 5 это нагрузка на контроллер массива, возникающая в результате обработки операции XOR в момент чтения и записи, отказоустойчивость всей системы равна одному диску. В случае одновременного выхода из строя нескольких дисков информация будет утеряна.

RAID 10 – массив состоящий как минимум из четырех дисков. Представляет собой массив нулевого уровня, но в качестве отдельных дисков используются массивы первого уровня. В массив рекомендуется включать резервные диски из расчёта один резервный на пять рабочих.

Десятый уровень имеет высокую производительность, а также и достаточную отказоустойчивость.

Система RAID применяется не только для повышения производительности и надежности хранения данных, но и для хранения крупных неделимых объектов, таких как, СУБД. При таком объединении некоторые диски могут передавать данные, в то время как другие позиционируют блок головок или ждут подхода нужного сектора к головке.

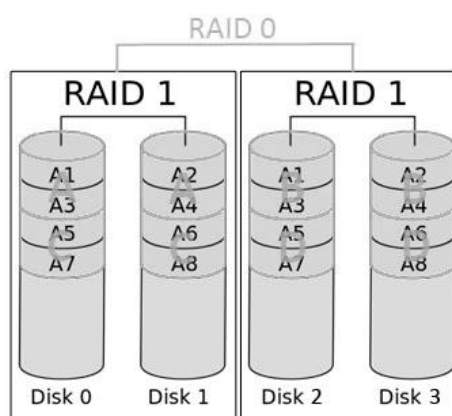


Рисунок 3 – RAID 10

## **Литература**

1. Кенин, А. Самоучитель системного администратора / А. Кенин – СПб: БХВ-Петербург, 2008. – 640 с.
2. Таненбаум, Э. Современные операционные системы / Э. Таненбаум – СПб: Питер, 2013. – С. 423–441.

**Д.Л. Фащенко (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **Е.А. Левчук**, к.т.н., доцент

### **ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ПРОЕКТА АСУ «ДКР»**

В условиях развивающихся рыночных отношений остро встают вопросы повышения эффективности работы железнодорожного транспорта, снижения издержек, уменьшения стоимости перевозок, соблюдения сроков доставки грузов и обеспечения их сохранности. Эти обстоятельства выдвигают принципиально новые требования к системе управления перевозками.

Был разработан проект АСУ «ДКР» для комплексной автоматизации и информационной поддержки бизнес-процессов работы с клиентами в области грузовых перевозок и коммерческой работы АО «НК «ҚТЖ».

В ходе выполнения проекта было рассмотрено понятие перевозочного процесса на железнодорожном транспорте и способе его автоматизации путем внедрения информационных технологий.

Были рассмотрены требования к разработке портлета поиска уведомлений, взаимодействие уведомлений о завершении грузовых операций с другими документами в процессе организации перевозки. Проработаны функциональные возможности разрабатываемого портлета и спроектирован эскиз страницы поиска уведомлений. Были описаны сценарии страницы процесса поиска уведомлений, а также рассмотрены таблицы для хранения информации об уведомлениях.

В результате в рамках проекта АСУ «ДКР» был разработан портлет «Поиск уведомлений о завершении грузовых операций».

**Д.Л. Фащенко (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **Е.А. Левчук**, к.т.н., доцент

### **АВТОМАТИЗАЦИЯ ПЕРЕВОЗОК НА ЖЕЛЕЗНОДОРОЖНОМ ТРАНСПОРТЕ В ПРОЕКТЕ АСУ ДКР**

Создание автоматизированной системы управления перевозками предполагает наличие единого информационного пространства для

взаимодействия органов управления транспортным комплексом и пользователей рынка транспортных услуг; системы информационного сопровождения перевозок в международных транспортных коридорах; государственного информационного ресурса транспортного комплекса страны; системы сбора и обработки статистической информации по транспортному комплексу. Важнейшим направлением развития транспортной отрасли остается развитие международных транспортных коридоров. Одним из элементов, определяющих развитие этого направления, является предоставление клиентуре в реальном масштабе времени информации о местонахождении груза и прогнозе его доставки. При этом сложнейшая система железнодорожного транспорта требует оптимизации управления с использованием новейших технологий.

Высокий уровень требований к эффективности управления перевозками на железнодорожном транспорте определяет потребность в высоком уровне его информатизации. Информационные технологии сегодня становятся не просто средствами поддержки управления, а одним из важнейших элементов инфраструктуры железных дорог. Использование сети Internet решает задачу выхода во всемирные сети телекоммуникаций.

Оптимальное использование возможностей информационной системы железных дорог в интересах всего транспортного комплекса страны позволяет существенно снизить затраты на управление и связь при организации и осуществлении внутренних и международных перевозок, обеспечивает существенное повышение качества транспортных услуг. В качестве приоритетных направлений внедрения информационных технологий определяются:

- информационная интеграция на железнодорожном транспорте на основе сетевых технологий с целью обеспечения глобального мониторинга движения грузов;
- совершенствование внутреннего и внешнего документооборота;
- внедрение электронных форм контрактов, перевозочных документов и платежей;
- решение проблем сокращения простоя подвижного состава на пограничных переходах путем активного внедрения технологий, основанных на электронном документообороте;

Актуальность разработки проекта АСУ ДКР обусловлена тем, что удельный вес железных дорог в общих грузовых перевозках постоянно растет и создание автоматизированных систем управления (АСУ) обеспечивает повышение эффективности использования подвижного состава, улучшение качества работы железнодорожных станций, рост конкурентоспособности железнодорожного транспорта.

Основным назначением проекта является комплексная автоматизация и информационная поддержка бизнес-процессов работы с клиентами в области грузовых перевозок и процессов грузовой и коммерческой работы всех уровней АО «НК «ҚТЖ», упрощение сложной системы взаимодействия различных подразделений АО «НК «ҚТЖ» и ее клиентов в области организации грузовых перевозок и их выполнения.

АСУ ДКР предназначена для автоматизации процессов:

- формирования заказов на перевозки и организации непрерывного слежения за ходом их выполнения;
- оформления заявок на перевозки грузов, перевозочных документов с минимальными затратами времени;
- выполнения функций организации и учета реализации технологических операций по каждому виду грузовой работы на станциях, а также на путях промежуточных станций, открытых для выполнения грузовых операций;
- заполнения грузовой документации;
- формирования отчетности по данному виду деятельности;
- интеграции сводной информации и выдачи различных видов оперативной отчетности, отражающей показатели грузовой работы с детализацией ее по регионам, станциям, клиентам, перевозчикам (при необходимости), роду вагонов, принадлежности подвижного состава;
- ведения баз данных АСУ ДКР и эталонной НСИ.

АСУ ДКР предоставляет специалистам и руководителям дороги оперативную и достоверную информацию о производимых и выполненных грузовых перевозках, обеспечивает информационную поддержку принятия обоснованных управленческих воздействий при решении основных задач по повышению эффективности работы и конкурентоспособности железнодорожного транспорта на рынке услуг, увеличению объемов и качества перевозок, сокращению транспортных издержек.

Основными целями создания АСУ ДКР являются:

- повышение эффективности управления в области коммерческой и грузовой работы;
- достижение качественно нового уровня обслуживания клиентов железной дороги;
- обеспечение взаимного предварительного информирования участников перевозок о ж/д грузах;
- ускорение и упрощение процедуры оформления грузов в таможенных и других органах государственного контроля на пограничных железнодорожных станциях, станциях отправления и назначения;
- повышение пропускной способности на международных железнодорожных пунктах пропуска;

– улучшение координации деятельности железной дороги с другими участниками перевозочного процесса;

– повышение эффективности функционирования АО «НК «ҚТЖ», увеличение прибыльности и повышение рентабельности грузовых перевозок и коммерческой деятельности.

Архитектурные решения реализации АСУ ДКР обсуждаются в докладе.

**А.Г. Флерко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.Н. Куличенко**, старший преподаватель

## **АВТОМАТИЗАЦИЯ ПЕРЕДВИЖНЫХ ОТДЕЛЕНИЙ ПОЧТОВОЙ СВЯЗИ ПРЕДПРИЯТИЯ «ЛЕЛЬЧИЦКИЙ РАЙОННЫЙ УЗЕЛ ПОЧТОВОЙ СВЯЗИ»**

Практически все современные крупные предприятия и фирмы по всему миру для улучшения своей работы в целом и взаимодействий внутри коллектива в частности создают компьютерные локальные сети. Благодаря организации этих так называемых интранет-систем облегчается передача информации из одного отдела в другой, повышается надёжность хранения данных при общем доступе всех сотрудников к ним и так далее.

Республиканское унитарное предприятие «Белпочта» предоставляет услуги почтовой связи для населения, органов государственного управления, предприятий и организаций.

Не секрет, что почта всегда была одним из важнейших центров притяжения в сельских населенных пунктах. Но в связи с сокращением числа жителей в селах и деревнях, появлением интернета, мобильной связи, почтовые отделения стали терять клиентов, переходить на сокращенный рабочий день, а некоторые и вовсе закрылись. Использование передвижных отделений почтовой связи (ПОПС) позволит решить проблему доступности почтовых услуг для жителей малонаселенных и удаленных населенных пунктов.

Мобильные отделения предназначены для оказания высококачественных услуг почтовой связи в удаленных, труднодоступных и малонаселенных местах, где по той или иной причине отсутствует инфраструктура, и нет стационарных отделений почтовой связи. В передвижных отделениях можно воспользоваться всем спектром почтовых услуг. Это и денежные переводы, и доставка почтовых отправок (периодическая печать, телеграммы, письма, бандероли, посылки), оплата



коммунальных платежей. Смонтированы ПОПСы на базе полноприводной «Газели» повышенной проходимости и УАЗ, что позволит им в любую погоду добраться до самых отдаленных уголков района. Машины оснащены системой мониторинга транспортных средств. В салоне такой машины создано рабочее место оператора, оснащенное компьютером с организацией канала связи через мобильного оператора по технологии GPRS, а также витрина с почтовой продукцией и товарами повседневного спроса.

GPRS – это беспроводной стандарт, призванный заложить основу для большого количества функций передачи данных при помощи пакетной передачи. Это новые, уникальные услуги операторов GSM, абоненты которых будут платить за использование только тех ресурсов, которыми они пользуются. Основным ресурсом оператора сотовой связи – частотный диапазон – можно будет делить между многими пользователями одновременно, поскольку стандарт GPRS поддерживает одновременную передачу данных для гораздо большего числа пользователей.

GPRS – это эффективная технология, позволяющая производить высокоскоростную пакетную передачу данных, необходимую большинству операторов сотовой связи для обеспечения пользователей соответствующими возможностями. GPRS разработан с целью расширить мобильный доступ конечных пользователей к данным, делая постоянное соединение возможным и доступным по цене, а скорость передачи данных значительно выше. GPRS не просто дополнит существующие возможности передачи данных, предоставляемые операторами GSM, но и обеспечит внедрение услуг передачи данных, характерных для сетей сотовой связи «третьего поколения», за несколько лет до их появления.

Основное назначение GPRS – обеспечить эффективный, дешевый, пакетно-ориентированный доступ к услугам сети Интернет по принципу «от посылающего к получателю». GPRS позволяет абонентам использовать радиоканалы одновременно и создает перекрывающуюся пакетную сеть с узлами услуг доступа в Интернет. Конечная цель построения сети с такой архитектурой – предоставление недорогого доступа к данным при минимальном воздействии на MSC. GPRS задействует два новых основных сетевых узла GSM PLMN: узел поддержки услуг GPRS (Serving GPRS Support Node, SGSN) имеет тот же уровень иерархии, что и MSC. SGSN отслеживает расположение мобильных пакетных передатчиков, обеспечивает защитные функции и контроль доступа. SGSN соединяется с BSS, используя Frame Relay – сеть коммутации фреймов (пакетов второго, канального уровня). Узел шлюзовой поддержки GPRS (Gateway GPRS Support Node, GGSN) взаимодействует с внешними

сетями пакетной передачи данных (Packet Data Networks, PDNs), обеспечивая проводку данных к MS и пересылку данных с мобильных передатчиков по назначению. GGSN связан с внешними сетями с пакетным переключением и соединяется с узлами SGSN по сетям стандарта GPRS, использующим протокол IP. Единственный новый блок оборудования, требуемый для модернизации Motorola GSM BSS – это Управляющий блок пакетной коммутации (Packet Controller Unit, PCU) подключаемый к BSC и обеспечивающий услуги пакетной радиопередачи на область охвата BSC. Для соединения узлов SGSN и GGSN с элементами систем GSM и прочих систем, обеспечивающих глобальную пакетную передачу данных.

