

Министерство образования Республики Беларусь

**Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»**

**АКТУАЛЬНЫЕ ВОПРОСЫ
ФИЗИКИ И ТЕХНИКИ**

**Материалы
III Республиканской научной конференции
студентов, магистрантов и аспирантов**

(Гомель, 17 апреля 2014 года)

В двух частях

Часть 2

Гомель
ГГУ им. Ф. Скорины
2014

УДК 530

«Актуальные вопросы физики и техники», III Республиканская научная конференция студентов, магистрантов и аспирантов, (2014, Гомель). III Республиканская научная конференция студентов, магистрантов и аспирантов «Актуальные вопросы физики и техники», 17 апреля 2014 г. : [материалы] : в 2 ч. Ч. 2 / редкол. : А. В. Рогачев (гл. ред.) [и др.]. – Гомель : ГГУ им. Ф. Скорины, 2014. – 270 с.

ISBN 978-985-439-652-1

ISBN 978-985-439-651-4 (Ч. 2)

В сборник вошли материалы докладов, представленные на III 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. Мышковаец, В. Е. Гайшун, Е. Б. Шершневец, Т. П. Желонкина

ISBN 978-985-439-652-1
ISBN 978-985-439-651-4 (Ч. 2)

© УО «Гомельский государственный университет им. Ф. Скорины», 2014

Секция 3 «Автоматизация исследований»

Председатели:

Левчук Виктор Дмитриевич, канд. тех. наук, доцент,
Воруев Андрей Валерьевич, канд. тех. наук, доцент

Е.С. Абрамов (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **А.И. Кучеров**, ст. преподаватель

МОДЕЛЬ ПРОЕКТИРОВАНИЯ MVC

Model-view-controller (MVC, «модель-представление-контроллер») – схема использования нескольких шаблонов проектирования, с помощью которых модель данных приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные.

Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области.

Основная цель применения этой концепции состоит в разделении бизнес-логики (модели) от её визуализации (представления, вида). За счет такого разделения повышается возможность повторного использования. Наиболее полезно применение данной концепции в тех случаях, когда пользователь должен видеть те же самые данные одновременно в различных контекстах и/или с различных точек зрения.

К одной модели можно присоединить несколько видов, при этом не затрагивая реализацию модели. Например, некоторые данные могут быть одновременно представлены в виде электронной таблицы, гистограммы и круговой диаграммы.

Не затрагивая реализацию видов, можно изменить реакции на действия пользователя (нажатие мышью на кнопку, ввод данных), для этого достаточно использовать другой контроллер.

Ряд разработчиков специализируется только в одной из областей: либо разрабатывают графический интерфейс, либо разрабатывают бизнес-логику. Поэтому возможно добиться того, что программисты, занимающиеся разработкой бизнес-логики (модели), вообще не будут осведомлены о том, какое представление будет использоваться.

Концепция MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента (рисунок 1).

Модель (англ. Model). Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.

Представление, вид (англ. View). Отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма (окно) с графическими элементами.

Контроллер (англ. Controller). Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.



Рисунок 1 – Модель MVC

Важно отметить, что как представление, так и контроллер зависят от модели. Однако модель не зависит ни от представления, ни от контроллера. Тем самым достигается назначение такого разделения: оно позволяет строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели.

Литература

1. Осборн, Д. JavaBeans: Developing Component Software in Java, 2-е изд.: Компьютер/ Д. Осборн. – Литера, 2013. – 389 с.
2. Эккель, Б. Философия Java. Библиотека программиста. 4-е изд. / Б. Эккель. – М.: СПб: Питер, 2012. – 640 с.
3. Блинов, И.Н. Java. Промышленное программирование, / И.Н. Блинов, В.С. Романчик. – М.: УниверсалПресс, 2007. – 704 с.

Е.С. Абрамов (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **А.И. Кучеров**, ст. 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).