Преимущества GPRS:

- постоянное виртуальное соединение;
- ширина частотной полосы определяется потребностями;
- тарификация за единицу переданной информации.

Внедрение передвижных отделений связи позволяет повысить доступность и качество предоставляемых услуг, расширить их перечень, повысить рентабельность почтового оператора. По объему работы одно такое отделение способно заменить порядка пяти стационарных почтовых отделений в небольших и отдаленных поселениях.

**А.И. Хобня (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **О.М. Демиденко**, д.т.н., профессор

## **УНИФИКАЦИЯ ПРОЦЕССА РАЗРАБОТКИ ИМИТАЦИОННОЙ МОДЕЛИ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА UML**

Специфика современных инструментальных средств имитационного моделирования требует от разработчика осуществлять автоматизированное преобразование концептуальной модели в формальную модель. Данная задача может быть выполнена при условии использования некоторой стандартизированной графической нотации, которая с одной стороны описывает структуру имитационной модели, с другой является понятной пользователю (заказчику). В качестве основы для нотации может быть использован язык UML.

Т. к. язык UML является стандартом, широко распространен, используется в бизнес-анализе и сравнительно прост для чтения, он позволит создать концептуальную модель в понятной для пользователя форме. А т. к. UML является языком моделирования общего назначения и содержит в себе множество различных типов диаграмм, на его основе

возможно создать нотацию описывающую структуру имитационной модели.

Описание имитационной модели должно включать в себя статическую иерархическую структуру модели (декомпозицию), потоковую граф-схему модели (схему движения потоков транзактов между устройствами имитационной модели) и схемы механизмов обслуживания.

Для описания статической структуры модели может использоваться диаграмма компонентов языка UML (component diagram). Она представляет собой статическую структурную диаграмму показывающую разбиение системы на структурные компоненты и связи (зависимости) между компонентами. Таким образом, т.к. данный тип диаграмм предоставляет графическую нотацию для описания декомпозиции какой-либо системы, он может быть использован в том числе и для описания иерархической структуры имитационной модели.

Для описания граф-схемы модели может использоваться диаграмма коммуникации языка UML (communication diagram). Диаграмма коммуникации моделирует взаимодействия между объектами или частями в терминах упорядоченных сообщений. Устройства имитационной модели могут быть представлены объектами, а транзакты – сообщениями. Сообщения на диаграмме коммуникации упорядочены. Для описания граф-схемы порядок сообщений может быть указан в качестве дополнительной, приблизительной информации, т.к. граф-схема не описывает порядок сообщений, который, ко всему прочему, может динамически изменяться в ходе работы модели. Пример диаграммы представлен на рисунке 1.

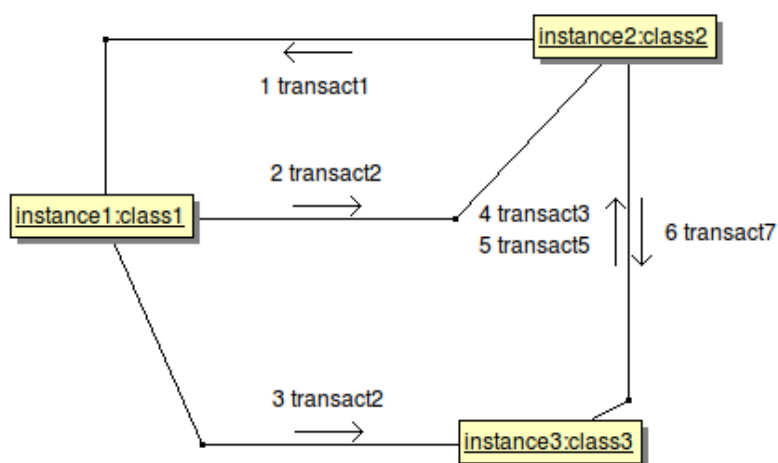


Рисунок 1 – Пример граф-схемы, описанной при помощи нотации UML

Для описания механизмов обслуживания может использоваться диаграмма активностей языка UML (activity diagram). Механизм обслуживания

транзакта на устройстве представляет собой граф переходов между активностями. Отдельные активности могут быть представлены действиями (action) на диаграмме. Переходы обозначены потоками (flow) между действиями. Также на диаграмме изображается точка входа транзакта в механизм обслуживания и точка выхода. Пример представлен на рисунке 2.

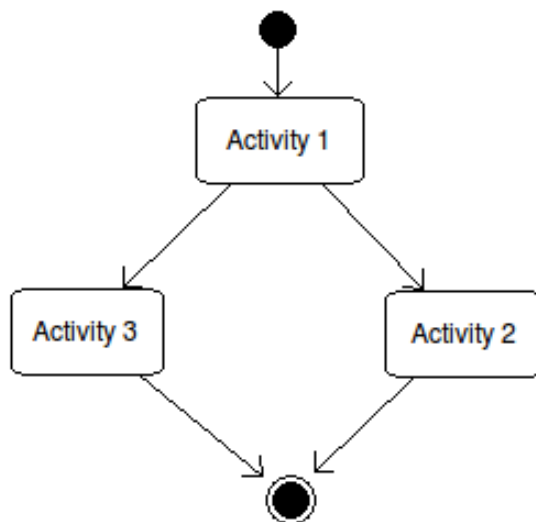


Рисунок 2 – Пример схемы механизма обслуживания

Таким образом язык UML может использоваться в качестве инструмента описания структуры имитационной модели, что позволит унифицировать процесс разработки моделей. Описание на языке UML может выступать концептуальной моделью, понятной пользователю (заказчику), и в то же время является формальной моделью, на основе которой возможно сгенерировать программный код. Например, на основе UML-диаграмм можно генерировать код на языке моделирования или на языке программирования общего назначения. Таким образом осуществляется автоматизация процесса преобразования концептуальной модели в формальную модель.

**А.И. Хобня (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **О.М. Демиденко**, д.т.н., профессор

## **ПРОЕКТИРОВАНИЕ СТРУКТУРЫ ИМИТАЦИОННОЙ МОДЕЛИ СРЕДСТВАМИ ЯЗЫКА UML**

Современные инструментальные средства имитационного моделирования (GPSS World, AnyLogic и т.д), занимаются автоматизацией этапов проектирования, кодирования и отладки имитационной модели.

Однако, большинство существующих средств имитационного моделирования, во-первых, требуют разработки формального описания модели по определенным правилам, которые требуют определенной подготовки от пользователя, во-вторых, не затрагивают этап создания концептуальной модели, отражающей, в первую очередь, взгляд на объект моделирования со стороны коллектива пользователя или аналитика, компетентного в предметной области, касающейся моделируемого объекта.

Специфика современных инструментальных средств имитационного моделирования требует от разработчика осуществлять преобразование концептуальной модели в формальную модель. Этот этап может быть автоматизирован при условии использования некоторой стандартизированной графической нотации, которая с одной стороны описывает структуру имитационной модели, с другой является понятной пользователю (аналитику, заказчику). В основу универсальной графической нотации может быть положен язык UML. С одной стороны язык UML является стандартом, широко распространен, используется в бизнес-анализе и сравнительно прост для чтения, поэтому его использование позволит создать концептуальную модель в понятной для пользователя форме. С другой стороны UML является языком моделирования общего назначения и содержит в себе множество различных типов диаграмм, следовательно на его основе может быть создана нотация, описывающая структуру имитационной модели.

Описание имитационной модели должно включать в себя статическую иерархическую структуру модели (декомпозицию), потоковую граф-схему модели (схему движения потоков транзактов между устройствами имитационной модели) и схемы механизмов обслуживания.

Для описания статической структуры модели может использоваться диаграмма компонентов языка UML (component diagram), для описания граф-схемы модели – диаграмма коммуникации (communication diagram), для описания механизмов обслуживания – диаграмма активностей (activity diagram).

**Ю.В. Хомицкий (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

## **АНТИВИРУСНАЯ ЗАЩИТА МОБИЛЬНЫХ УСТРОЙСТВ**

Мобильные телефоны стремительно ворвались в нашу жизнь. Не менее стремительно развиваются их функциональные возможности, такие как электронные игры, Интернет, органайзеры и многое другое.

Программное обеспечение мобильных телефонов становится все сложнее, и как следствие более уязвимое для вирусов. Следовательно, надежная антивирусная защита современного мобильного телефона становится все более актуальной, как с точки зрения нормальной работоспособности смартфона, так и сохранения информации его владельца. Вирусы могут стать причиной неправильного выставления счетов, нежеланного раскрытия, изменения или кражи данных, хранимых в мобильном телефоне. Кроме того, ваш телефон может стать распространителем вирусов, и вы, даже не подозревая об этом, можете заразить телефоны ваших близких, друзей и знакомых через SMS, MMS, Bluetooth, IrDA и другие средства коммуникации. Общеизвестно, что легче и дешевле предотвратить заражение вирусами, чем избавиться от них и устранить все разрушительные последствия.

Самый простой и эффективный способ обеспечить необходимую защиту мобильного телефона или КПК — это установить современную антивирусную программу, специально разработанную для защиты мобильных устройств от всех вирусных угроз. Рассмотрим несколько решений от наиболее известных разработчиков.

*Kaspersky Mobile Security Lite.* Для начала программа предлагает установку секретного кода, чтобы никто кроме нас не мог изменить настройки. Затем вводим e-mail, чтобы восстановить пароль при необходимости (рисунок 1).

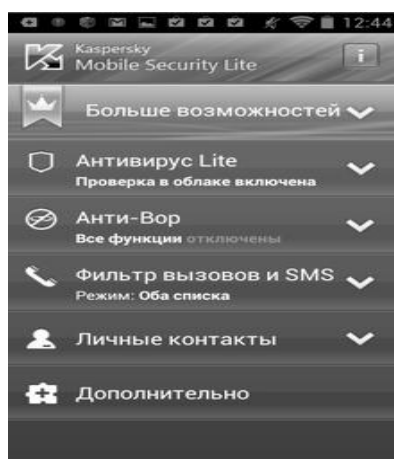


Рисунок 1 – Kaspersky

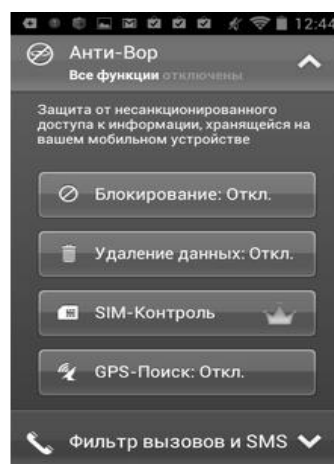


Рисунок 2 – Анти-Вор

В главном меню видим список всех основных компонентов, пройдемся по ним. Антивирус Lite проверяет программы до первого их запуска в «облачном» сервисе Kaspersky Security Network – то есть, присутствие неких вирусных баз на телефоне не требуется. Отмечу, что проверка не обязательна, делать ее, или нет, пользователь каждый раз

решает самостоятельно. Анти-Вор, как можно понять из названия, связан с ситуациями утери смартфона (рисунок 2). Он а) удаленно блокирует телефон SMS-кой б) удаленно стирает все файлы в) помогает найти телефон по координатам GPS. «Фильтрация звонков и SMS» говорит сама за себя – просто настраиваем списки блокирования, в любом случае ведется журнал событий.

По ходу навигации постоянно видны пункты с золотой короной, они работают при покупке полной версии. Стоит признать, «реклама» получилась неназойливой и уместной. Главное меню организовано весьма удобно – при касании пункта он «раскрывается», а не переходит в окно.

*Антивирус Dr. Web Light.* Доктор Веб не стал мучить нас паролями, зато сразу же в меню намекнул на необходимость обновления вирусных баз. После клика буквально за долю секунды статус сменился на «зеленый» — чудеса, да и только. Еще одним «желтым» пунктом (данный цвет свидетельствует о наличии проблемы) стоял SpiDer Guard, значит, мониторинг системы нужно включать вручную. Странное решение, вроде бы сама по себе установка программы подразумевает, что именно для этого она и скачивалась (рисунок 3). К минусу интуитивности интерфейса отнесем то, что для вызова настроек необходимо нажать на телефоне сенсорную кнопку контекстного меню. Итак, сканирование производится в постоянном режиме для новых файлов. Ручной запуск предлагает быструю проверку (установленных программ), полную (все файлы) и выборочную с пометкой нужных папок. История действий антивируса сохраняется в пункте «Статистика» (рисунок 4).



Рисунок 3 – Dr. Web Light

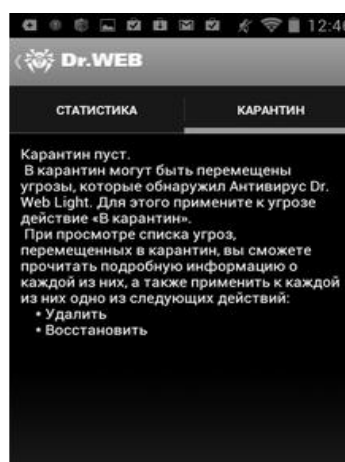


Рисунок 4 – Статистика

Если антивирус обнаруживает угрозу, сначала она помещается в карантин. Отсюда можно узнать более подробную информацию о модуле и удалить его или восстановить. После установки программы из Google Play антивирус выводит сообщение в верхней панели системы.

*AVG Бесплатный антивирус.* Лаборатория Касперского каждый месяц находит по 3 тысячи новых угроз для Android. Но почему-то только AVG после запуска предложил сканирование уже установленных программ.

Основной экран, собственно, представляет собой картинку, по касанию которой запускается сканер. Чтобы добраться до меню, нужно нажать кнопку контекстных опций. Зачем было так неудобно организовывать интерфейс – решительно непонятно (рисунки 5, 6).



Рисунок 5 – Интерфейс

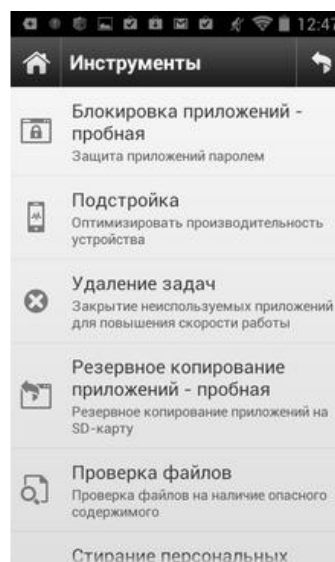


Рисунок 6 – Инструменты

В деле безопасности Android рекомендую доверять крупным опытным разработчикам, которые обладают необходимыми ресурсами для глобального мониторинга ситуации и поддержки своих продуктов (таблица 1).

Таблица 1 – Антивирусные продукты

	Kaspersky Security Lite	Dr. Web Light	Avast! Mobile Security	Eset Mobile Security
Symbian	9.1, 9.2, 9.3	S60, 9 и выше	–	S60 3rd Edition Feature Pack 1 или 2, S60 5th Edition
Windows Mobile	5.0 - 6.5	2003, 2003 SE, 5.0, 6.0, 6.1, 6.5	–	5.0, 6.0, 6.1, 6.5
Android	2.2-2.x, 4.0	1.5, 1.6, 2.0-2.3, 3.0-3.2	2.x, 3.x, 4.x	2.0-2.3
IOS	4, 5	–	–	–
BlackBerry	4.5-6.0, 7.0	–	–	–



**А.Н. Хоруженко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **С.Ф. Маслович**, к.т.н., доцент

## **ИМИТАЦИОННАЯ МОДЕЛЬ РЕКЛАМНОГО ТРАФИКА**

Модель рекламного сетевого трафика представляет собой совокупность потенциальных клиентов, рекламодателей, производителя и связей между ними. Модель реализована в среде AnyLogic [1], где были созданы собственные активные объекты, со своими параметрами и методами, имитирующие поведение реальных объектов.

Главной задачей данной модели является, учет всех потенциальных клиентов и рекламодателей, осуществление связи между ними, таким образом, что бы максимизировать прибыль производителя. В качестве производителя выступает разработчик или владелец некоторого мобильного приложения или сайта на котором в последствии будет размещена реклама. Реклама тщательно подбирается под аудиторию ресурса по некоторым критериям, например по его тематике, местоположению и др. В качестве рекламодателей выступают рекламные компании. Они ориентированы на привлечение потенциальных клиентов посредством рекламирования своего продукта. Рекламодатели платят производителю за показ их рекламы, переход по их ссылкам и т. д. В качестве потенциальных клиентов выступают группы людей, различного возраста, социального статуса и места проживания. Они являются пользователями некоторых интернет ресурсов или мобильных приложений. Для реализации данной модели был выбран метод агентного моделирования. Данный метод моделирования максимально подходит для описания поведения потенциальных клиентов. В среде AnyLogic была сделана детальная презентация исследования, с графиками зависимостей, диаграммами и анимацией, позволяющая наглядно оценить актуальность системы. Созданную модель можно симулировать в реальном времени или задать любую единицу модельного времени. В ходе симуляции модели можно изменять скорость времени. Таким образом, можно оценить поведение потенциальных покупателей, за любой промежуток времени и на основе сделанных выводов поменять некоторые концепции и повысить доход.

### **Литература**

1. Карпов, Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 / Ю.Г. Карпов. – СПб.: БХВ-Петербург, 2006. – 400 с.

**Д.Б. Цырибко (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Кучеров А.И.**, старший преподаватель

## **АВТОМАТИЗАЦИЯ РАЗРАБОТКИ WEB-САЙТОВ**

Современный сайт – это уже не просто набор страниц со ссылками между ними (как это было лет десять назад), а полноценный программный продукт, который отслеживает действия пользователей, позволяет им между собой общаться и предлагает множество полезных сервисов в зависимости от поставленных владельцем сайта задач. В разработку подобной системы вкладываются внушительные суммы денег. Зачастую бюджет большого корпоративного сайта сравним с бюджетом небольшой фирмы. Задачи администрирования, управления и функциональности решают так называемые «Системы Управления Контентом» (ContentManagementSystem, CMS). Уже написано предостаточно как платных, так и бесплатных систем. В данной статье рассмотрим наиболее популярные из них:

- CMS Drupal.
- Joomla.
- WordPress.

Drupal – это не только система управления контентом в классическом понимании этого термина. Это больше ядро, на основе которого можно собрать практически неограниченный функционал сайта. Нужен новостной портал? Без проблем! Собираем необходимые модули и получаем портал с новостями. Нужен каталог файлов? Поступаем по тому же принципу и получаем полнофункциональный каталог. Drupal считается одной из самых надежных систем на сегодняшний день.

Joomla полностью бесплатна, обладает множеством модулей расширения функционала (они имеют разные названия «модули», «компоненты», «мамботы» и направлены на решения различных задач). С помощью этих модулей можно добавить интернет-магазин на сайт компании буквально за несколько минут! При этом Вы получаете надежный магазин, который будет стабильно работать. Удобство Joomla в том, что в стандартном дистрибутиве практически нет нестандартных модулей и компонент. Всё внимание разработчиков переносится на удобство и безопасность. Очень сложно придумать задачу, с которой не под силам справится Джумле!

WordPress Бесплатная CMS для ведения блога, считающаяся лучшей в мире. Функциональные возможности не уступают, а зачастую и превосходят знаменитые Живые Журналы. С вордпрессом Вы можете с легкостью публиковать видеофайлы с различных ресурсов, создать блог

фотографа с демонстрацией работ, блог музыканта с онлайн аудио-примерами. А можете просто записывать в своеобразный онлайн-дневник свои мысли и чувства. Кстати, многие известные люди в последнее время предпочитают вести блоги. Считается, что это способствует их популярности. Функциональность WordPress, как и во всех остальных CMS расширяется установкой дополнительных плагинов. Это делать крайне просто, как и работать с системой вордпресс. Отдельно стоит упомянуть знаменитую установку вордпресса «в три шага». Блоги – это не единственное, для чего используется WordPress. На нем можно построить порталы и фотогалереи, каталоги статей и витрины товаров.

**А.С. Чеботаревский (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **В.С. Давыдов**, к.т.н., доцент

## **АЛГОРИТМ ПОИСКА ПУТИ JUMP POINT SEARCH**

Данный алгоритм является улучшенной версией алгоритма поиска пути  $A^*$ . Основная идея данного алгоритма заключается в том, что он ускоряет поиск пути, «перепрыгивая» те места, которые должны быть просмотрены. Алгоритм JPS впервые был представлен в 2011 году Daniel Harabor.

JPS работает на ориентированном графе единой стоимости. Каждая вершина должна иметь восемь соседей (соседних вершин). Каждый шаг по вертикали и горизонтали имеет стоимость 1 (по-другому можно сказать, что эти ребра имеют вес 1), шаг по диагонали имеет стоимость  $\sqrt{2}$ . Препятствия – это те вершины, стоимость перехода к которым бесконечна, или по-другому: движение через препятствия запрещено.

Главной отличительной особенностью JPS от  $A^*$  является наличие «прыжковых точек». Эти точки позволяют ускорить алгоритм поиска пути. Они описываются двумя правилами выбора соседей: одно для прямолинейного и второе – для диагонального движения. В обоих случаях необходимо доказать, что, исключая из набора ближайших соседей вокруг точки, найдётся оптимальный путь из предка текущей точки до каждого из соседей, и этот путь не будет содержать в себе посещенную точку.

Прыжковые точки интересны тем, что они имеют соседей, которые не могут быть достигнуты альтернативным путём: оптимальный путь должен идти через текущую точку. Таким образом, стоимость перемещения  $g(y) = g(x) + \text{dist}(x; y)$ .

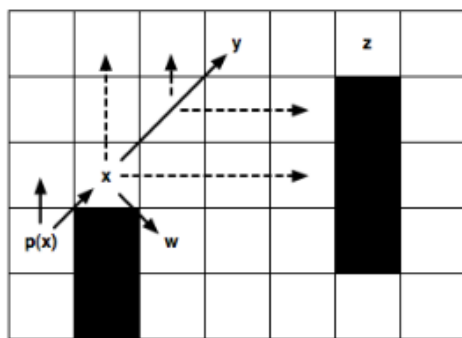


Рисунок 1 – Пример «прыжковой точки»

На рисунке 1 показан пример точки прыжка. Движение начинается в точке  $x$  и продолжается по диагонали, пока не встречается на пути точка  $y$ . Из точки  $y$  в точку  $z$  можно попасть  $k_i$  количеством шагов по горизонтали. Таким образом,  $z$  является преемником точки для прыжка  $x$ , а это, в свою очередь, определяет  $y$  как преемник для прыжка из точки  $x$ .

Для сравнения скорости работы JPS с другими алгоритмами мы использовали сервис, реализованный на js который можно найти по адресу [qiao.github.com/PathFinding.js/visual/](http://qiao.github.com/PathFinding.js/visual/). Благодаря нему можно узнать скорость работы всех основных алгоритмов на ориентированном графе единой стоимости. В качестве алгоритмов для сравнения были выбраны  $A^*$  (так как он является базовым по отношению к JPS) и алгоритм Дейстры. Вот какие результаты были получены. На рисунке 2 один мы видим реализацию  $A^*$ .

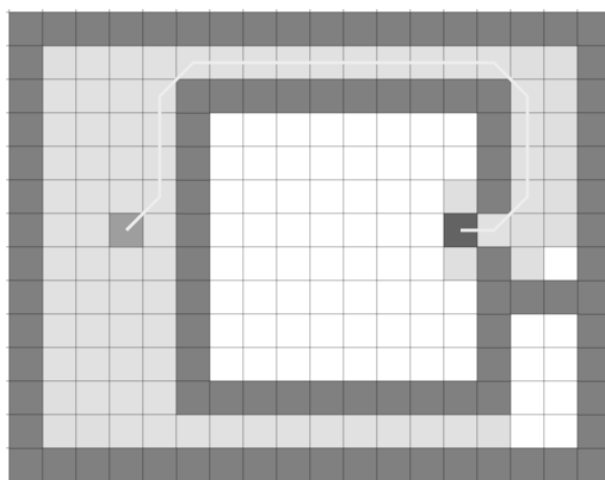


Рисунок 2 – Реализация алгоритма  $A^*$

Количество операций необходимых для нахождения пути составило 154, а время необходимое для поиска пути равно 4 миллисекундам. На рисунке два показан алгоритм Дейстры, количество операций которого составило 185, а время оказалось равным 5 миллисекундам.