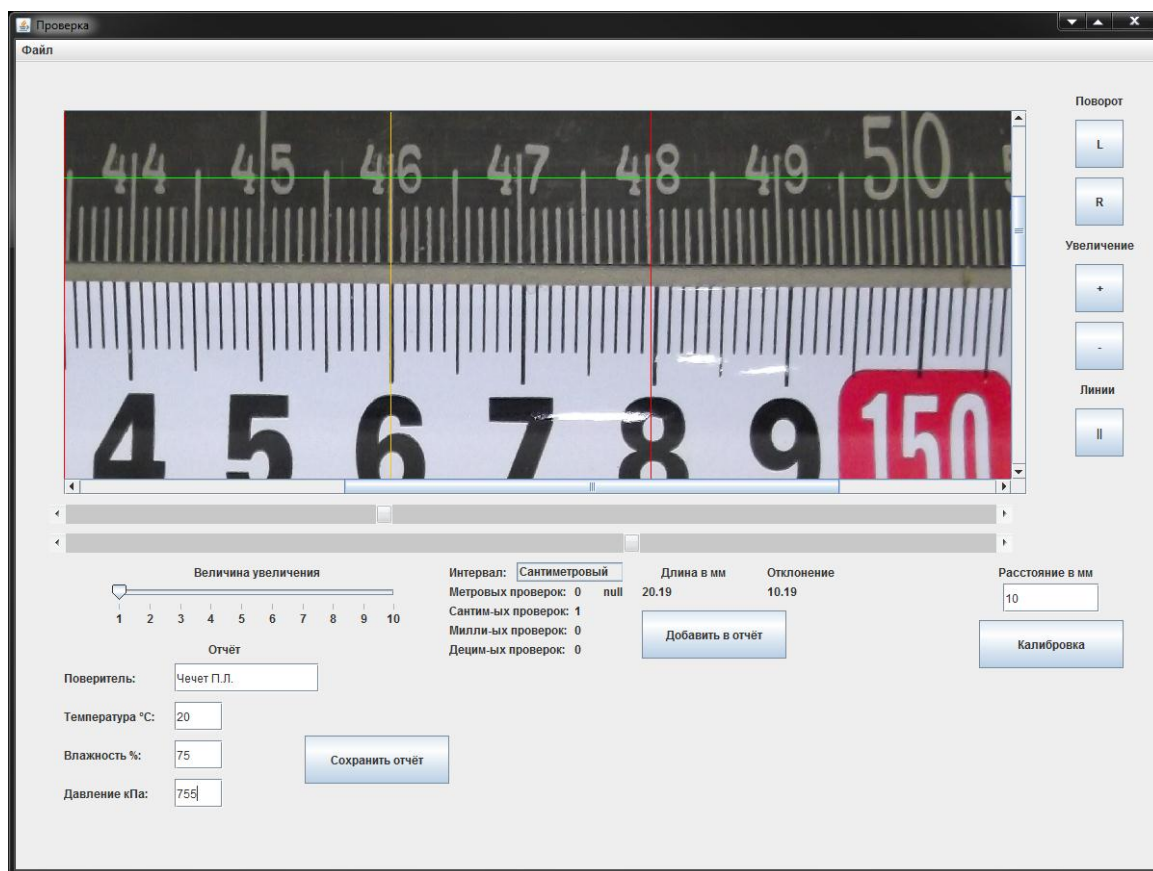


Рисунок 1 – Вид программы

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 мм, а второй – на 47 мм 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. JavaServer Faces. / Д. Гери, К. Хорстман // Библиотека профессионала, 3-е изд.: Пер. с англ. – М. : Издательский дом «Вильямс», 2011. – 544 с.

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

П.Н. Алексеенко (УО БрГТУ, Брест)

Науч. рук. **В.И. Хвещук**, канд. тех. наук, доцент

АРМ ДЛЯ РАСЧЕТА И РАСПРЕДЕЛЕНИЯ УЧЕБНОЙ НАГРУЗКИ НА КАФЕДРЕ ВУЗА

Расчет и распределение учебной нагрузки для преподавателей является одной из сложных и трудоемких работ, выполняемых на каждой кафедре вуза. Особую сложность представляет задача распределение нагрузки между преподавателями, которая носит итеративный характер и при ее решении необходимо выполнять многочисленные требования и ограничения.

В данной работе представлен вариант автоматизации решения данной задачи в виде автоматизированного рабочего места (АРМ), которое реализовано на основе клиент – серверной архитектуры. АРМ состоит из серверной и клиентской части.

Серверная часть АРМ – это база данных (БД), представленная в виде совокупности взаимосвязанных таблиц, реализованных с использованием T-SQL.

Клиентская часть АРМ – это пользовательское приложение, обеспечивающее удобный диалоговый интерфейс и реализовано на С#. Клиентское приложение обеспечивает следующие группы возможностей: формирование исходных данных и расчет нагрузки для кафедры; распределение нагрузки между сотрудниками кафедры; документирование результатов; справочные функции и другие.

Исходной информацией для применения АРМ являются документы, которые используются на кафедре при решении задач расчета нагрузки и ее распределении между преподавателями (учебный план, закрепление дисциплин за кафедрами, контингент студентов, штатное состав кафедр, нормы для расчета учебной нагрузки и другие документы).

В качестве выходной информации АРМ являются документы - расчет учебной нагрузки по кафедре, распределение нагрузки между преподавателями кафедры и другие.

На данный момент АРМ находится на стадии комплексных испытаний. Для АРМ разрабатывается комплект эксплуатационной документации. Планируется внедрение АРМ в деятельность кафедр БрГТУ.

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

Науч. рук. **П.Л. Чечет**, канд. тех. наук, доцент

ВЕДЕНИЕ ЛОГОВ В ПРИЛОЖЕНИЯХ НА JAVA

В современных приложениях важное место занимает процесс сохранения некоторой статистики, характеризующей процесс работы приложения, возникновение ошибок. Это называется ведением логов или логированием. Логирование – это пример дневника, в который записываются все важные действия, ошибки, сбои и т.д. Лог-файл – файл, в который помещен журнал логирования.

Log4j – самая первая унифицированная система логирования. Из-за отсутствия аналогов на рынке и хорошей проработки архитектуры, этот фреймворк быстро добился популярности и широко используется в наши дни. На данный момент эта система практически лишена недостатков, разработка проекта не ведется, а только исправляются ошибки.

Основные понятия в этой библиотеке – *logger*, *appender* и *layout*.

Logger – это объект класса `org.apache.log4j.