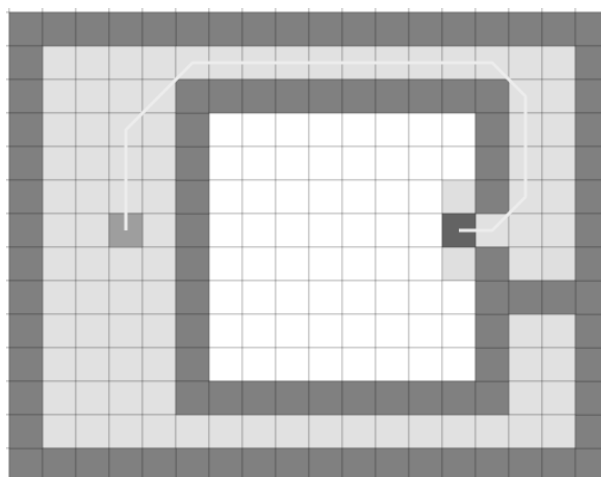


Рисунок 3 – Реализация алгоритма Дейстры

На рисунке 3 реализован поиск методом JPS. Количество операций составило 22, а время выполнения было равным 2 миллисекундам.

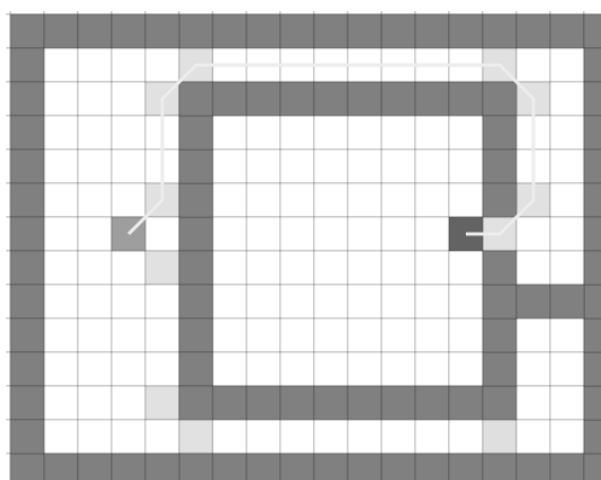


Рисунок 4 – Реализация алгоритма JPS

Выводом из нашего исследования является то, что алгоритм JPS на данный момент является самым быстрым алгоритмом ориентированным на поиск пути во взвешенном графе единой стоимости. Скорость поиска пути JPS оказалась в семь раз быстрее, чем алгоритмом  $A^*$ , и в девять раз быстрее, чем алгоритмом Дейстры.

**А.И. Чернышев (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

## **МОДДИНГ ПРИНТЕРОВ**

На сегодняшний день, принтер является самым популярным периферийным устройством, подключаемым к персональному компьютеру.

Пользователей компьютеров при покупке принтеров, как правило, волнует уже не только вопрос, какую именно модель приобрести, но и не менее важные технические особенности и проблемы, связанные, например, с постоянным наличием расходных материалов у фирмы-продавца, дальнейшим сервисным обслуживанием печатающих устройств, надежность, быстродействие.

В последнее время все известные бренды перешли на чипованные расходные материалы. Это предоставляет пользователям ряд удобств при работе с техникой. Именно благодаря электронному интеллекту принтер или многофункциональное устройство вовремя сообщает о необходимости заменить картриджи. Чип следит за ресурсом принтера.

Чип – это небольшая микросхема. На ней «прошита» информация о расходном материале, «язык» общения с необходимым устройством и ресурс, на который рассчитан картридж. На нём же содержится техническая информация типа серийного номера самого электронного компонента и более специфические данные.

Чипы делятся на: контактные и бесконтактные, их легко отличить визуально.

Большинство производителей стали использовать чипы в своих устройствах, которые не позволяли использовать один и тот же картридж несколько раз, просто заправив его. Это привело к тому что пользователи стали перепрошивать устройства для их дальнейшего использования.

Для прошивки принтеров используются самопайные программаторы, с питанием от USB (рисунок 1).

Термин «программирование микросхем» обозначает процесс записи (занесения) информации в постоянное запоминающее устройство (ПЗУ) микросхемы. Как правило, запись информации (программирование), производится при помощи специальных устройств – программаторов.

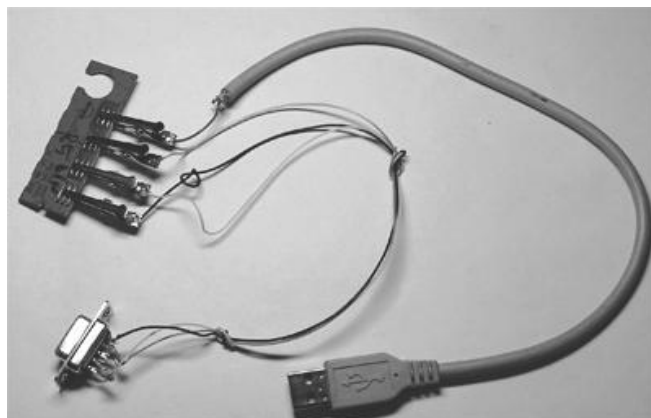


Рисунок 1 – Программатор

Для программирования используется программа **PonyProg** (рисунок 2).

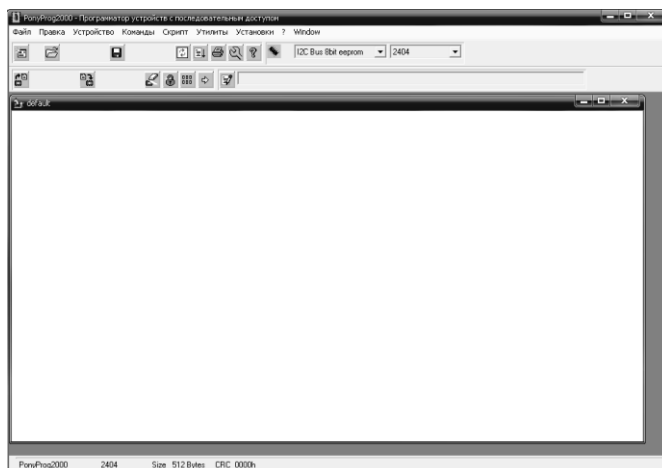


Рисунок 2 – PonyProg

Эта программа позволяет считать данные, с подключенного программатором, чипа, внести свои изменения либо открыть уже готовую прошивку, и записать эти данные обратно на чип.

Для начала следует настроить программу для работы: произвести калибровку и выбрать порт подключения. В процессе калибровки программа настраивает свои процедуры формирования временных интервалов под конкретный компьютер.

Цель калибровки – повышение точности формирования интервалов времени. При выполнении этой процедуры компьютер не должен выполнять никаких других программ. Затем считывается информация с подключенного чипа, и на него можно записать исправленные данные. После загрузки информации (программы или данных) в окне появляется дамп памяти. Дамп – это широко распространенный способ представления цифровых данных. Он представляет собой таблицу шестнадцатеричных чисел, записанных рядами по 16 чисел в ряду.

Пакетный режим очень удобен в процессе отладки программы. Если в пакет включена опция «**Reload Files**», то достаточно один раз вручную открыть нужный файл нужного проекта, а затем можно просто нажимать кнопку «Программирование» каждый раз, когда нужно перепрошить микросхему новой версией программы. Все остальное PonyProg сделает за вас.

Вносить свои изменения можно, если разобраться в значениях содержимое ячейки памяти (рисунок 3).





– необходимо тестировать быстро, соблюдая жесткие сроки поставки программных продуктов;

– необходимо обеспечивать достаточно высокое качество тестирования, которое бы гарантировало, что дефекты, приводящие к разрушительным последствиям, не просочатся на компьютеры конечных пользователей.

Проблема связана с тем, чтобы удовлетворить каждое требование без ущерба для другого. Невозможно гарантировать отсутствие ошибок в нетривиальной программе, но можно автоматизировать основную часть ежедневной однотипной работы, что позволит снизить часть нагрузки с тестировщиков и позволит использовать выигранное время для более интеллектуального поиска дефектов в других элементах игровых приложений.

Задача тестировщика в таком случае сводится к поиску элементов приложения, тестирование которых возможно автоматизировать, проанализировать и выбрать средство автоматизации, и написать скрипт, который будет вести проверку приложения и выдавать в нужной форме отчет.

**А.В. Шамело (УО «ГГУ им. Ф.Скорины», Гомель)**

Науч. рук. **С.В. Карпенко**, к.э.н., доцент

## **РАЗВИТИЕ В РЕСПУБЛИКЕ БЕЛАРУСЬ РЫНКА БАНКОВСКИХ ПЛАТЕЖНЫХ КАРТОЧЕК**

Банковская платежная карточка в Республике Беларусь является платежным инструментом, обеспечивающим доступ к банковскому счету и проведение безналичных платежей за товары и услуги, получение наличных денежных средств и осуществление иных операций в соответствии с законодательством Республики Беларусь. В целях обеспечения сохранности денежных средств на счетах клиентов, повышения уровня финансовой грамотности населения и предотвращения мошеннических операций с использованием банковских платежных карточек Национальным банком разработаны Рекомендации по безопасному использованию банковских платежных карточек.

В соответствии с Банковским кодексом Республики Беларусь выпуск банковских платежных карточек в обращение осуществляется банками–эмитентами. Операции с использованием банковских платежных карточек проводятся банками–эмитентами в пределах полномочий, предоставленных

имеющимися у них лицензиями Национального банка Республики Беларусь и в соответствии с Инструкцией о порядке совершения операций с банковскими платежными карточками, утвержденной постановлением Правления Национального банка Республики Беларусь от 18.01.2013 № 34.

По данным Нацбанка, на 1 января 2013 года 24 банка Республики Беларусь эмитируют банковские платежные карточки внутренних, международных и внутренних частных платежных систем: ОАО "Белагропромбанк"; ОАО "БПС-Сбербанк"; ОАО "АСБ Беларусбанк"; ОАО "Белинвестбанк"; "Приорбанк" ОАО; ОАО "Банк БелВЭБ"; ОАО "Паритетбанк"; ОАО "БНБ-Банк"; ОАО "Белгазпромбанк"; ЗАО "РРББанк"; ЗАО "МТБанк"; ОАО "Технобанк"; ОАО "Франсабанк"; ЗАО "Трастбанк"; ЗАО Банк ВТБ (Беларусь); ЗАО "Альфа-Банк"; ОАО "Банк Москва-Минск"; ЗАО "Дельта Банк"; ЗАО "ИнтерПэйБанк"; ОАО "ХКБанк"; ЗАО "БТА Банк"; ЗАО "БелСвиссБанк"; ЗАО "Альфа-Банк Финанс"; ЗАО "Идея Банк".

Эмиссию электронных денег осуществляют 6 банков, которые выпускают в обращение электронные деньги следующих систем расчетов: EasyPay, Берлио, iPay, ОСМП и другие. Банками активно развиваются системы дистанционного банковского обслуживания, преимущественно интернет-банкинг, а также ЕРИП.

Количество банковских платежных карточек, находящихся в обращении по состоянию на 1 января 2013 года составило 10,4 млн. ед., в том числе 4,9 млн. карточек платежной системы "БелКарт", 5,5 млн. – международных платежных систем.

В Республике Беларусь установлено 3 342 инфокиоска, 3701 банкомат, 80 импринтеров, 37 063 организации торговли (сервиса) оснащены 56 796 платежными терминалами.

За 2012 год на территории Республики Беларусь осуществлено 643 627 814 операций с использованием банковских платежных карточек в белорусских рублях на сумму 160 211 292 млн. рублей. Удельный вес безналичных операций в общем количестве операций с использованием платежных карточек составил 62,6 %, а в суммарном выражении – 18,9 %.

Общее количество операций в иностранной валюте за 2012 год составило 3 339 966 операций на сумму 1 506 029,3 тыс. долларов США. Удельный вес безналичных операций в общем количестве операций с использованием платежных карточек в иностранной валюте составил 16,2 %, а в суммарном выражении – 9,1 %.

Для сравнения: в России эти показатели составили 47,5 % и 14,7 %; на 1.01.2012 г. в Германии — 56 % и 29,5 %, Финляндии — 87 % и 70,5 %, Швеции — 88,5 % и 73,5 % соответственно.

В валюте за 9 месяцев совершено свыше 2,45 млн. операций на сумму более 1,127 млн. USD, из них безналичных операций — 17,2 %, в суммарном выражении — 10,1 %.

**И.Г. Шедова (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **Ю.С. Митченко**, ассистент

## **РАЗРАБОТКА ДИАЛОГОВОГО ОКНА ПРИ СОЗДАНИИ ОТЧЕТОВ В СРЕДЕ 1С: ПРЕДПРИЯТИЕ 7.7**

Создание диалогового окна отчета, как и любого другого объекта метаданных в системе 1С: Предприятие 7.7, является одним из ключевых моментов разработки этого объекта. Назначение диалога – предоставить пользователю удобную форму для ввода и просмотра информации. Можно сказать, что диалог является компьютерным аналогом бумажного документа. Как и бумажный документ, диалог позволяет быстро ввести необходимую информацию и запомнить ее для последующей обработки, а при необходимости – вновь вернуться к ранее введенным данным для просмотра или корректировки.

Диалоги форм должны содержать поля ввода и выбора. Эти поля могут иметь не только базовый тип (строка, число, дата), но и тип любого типообразующего элемента метаданных (справочника, документа, перечисления и т.д.), из числа определенных в конфигурации, являющейся текущей во время создания этого элемента диалога.

Элементы диалога на форму можно добавить из пункта меню Вставить. Все элементы диалога, кроме элемента для выбора периода, имеют заголовок и идентификатор. Заголовок виден пользователю на форме элемента, а идентификатор служит для именованя элемента в модуле отчета. Имя идентификатора подчиняется правилам именованя в 1С: Предприятие и не может начинаться с цифры, содержать в себе знаки препинания или пробелы. Текст заголовка может быть любым. При создании элемента диалога можно задать автоматическое создание относящейся к нему процедуры.

Существует 11 видов элементов, добавляемых на форму: кнопка, флажок, переключатель, список, поле со списком, рамка группы, текст, реквизит диалога, картинка, таблица, выбор периода. В модуле формы к элементам диалога можно обращаться по их идентификаторам через конструкцию Форма.ЭлементДиалога. Для работы с ними существует несколько методов. Элемент формы можно сделать доступным или

недоступным для редактирования. Чтобы не загромождать маленькую форму большим количеством элементов сразу, элементы можно делать видимыми или невидимыми. Например, в листинге 1 показано, что можно делать видимым поле при выборе другого элемента формы, в данном случае флага.

Листинг 1 – Процедура ВыбранИсточник() из модуля отчета для элемента ПолеИсточникФинансирования

```
Процедура ВыбранИсточник()
```

```
    Если ФлагИсточникФинансирования = 1 Тогда
```

```
        Форма.ПолеИсточникФинансирования.Видимость(1);
```

```
    Иначе
```

```
        Форма.ПолеИсточникФинансирования.Видимость(0);
```

```
    КонецЕсли;
```

```
КонецПроцедуры
```

Кроме того, можно менять заголовки на кнопке и на других элементах диалога. Например, сначала на кнопке написано «Далее», а после выполнения некоторых функций текст можно сменить на «Готово». Можно менять цвет элементов диалога, шрифт и положение текста в них.

В процессе создания диалога в Конфигураторе возможно определить необходимое количество слоев диалога и закладок. Слой диалога – это некоторое множество элементов диалога. При этом каждый элемент диалога принадлежит одному и только одному слою. Элементов, не принадлежащих каким-либо слоям, быть не может. На этапе исполнения можно управлять видимостью определенных слоев. Для этого применяется метод ИспользоватьСлой() объекта Форма. Слой можно скрыть, показать, показать и скрыть остальные слои при этом.

Закладки – это элемент управления диалога, который выглядит, как несколько ярлычков с надписями. Данный элемент диалога невозможно создать в Конфигураторе, он создается только при исполнении методом ИспользоватьЗакладки(). Когда закладки определены, пользователь может по ним перемещаться щелчками мыши. При щелчке мыши по закладке система пытается найти и выполнить предопределенную процедуру ПриВыбореЗакладки(). Данная процедура принимает два параметра: номер закладки и значение закладки, которое было использовано при создании закладки.

Традиционно закладки используют совместно со слоями. То есть при открытии формы определяется набор закладок в виде набора значений вида Представление закладки – Слои диалога, когда в качестве значения закладки указывается набор идентификаторов слоев через запятую.

По умолчанию на создаваемой форме уже находятся две кнопки: Сформировать и Закреть. Кнопка Сформировать инициирует выполнение процедуры Сформировать() из модуля отчета. Как правило, в текст этой процедуры помещают программный код всех действий, необходимых для выборки данных и создания отчета. Методы установки параметров в для элементов диалога обычно располагаются вне процедуры Сформировать().

Щелкнув правой кнопкой мыши на форме диалога и выбрав пункт меню Разметка, можно включить или отключить отображение горизонтальных и вертикальных пунктирных линий на форме. Эти линии служат для точного и ровного размещения элементов на форме и отображаются только в режиме Конфигуратора.

Итак, к основным действиям при создании формы диалога относятся:

- создание, копирование, изменение размеров и свойств элементов диалога;
- создание слоев;
- изменение порядка обхода элементов диалога;
- использование разметки и кнопок выравнивания;
- построение многослойных форм диалогов с использованием закладок.

Диалоги форм отчетов, как и любые другие диалоги форм, являются связывающим звеном в диалоге пользователя с системой 1С: Предприятие. Правильно созданные диалоги облегчают и ускоряют работу пользователя, тем самым повышая производительность работы с системой.

**А.С. Якубова (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

## **ИСПОЛЬЗОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ «ЭКСПРЕСС-3» БЕЛОРУССКОЙ ЖЕЛЕЗНОЙ ДОРОГОЙ**

С мая 2007 года на Белорусской железной дороге был введен новый вид услуг для населения — заказ и доставка пассажирских билетов через Интернет, а так же безналичный расчет за приобретаемые проездные документы с использованием банковских пластиковых карточек. Это стало возможным в связи с переходом на АСУ «Экспресс-3» — вычислительная сеть по обслуживанию пассажиров для автоматизации продажи билетов и резервирования мест в поездах и оперативного управления пассажирскими. В настоящее время представляет собой современный

программно-аппаратный комплекс, построенный на базе высокопроизводительных ЭВМ, с широко развитой сетью продажи билетов.

Система «Экспресс-3» представляет собой самостоятельный программный комплекс и базу данных для управления пассажирскими перевозками, тесно связанные с другими компонентами единой информационной системы железнодорожного транспорта.

Наиболее существенными технологическими отличиями системы «Экспресс-3» в части резервирования мест и продажи билетов являются реализация продажи билетов по ходу следования поезда с нумерацией мест и хранение мест и всех проездных документов, оформленных в течение последних шести месяцев, в базе данных, доступной в оперативном режиме.

**Вычислительный комплекс АСУ «Экспресс-3»** включает в себя две ЭВМ типа IBM 9672 моделей R36 и R37 с периферийным оборудованием и средствами телеобработки, в которые входят маршрутизаторы, высокоскоростные коммутаторы, системы управления сетью и др. Конфигуратор вычислительного комплекса «Экспресс-3» представлен на рисунке 1.

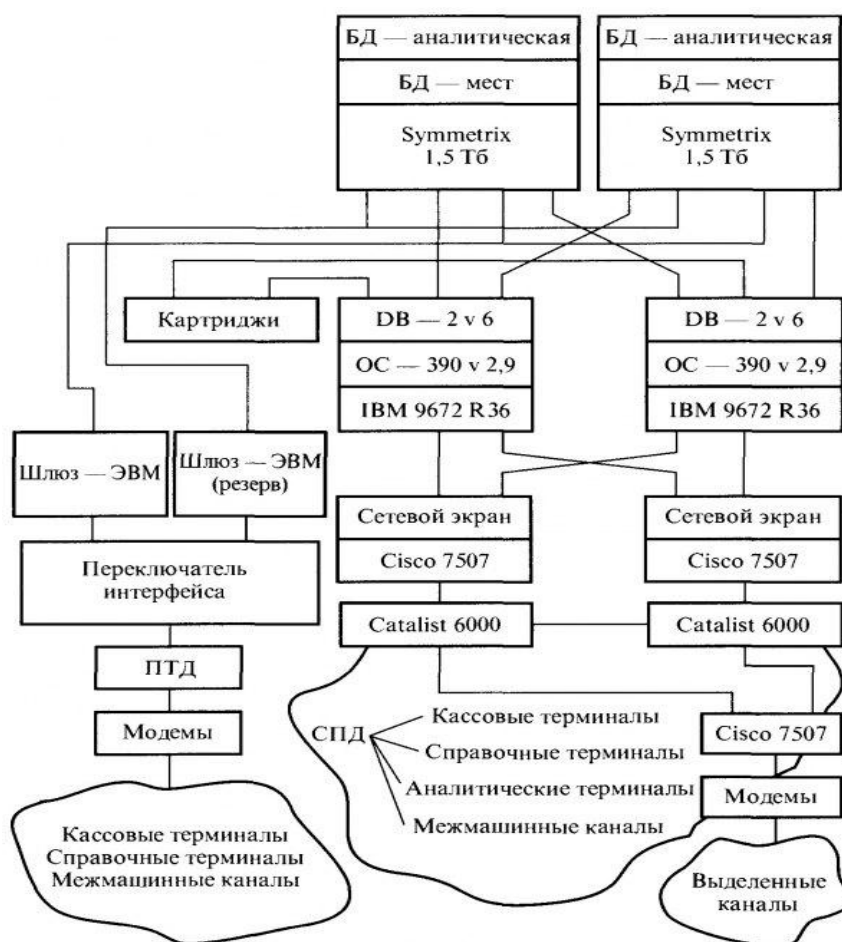


Рисунок 1 – Конфигуратор ВК «Экспресс-3»

Вычислительный комплекс имеет три возможности подключения терминальной, справочной и другой аппаратуры, включая ЭВМ других систем: по старым протоколам В8С-1 и В3С-3 с использованием оставшихся каналов от «Экспресс-2»; по новому протоколу ТСР/ІР через сеть передачи данных российских железных дорог с использованием каналов УРМ; по выделенным каналам для прямого подключения терминалов по протоколу ТСР/ІР.

Это дает возможность временно использовать старую периферийную аппаратуру системы «Экспресс-2». Для преобразования протоколов в переходный период от «Экспресс-2» к «Экспресс-3» используются шлюзы — ЭВМ «Экспресс-2». После замены на дорогах старого оборудования «Экспресс-2» и развития сети передачи данных (СПД) шлюзы могут быть исключены из конфигуратора.

Все региональные системы «Экспресс-3» на дорогах сети имеют общий распределенный банк данных мест в поездах, на базе которого они взаимодействуют и функционируют. Входной информацией АСУ «Экспресс-3» являются заказы и сообщения, поступающие от ее абонентов через кассовые терминалы билетных кассиров, АРМ работников железных дорог, справочные установки и серверы сети Интернет. Абонентами-пользователями являются кассиры билетных и багажных касс, работники служб дорог и линейных предприятий, а также пассажиры, обращающиеся в систему через справочные устройства и сеть Интернет.

Программное обеспечение системы «Экспресс-3» построено на системно-технической платформе фирмы ІВМ с использованием операционной системы OS390, СУБД DB2, системных компонент ТСР/ІР и MQS.

Комплекс аналитической базы данных системы «Экспресс-3» собирает и хранит уникальную информацию о пассажирских перевозках, которой ранее никогда не было. База данных содержит более 200 видов объектов и занимает около 300 Гигабайт памяти.

Большое количество задач управления и пользователей этих задач создает высокую нагрузку на систему. Поэтому эксплуатация базы переведена на новую мощную ЭВМ ІВМ Z990, развитие технической базы необходимо предусматривать и в дальнейшем. Высокие требования по надежности и безопасности данных, использование этих данных не только для анализа, но и непосредственно для обслуживания пассажиров, высокая загрузка системы, значительная изменчивость состава и содержания объектов базы в связи с внедрением новых задач и технологий исключают возможность предоставления непосредственного доступа к базе внешним пользователям.

В Беларуси для продажи билетов по Интернету компания ИВА-Гомель разработала Систему продажи проездных документов в международном и внутривнутриреспубликанском сообщении физическим лицам по банковским пластиковым карточкам через сеть Интернет на Белорусской железной дороге.

Система позволяет получать on-line справку из АСУ «Экспресс-3» о расписании движения, наличии свободных мест, стоимости проезда и маршруте следования поездов. Реализован функционал заказа билетов и их оплаты через систему Интернет-банкинг.

Основой Системы является промышленная системно-техническая платформа Blade-сервера IBM серии JS23. Система работает под управлением операционной системы AIX, прикладное программное обеспечение функционирует в среде IBM WebSphere Application Server, в качестве базы данных используется DB2.

Дальнейшее развитие Системы предполагает оплату железнодорожных билетов через платежно-справочные терминалы АСБ "Беларусбанк" и Мобильный банкинг.

**С.С. Якубова (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **А.В. Воруев**, к.т.н., доцент

## **ПРИМЕНЕНИЕ АВТОМАТИЗИРОВАННЫХ СИСТЕМ ТРАНСЛЯЦИИ СПОРТИВНЫХ СОБЫТИЙ В РЕЖИМЕ LIVE**

Каждый день во всём мире проходит огромное количество спортивных событий. Телевизионные трансляции охватывают лишь небольшое их количество, но благодаря высокому уровню развития современных технологий любители спорта имеют возможность быть в курсе всех популярных мировых событий и следить за ними онлайн. Ставить ставки, как до начала поединка, так и во время него, на всех аренах и стадионах планеты, включая и такие экзотические страны, как Гана, Гватемала, Уэльс, Кипр, Коста-Рика и т. д.

Существует несколько компаний, которые поставляют информацию непосредственно со стадионов в режиме LIVE по всему миру. Так как предоставляемые ими данные используются букмекерскими конторами и различными медиа провайдерами, поэтому оперативность является одним из самых главных приоритетов в работе. Данные компании применяют разные технологии для трансляций спортивных событий и имеют разные уровни автоматизации передачи данных. Их работа основана



на наличии сети специально обученных скаутов или live-репортёров, которые в режиме 24/7/365 сообщают статистику событий.

Часть компаний использует следующий принцип: скаут, находясь на стадионе, принимает звонок на мобильный телефон из call-центра компании и далее на протяжении всего поединка остаётся на связи, сообщая оператору о происходящих событиях. Далее эти данные обрабатываются квалифицированными сотрудниками колл-центра с помощью специализированного программного обеспечения и системы одного клика.

Например, компания Real Time Sportscast использует «AJAX Push», также известный как потоковое Push – подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, что позволяет приложениям работать быстрее. Данные доступны менее чем за две секунды после того, как событие произошло. Это позволяет добиться того, что связь между репортёром на стадионе и клиентом компании осуществляется практически в режиме реального времени, тем самым, улучшая качество и скорость передачи данных.

Каждым модуль ПО реализован на JAVA и разработан как автономное приложение. Что позволяет улучшить масштабируемость деятельности, добавлять новые виды спорта и т. д. в будущем, что в свою очередь даёт больше возможностей для расширения и совершенствования продуктов. В дополнение к этому, система RTS хорошо функционирует с несколькими компьютерными системами и ядрами, что делает его удобным для крупных предприятий и более мелких компаний.

Примером компании с более усовершенствованной технологией является Sportsdata AG. Sportsdata AG – один из ведущих в мире поставщиков спортивной статистики в реальном времени.

Одним из вариантов передачи информации, который применяет выше названная компания, так же является использование мобильной связи. Но в данном случае скаут соединяется не с оператором, а непосредственно с системой при помощи широко распространенной в call-центрах технологии IVR. Interactive Voice Response – технология, позволяющая человеку взаимодействовать с компьютерной системой, передавая ей сигналы с помощью клавиатуры или голоса. Для ведения трансляции скауту необходимо нажимать различные комбинации на клавиатуре мобильного телефона, обозначающие те или иные события в данном виде спорта. Сама IVR платформа реализована на языке C++, по стандарту XML и поддерживает языки доступа SQL, HTTP.

Прежде чем приступить к освещению спортивных событий со стадионов, каждый скаут проходит процедуру тренировок, используя специально разработанный на Java эмулятор мобильного телефона.

Sportsdata AG имеет приложение на Java для Android устройств с поддержкой Java's Simple API for XML, которое позволяет освещать баскетбольные матчи. Интерфейс приложения представлен рисунком 1.



Рисунок 1 – Интерфейс приложения для Android

Ведения трансляций с пяти топовых футбольных чемпионатов, таких как Английская Премьер-лига, Испанская Ла Лига, и др., осуществляется с помощью ноутбуков через уникальный Soccer Scout Admin продукт. Однако технология с использованием мобильной связи сейчас остаётся наиболее используемой, так как является более доступной во всём мире.

Все данные, полученные от скаутов, обрабатываются и заносятся в Scout Client, где информация представлена в удобном виде. Интерфейс этого приложения представлен на рисунке 2 (в данном случае отображена статистика матча Первой лиги Чемпионата Беларуси по футболу между ФК «ДСК» Гомель и ФК «Руденск»). Клиенты могут импортировать, обрабатывать и анализировать XML данные, написанные на NET (C #) / MySQL.

Для автоматизации управления работой скаутов разработана онлайн Scout Management Platform на PHP/MySQL.

Компании-провайдеры трансляций спортивных продолжают совершенствовать свои технологии, постоянно расширяют свои сети скаутов, добавляют трансляции новых видов спорта, автоматизируют процесс своей работы. Они не оставляют без внимания и вторые, и третьи лиги,

дивизионы в том или ином виде спорта. Одним словом, дают клиентам возможность чувствовать себя так, как будто они находятся не перед компьютером, а на стадионе, даже если трансляция интересующего их поединка не показывается по TV.

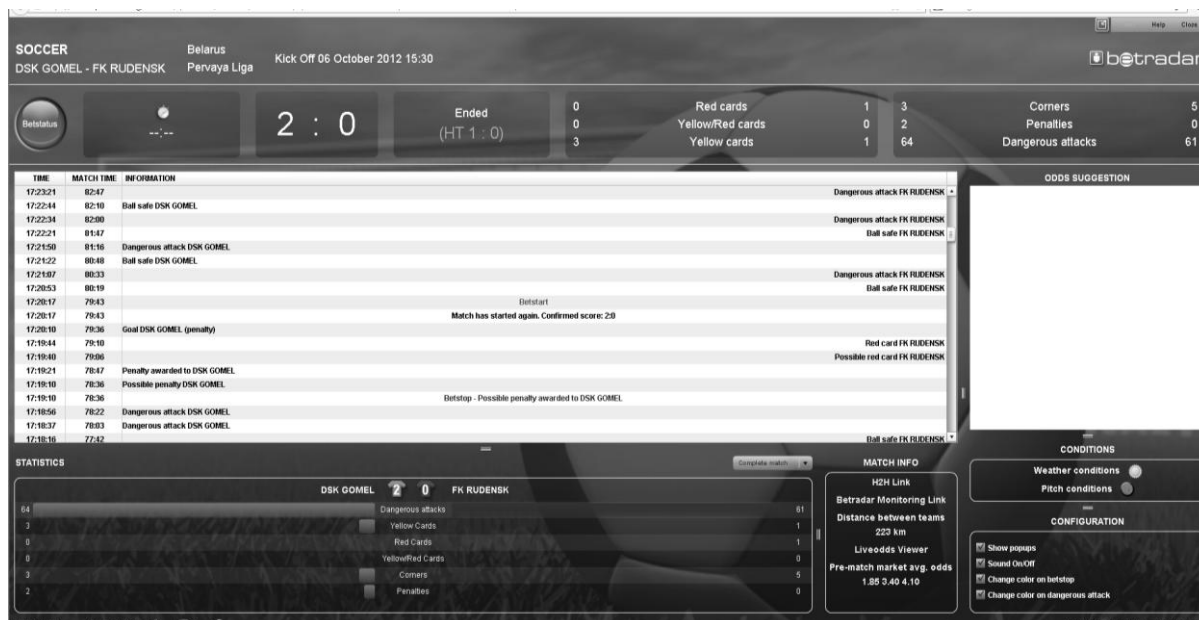


Рисунок 2 – Интерфейс Scout Client

**В.Н. Ярчук (УО «ГГУ им. Ф. Скорины», Гомель)**  
Науч. рук. **В.Н. Леванцов**, старший преподаватель

## **ЗАДАЧИ ВВОДА, ХРАНЕНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ НА ПРЕДПРИЯТИИ**

В настоящее время каждое предприятие сталкивается с постоянно растущим объемом хранимой и обрабатываемой информации. В связи с этим возрастает необходимость внедрения автоматизированных систем в работу предприятия. Актуальность таких систем возрастает с каждым годом. Одним из способов ведения бухгалтерского учета или иной деятельности предприятий является внедрение баз данных. Систему баз данных можно рассматривать как компьютеризированную систему хранения записей.

В представленной работе рассматривается разработка информационного и программного обеспечения для ввода, автоматизации хранения и обработки информации при организации работы предприятия машиностроительного комплекса.

Простой и удобный способ ввода документов можно осуществлять, используя специально разработанное для этих целей программное обеспечение.

Прикладные программы (приложения, application) предназначены для того, чтобы обеспечить применение вычислительной техники в различных сферах деятельности человека.

Разрабатываемое прикладное программное обеспечение должно решить основные вопросы, связанные со сбором и первичной обработкой данных. Интерфейс должен быть ориентирован на пользователя, не имеющего специальной подготовки.

Разработанное приложение «Подготовка данных для расчета заработной платы» работников РУП «Гомсельмаш» будет добавлено в уже существующий АРМ App03 «Персонал».

В приложении решаются задачи автоматизации деятельности как ответственных лиц структурных подразделений РУП «Гомсельмаш», принимающих решения по зарплате персонала, так и бухгалтеров-расчетчиков заработной платы. Подсистема используется совместно с подсистемой «Зарплата». В подсистеме используются данные личных карточек и табельного учета.

Предприятие РУП «Гомсельмаш» имеет сложную структуру, большое количество цехов, отделов и управлений, поэтому приложение рассчитано на интенсивную многопользовательскую работу, и помимо собственно быстродействия, оно должно обеспечивать также высокую параллельность работы пользователей. А значит, при его проектировании была внимательно проанализирована создаваемая структура на предмет возникновения узких мест при одновременной работе большого количества пользователей. Кроме этого был проведен системный анализ – уровень проблем, тип и взаимосвязь информации; определены основные задачи и конечные цели.

Основные задачи подсистемы: ввод любых документов для начислений и удержаний, настроенных на специфику предприятия, расчет и работа с документами. Разрешается обработка произвольного количества начислений и удержаний, данные сохраняются за любой период времени. Допускается обработка данных за прошлые месяцы. Обеспечивается формирование документов, ведомостей по введенным и рассчитанным данным.

Основные операции: расчет заработной платы вспомогательных рабочих, премий, надбавок, вознаграждений по итогам года, получение сводной аналитической и статистической отчетности, формирование отчетности в соответствии с требованиями, вывод печатных форм

проектов. В конфигурации особое внимание уделяется автоматизации процесса ввода и контроля информации.

Подсистема позволяет сократить число и копии документов, оптимизировать маршруты движения документов и алгоритмы их формирования, автоматизировать трудоемкие рутинные операции, высвободить работников для выполнения собственно аналитических и управленческих функций и тем самым повысить эффективность служб цехов, отделов и управлений.

**А.А. Ясонов (УО «ГГТУ им. П. О. Сухого», Гомель)**

Науч. рук. **И.А. Мурашко**, д.т.н., доцент

## **МЕТОДИКА АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ ГЕНЕРАТОРОВ ПСЕВДОСЛУЧАЙНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ НА ОСНОВЕ СВОЙСТВА ДЕЦИМАЦИИ М-ПОСЛЕДОВАТЕЛЬНОСТИ**

В процессе своего создания современное микроэлектронное устройство проходит два вида тестирования – функциональное тестирование и производственное тестирование. Несмотря на схожесть названий, целей и, иногда, методов функционального и производственного тестирования между ними существует большая разница. Цель функционального тестирования – обнаружить ошибки в функционировании устройства или доказать, что устройство выполняет заданные функции без ошибок. Задача производственного тестирования – обнаружить производственные дефекты в уже изготовленном устройстве или доказать, что они отсутствуют. Основные требования при производственном тестировании – обеспечить приемлемый процент покрытия неисправностей (faultcoverage – FC). В настоящее время считается приемлемым FC порядка 95 %.

Очевидным способом производственного тестирования является тестирование устройства с использованием уже существующих функциональных тестов, что довольно часто применяется. Практика показывает, однако, что такой подход не слишком хорош. Во-первых, наборы функциональных тестов для сложных устройств, как правило, довольно объемны. Следовательно, дорогостоящее тестовое оборудование будет сильно загружено, пока будет проверена вся партия кристаллов. Во-вторых, функциональные тесты не предназначены для выявления производственных дефектов, поэтому нельзя быть уверенным в том, что они способны выявить максимальное количество дефектов. Поэтому

практическим стандартом стал способ производственного тестирования с использованием специального порта – TAP- порта, или JTAG интерфейса, который перешел в системы на кристалле по наследству от печатных плат.

Основная идея использования JTAG интерфейса заключается в следующем. На стадии синтеза устройства по RTL описанию в современных компиляторах имеется возможность так устанавливать параметры синтезируемого устройства, что в него вносятся дополнительные цепи, соединяющие все триггеры схемы в одну цепочку (цепочек может быть и несколько). Одновременно компилятор генерирует систему тестовых воздействий для построенной цепочки так, чтобы процент покрытия неисправностей был приемлемым (считается приемлемым процент покрытия равный 97%). В процессе тестирования тестовые комбинации проходят по цепочке до нужных элементов. Достоинствами использования JTAG интерфейса по сравнению с ранее описанным подходом является более эффективное использование тестового оборудования и больший процент обнаруживаемых дефектов. Недостатками использования JTAG интерфейса можно считать необходимость наличия тестового оборудования. Это не позволяет тестировать устройство конечному пользователю, в том числе и процессе эксплуатации, что бывает часто необходимо, например, в системах криптографии. Внесение дополнительных цепей, не существующих на RTL уровне, также не всегда бывает приемлемым, кроме того, они могут привести к неконтролируемой деградации производительности устройства, что в высокопроизводительных системах также неприемлемо.

Есть еще один подход к тестированию устройств – самотестирование (BIST – built-inself-test), в котором отсутствуют указанные недостатки. Основными достоинствами самотестирования является дешевизна, а также возможность тестирования устройства конечным пользователем. Главным достоинством, конечно, является отсутствие необходимости использования какого-либо внешнего дополнительного оборудования.

Основная идея самотестирования – использования внутренних аппаратных ресурсов для целей тестирования. Разумеется, полностью обойтись без дополнительного оборудования в процессе самотестирования невозможно, но желательно, чтобы оно было минимальным. В настоящее время считается приемлемым 10-15% дополнительной аппаратуры при условии достижения нужного процента ФС.

В организации самотестирования можно выделить следующие основные проблемы:

- генерация тестовых воздействий;
- сбор информации;
- внесение дополнительных точек наблюдения;
- переконфигурация устройства или его отдельных частей.

Генерация тестовых воздействий и сбор информации – функции BIST-контроллера. BIST-контроллер – это специализированное устройство, предназначенное исключительно для организации самотестирования. На его долю обычно приходится 90–100 % дополнительного оборудования, следовательно, главное требование, которое к нему предъявляется, в дополнение к указанным функциям – компактность.

Генерация тестовых воздействий – одна из основных проблем, если не самая главная. В принципе, если эта задача удачно решена, то внесение дополнительных точек наблюдения и переконфигурация устройства не нужны: главное – обеспечить требуемый процент покрытия неисправностей. Возможны два решения задачи генерации тестовых воздействий – использовать детерминированные тестовые последовательности и случайные тестовые воздействия. В обоих случаях необходимо, чтобы длина тестовых последовательностей, называемых тестовыми векторами, была минимальной. Это требуется опять же для сокращения времени работы внешнего тестового оборудования, если оно будет использоваться для проверки кристаллов перед корпусированием. В идеале желательно сформировать такую тестовую последовательность, чтобы все элементы схемы переключались из 1 в 0 и наоборот.

Детерминированные тестовые последовательности использовать целесообразно в тех случаях, когда для их хранения разрешено использовать элементы памяти, например, в памяти микропрограмм процессоров или память программ контроллеров. В системах на кристалле, где в обязательном порядке имеется микроконтроллер, который можно использовать для тестирования, и постоянные запоминающие устройства, детерминированные тестовые последовательности в виде фрагментов программ, вполне можно использовать. Однако, этому подходу присущи те же недостатки, что и тестированию с помощью функциональных тестов (см. выше). Генерация детерминированных тестов чисто схемотехническими методами расточительна с точки зрения затрат дополнительной аппаратуры.

Использование случайных тестовых воздействий гораздо экономичнее как с точки зрения аппаратных затрат, так и с точки зрения количества тестовых векторов, необходимых для достижения нужного процента ФС. Для этого целесообразно использовать хорошо известное устройство – сдвигающий регистр с линейными обратными связями (linear-feedback shift register – LFSR).

Чтобы выходная последовательность такого устройства была максимально длинной, необходимо, чтобы обратные связи соответствовали коэффициентам полинома, для которого коэффициент пересчета (возврат в исходное состояние) был  $2^N$ , где  $N$  – число разрядов регистра. На рис. 1 изображена реализация LFSR, соответствующего полиному  $X^8 + X^6 + X^5 + X + 1$ .

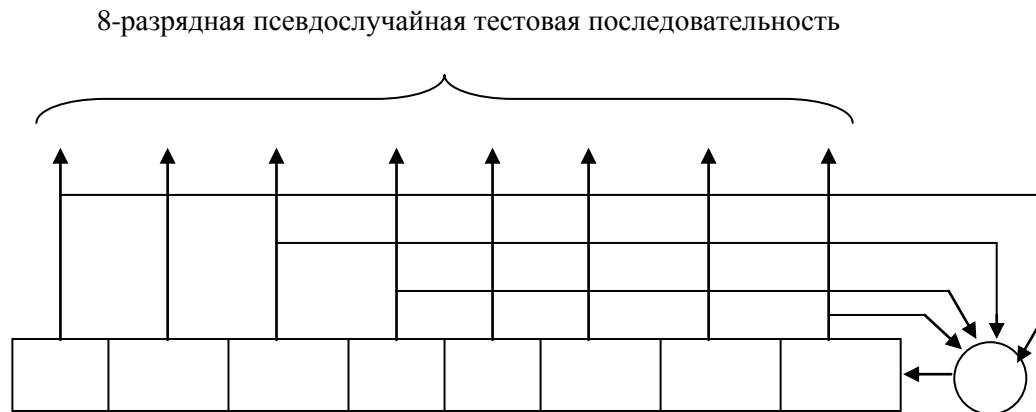


Рисунок 1 – Сдвигающий регистр с линейными обратными связями

Разрядность LFSR выбирают, исходя из практических соображений. Она не может быть меньше  $\log_2(K)$ , где  $K$  – число тестовых векторов. Как правило, 16-битового LFSR достаточно для тестирования большинства цифровых устройств. В случаях, когда LFSR используется для тестирования больших устройств, состоящих из нескольких СФ блоков, его разрядность может быть увеличена до 32.

При разработке встроенных средств самотестирования часто возникает задача подбора порождающего полинома определенной степени для повышения качества тестирования. Таблицы примитивных полиномов содержат, как правило, по одному полиному каждой степени. Кроме того уже для 19-й степени существует уже 27594 примитивных полиномов. Хранить таблицы полиномов не целесообразно из-за их объемов. Поэтому разработаны различные методики формирования множества полиномов заданной степени. Для решения этой задачи так же может быть применено свойство децимации  $M$ -последовательности (последовательности максимальной длины). Алгоритм нахождения примитивных полиномов достаточно прост.

1. Из таблицы выбирается полином заданной степени  $m$ . Для него строится порождающая матрица  $V$ , и находится период  $M$ -последовательности  $L = 2^m - 1$ .

2. Выбирается коэффициент децимации  $q$ . При этом на  $q$  накладыв-



ваются следующие ограничения:

- $(L, q) = 1$ , то есть  $L$  и  $q$  должны быть взаимно просты;
- $q \neq 2^i$ ,  $i=1, 2, 3, \dots$ . Данное ограничение связано с тем, что при  $q = 2^i$  новый полином будет совпадать с исходным;
- $q \neq (L - 1)2^i$ ,  $i=1, 2, 3, \dots$ . Данное ограничение связано с тем, что при  $q \neq (L - 1)2^i$  формируется взаимно-обратный полином, который можно вычислить исходя из следующего выражения. Для полинома  $\varphi(x)$  степени  $m$  существует взаимно-обратный полином  $\varphi^{-1}(x)$ , который связан с ним соотношением  $\varphi^{-1}(x) = x^m \varphi(1/x)$ . Если  $\varphi(x)$  примитивный, то и взаимно-обратный полином так же является примитивным.

**Ю.Н. Яшманов (УО «ГГУ им. Ф. Скорины», Гомель)**

Науч. рук. **М.И. Жадан**, к.ф.-м.н., доцент

## **ИНФОРМАЦИОННО-АНАЛИТИЧЕСКАЯ СИСТЕМА IBM COGNOS BI**

IBM Cognos BI – инструмент для построения эффективной системы управленческого учета, предназначенный для организации процесса сбора, многомерного анализа, подготовки и предоставления информации, которая необходима для поддержки принятия решений руководством компании.

Основная задача IBM Cognos BI состоит в своевременном предоставлении полной, достоверной информации лицам, ответственным за принятие решений.

Простой и интуитивно понятный инструментарий позволяет создавать аналитическую отчетность любой сложности. Интерфейс конечного пользователя рассчитан на пользователей информации и не требует постоянной поддержки ИТ-специалистов. Поддержка в отчетах функций drill-down и drill-through обеспечивает мгновенный переход от общего к более детальному представлению информации.

Система легко интегрируется со всеми уже существующими источниками информации на предприятии. Встроенная функция оповещения позволяет не тратить время на ежедневное отслеживание каких-либо показателей или событий. Надо лишь определить элементы отчета, по которым необходимо получать оповещения, задать условия и способ информирования, и система будет посылать оповещения при наступлении данных событий.

Основной задачей, которую нужно решить при внедрении системы IBM

Cognos BI, является построение хранилища данных – информационного ядра, собирающего необходимую информацию из всех существующих источников данных на предприятии. Данные структурируются, унифицируются, прописываются все необходимые взаимосвязи, после чего информация готова для анализа с помощью мощных инструментов IBM Cognos BI. Работа по построению хранилища данных составляет порядка 75–80 % трудозатрат по внедрению системы.

Система бизнес-анализа, построенная на платформе IBM Cognos BI, позволяет проводить анализ данных, полученных из нескольких источников, а также существенно сократить время и трудозатраты на обработку информации и подготовку отчетов. Руководство получает возможность доступа к информации в любой момент времени из любой географической точки.

В предлагаемой работе реализованы различные приемы создания компонентов отчета и их свойств, использование переменных и параметров:

- Добавление в отчет различных полезных элементов, которые повышают функциональные возможности отчета. Это использование различного форматирования, разбиение отчета на секции и подсекции, добавление графических элементов, преобразование отчета List в отчет Crosstab, использование таблиц для придания дружелюбного пользовательского внешнего вида.

- Создание многостраничного отчета с титульным листом.

- Использование в отчете условного форматирования (включает в себя создание многоязычных отчетов с использованием языковой переменной, выделение необходимых данных с помощью логической, строковой переменных или свойства Conditional Style, скрывание/показ объектов отчета в зависимости от условия с помощью переменных и свойства объекта StyleVariable).

- Переход из основного отчета в детально разработанный отчет. Данный переход используется для более полного отображения информации, выбранной в основном отчете. Реализуется в отчете с помощью свойства Drill-Through Definition и имеет вид ссылки. При создании перехода необходимо передать параметры из основного отчета в детальный отчет.

- Вывод выбранных пользователем значений параметров для создания более простых для понимания отчетов или вывод переданных в детальный отчет параметров для улучшения восприятия информации.

- Использование макро запросов. Макро запрос – это выражение, которое вкладывается непосредственно в SQL-запрос и динамически

изменяет SQL во время выполнения.

– Создание колонок, выводящих различные поля динамически во время работы отчета. Данный прием включает создание параметра, создание элемента меню для выбора пользователем значения данного параметра, создание вычисляемого элемента данных, использующего данный параметр.

– Использование различных видов полей выбора параметров, значения которых используются в отчете для условного форматирования, создания динамических элементов данных, фильтрации данных, сортировки данных, создания переменных.

Следует отметить, что это далеко не полный список приемов и практик, применяемых при создании разнообразных отчетов, и он ограничивается только фантазией автора отчета.

## Содержание

### Секция 3 «Автоматизация исследований»

<b>Алексейчук О.А.</b> Автоматизация учета электрооборудования АЗС.	3
<b>Батанов М.И.</b> Интеграция приложений в социальные сети.....	4
<b>Бердников Е.Е.</b> Модернизация ЛВС предприятия ЧСУП «ВТИ» ....	6
<b>Бокатюк С.С.</b> Интернет – магазин для ООО «Фотопереходник»....	7
<b>Бокий М.Ю.</b> Разработка подсистемы интеграции данных в проекте консолидации финансовых отчетов.....	8
<b>Борихин И.А.</b> Средства учета дипломных проектов.....	11
<b>Бугримов Д.О.</b> Разработка компонента по загрузке ресурсов в DAM для корпоративного сайта www.eram.com.....	12
<b>Бугримов Д.О.</b> Разработка компонентов пользовательского интерфейса для корпоративного сайта www.eram.com.....	15
<b>Будько Д.В.</b> Применение фреймворка Hibernate для разработки веб-приложения.....	16
<b>Будько Д.В.</b> Управление доступа к данным в системе заказов нематериальных ресурсов.....	17
<b>Васильев О.А.</b> Разработка клиента передачи данных для Web-сервиса Amazon Glacier.....	20
<b>Васильев О.А.</b> Применение JSON в построении клиента передачи данных для Web-сервиса Amazon Glacier.....	21
<b>Васильев О.А.</b> Организация многопоточности в клиенте передачи данных для Web-сервиса Amazon Glacier.....	22
<b>Веренчиков Р.А.</b> Реализация шаблона Model View Controller при создании Web-системы управления проектами.....	23
<b>Витко А.А.</b> Система управления производственным оборудованием SIMATIC PCS7.....	24
<b>Воробьева А.С., Кучеров А.И.</b> Исследование интернет-магазинов Республики Беларусь.....	27
<b>Гетиков Д.В.</b> Использование агентов в Web-приложениях, созданных в корпоративной среде документооборота Lotus Notes.....	29
<b>Глушаков И.Г.</b> Автоматизация регистрации технико-тактических игровых действий футболистов.....	31
<b>Гончаров А.С.</b> Тест тьюринга и виртуальные собеседники.....	34
<b>Григорьев Д.Г.</b> Прошивка и модернизация bios.....	37
<b>Григорьев Д.Г.</b> Применение технологии Oracle в Беларуси.....	40
<b>Гриневич М.И.</b> Современные технологии разработки Web-сайтов..	41

<b>Гриневич М.И.</b> Создание отчётов для предприятий торговли.....	43
<b>Демиденко А.С.</b> Компьютерная диагностика автомобиля.....	44
<b>Деревянко Д.В., Абрамов Е.С.</b> Автоматизация определения погрешности измерителей линейных размеров.....	47
<b>Дешкович Р.В.</b> Разработка Web-ресурса решения задач по физике для студентов.....	48
<b>Диваков Н.Н.</b> Создание автоматизированной системы учета педагогического и ученического состава школы «Гимназия №56».....	50
<b>Дробышевская О.В.</b> Алгоритм сортировки массивов QuikSort.....	51
<b>Зайченко А.Г.</b> Использование внешней памяти для конечно-элементного моделирования системы «Плита – грунтовое основание» .....	52
<b>Зубов Г.А.</b> Использование физических законов для описания взаимодействий трехмерных объектов.....	53
<b>Киселев И.В.</b> Основные этапы создания Web-сайта.....	54
<b>Кисель Н.В.</b> Средства автоматизации моделирования и оценки моделей автоматизированных систем.....	57
<b>Кисло Д.А.</b> Средства автоматизации построения моделей предприятий .....	58
<b>Климович М.А.</b> Оптимизация приложений в Unity3d.....	60
<b>Климович А.В.</b> Разработка проекта диагностического стенда для ремонта бортовых компьютеров предприятия «Цветотрон».....	62
<b>Ковалёв А.А.</b> Функциональные возможности сервиса для экспорта журнала регистрации исключительных ситуаций в частное облако..	64
<b>Ковалёв А.А.</b> Архитектурные решения реализации сервиса для экспорта журнала регистрации исключительных ситуаций в частное облако.....	65
<b>Ковалёва Е.А.</b> Проектирование системы автоматизации услуг строительной организации.....	68
<b>Козлов И.А.</b> Программа «Remember these verbs» .....	69
<b>Козырев И.В.</b> Интеграция системы 1С: предприятие 8 с другими информационными системами.....	72
<b>Коледа Я.А.</b> Создание наглядного пособия для курса «Архитектура ЭВМ».....	75
<b>Кондратенко М.Г.</b> Интернет-банкинг в Республике Беларусь.....	76
<b>Королёва А.В.</b> Разработка подсистемы по учету запасов предприятия в среде 1С: предприятие.....	78
<b>Костюкевич В.В.</b> Создание Cognos приложения на платформе Cognos BI.....	79
<b>Костюкевич В.В.</b> Программный продукт Framework Manager.....	82

<b>Костюченко В.И.</b> Web-ориентированное приложение «Issue Tracker».	83
<b>Котович А.А.</b> Облачный хостинг и облачные вычисления.	85
<b>Крупский А.А.</b> Высоконагруженная система пакетной обработки информации.	87
<b>Куликова Е.В.</b> Развитие 3G и 4G мобильной связи.	88
<b>Купреева А.В.</b> Топ браузеров для различных ОС.	90
<b>Лебедева А.В. , Михайлич Я.В.</b> Программная поддержка дистанционного обучения.	93
<b>Леванцов С.В.</b> Плюсы и минусы технологии IPoE.	96
<b>Левченко А.В.</b> Разработка плагина на платформе Google для индивидуализации заданий.	98
<b>Леоненко А.М.</b> Описание ERP-системы SAP R/3 и примеры её применения на белорусских предприятиях.	99
<b>Марченко К.С.</b> Язык запросов в 1С предприятие 8.2.	101
<b>Микулинская О.И.</b> О разработке системы метрологического учета ТНПА и средств измерений.	103
<b>Митченко Ю.С.</b> Организация единовременных выплат студентам в подсистеме «1С:Стипендия».	104
<b>Митченко Ю.С.</b> Назначение основных начислений и удержаний в подсистеме «Стипендия»	106
<b>Мостовцов А.В.</b> Проектирование системы автоматизации составления расписания телевидения.	108
<b>Навныко Д.А.</b> Микроконтроллеры.	110
<b>Навныко Н.А.</b> Автоматизация работы фирмы по ремонту бытовой техники.	112
<b>Небышинец И.А.</b> Электронная торговая площадка.	113
<b>Пименов А.О.</b> Подключение и работа с HRM устройствами Garmin.	115
<b>Пименов А.О.</b> Обработка данных и взаимодействие с GPS-устройствами.	116
<b>Пиюш Барскар</b> Минимизация энергопотребления многоходовых сумматоров по модулю два.	119
<b>Потапенко А.В., Кузьменко О.А.</b> Влияние современного экономического кризиса на информационные технологии.	122
<b>Рагин В.И.</b> Замена двоичной логики, увеличение производительности.	125
<b>Речицкий А.Ю.</b> Автоматизация документооборота регистратуры в учреждении «Мозырский городской родильный дом».	126
<b>Ролич А.М.</b> Организация системы делопроизводства.	128
<b>Романова О.Ю., Нестерович И.С.</b> Состояние и перспективы IT-рынка	130

<b>Рубин В.А.</b> Разработка веб-сервиса «Календарь событий».....	133
<b>Рябиков Ф.С.</b> Моддинг BIOS видеокарты.....	134
<b>Савицкий Д.А.</b> Управление списками задач на платформе iOS.....	137
<b>Савченко А.А.</b> Структура современных систем контроля и учета энергоресурсов.....	138
<b>Сапанович А.В.</b> Разработка системы автоматизации работы с расписанием занятий факультета.....	140
<b>Семенцов С.В.</b> Обработка результатов геофизических исследований при проектировании нефтяных скважин.....	143
<b>Семионов А.А.</b> Цветокоррекция с помощью кривых.....	144
<b>Семионов А.А.</b> Применение эффекта реверберации при обработке звука.....	146
<b>Слука С.В.</b> Новое в версии 8.2 платформы 1С:предприятия 8.....	149
<b>Слюсарева М.А.</b> Вопросы автоматизации планирования производства автоматизированных систем.....	152
<b>Судакова Е.С.</b> Основные функции и особенности Lotus Notes.....	153
<b>Судакова Е.С.</b> Разработка XPages-страниц с использованием JavaScript в Lotus Notes.....	156
<b>Тишко Д.А.</b> Разработка базы данных автоматизации учёта оборудования для руп «Гомельэнерго».....	157
<b>Трибилустов А.А.</b> Система управления проектами и задачами JIRA.....	159
<b>Умаров Т.Ф.</b> Повышение эффективности накопителей с использованием RAID-массивов.....	162
<b>Фащенко Д.Л.</b> Функциональные возможности проекта АСУ «ДКР» .	165
<b>Фащенко Д.Л.</b> Автоматизация перевозок на железнодорожном транспорте в проекте АСУ ДКР.....	165
<b>Флерко А.Г.</b> Автоматизация передвижных отделений почтовой связи предприятия «Лельчицкий районный узел почтовой связи»....	168
<b>Хобня А.И.</b> Унификация процесса разработки имитационной модели с использованием языка UML.....	170
<b>Хобня А.И.</b> Проектирование структуры имитационной модели средствами языка UML.....	172
<b>Хомицкий Ю.В.</b> Антивирусная защита мобильных устройств.....	173
<b>Хоруженко А.Н.</b> Имитационная модель рекламного трафика.....	177
<b>Цырибко Д.Б.</b> Автоматизация разработки Web-сайтов.....	178
<b>Чеботаревский А.С.</b> Алгоритм поиска пути jump point search.....	179
<b>Чернышев А.И.</b> Моддинг принтеров.....	181
<b>Шабалин В.А.</b> Особенности автоматизации тестирования игровых приложений.....	184

<b>Шамело А.В.</b> Развитие в Республике Беларусь рынка банковских платежных карточек.....	185
<b>Шедова И.Г.</b> Разработка диалогового окна при создании отчетов в среде 1С: предприятие 7.7 .....	187
<b>Якубова А.С.</b> Использование автоматизированной системы управления «Экспресс-3» Белорусской железной дорогой.....	189
<b>Якубова С.С.</b> Применение автоматизированных систем трансляции спортивных событий в режиме LIVE.....	192
<b>Ярчук В.Н.</b> Задачи ввода, хранения и обработки информации на предприятии.....	195
<b>Ясонов А.А.</b> Методика автоматизации проектирования генераторов псевдослучайных последовательностей на основе свойства децимации M-последовательности.....	197
<b>Яшманов Ю.Н.</b> Информационно-аналитическая система IBM Cognos BI.....	201



Научное издание

# **АКТУАЛЬНЫЕ ВОПРОСЫ ФИЗИКИ И ТЕХНИКИ**

**Материалы  
II Республиканской научной конференции  
студентов, магистрантов и аспирантов,  
посвященной памяти первого ректора  
Гомельского государственного университета,  
академика Владимира Алексеевича Белого**

(Гомель, 18 апреля 2013 года)

**В двух частях**

**Часть 2**

Подписано в печать 14.06.2013. Формат 60x84 1/16.  
Бумага офсетная. Ризография. Усл. печ. л. 12,3.  
Уч.-изд. л. 13,5. Тираж 20 экз. Заказ 419.

Издатель и полиграфическое исполнение :  
учреждение образования  
«Гомельский государственный университет  
имени Франциска Скорины».  
ЛИ № 02330/0549481 от 14.05.2009.  
Ул. Советская, 104, 246019, Гомель.

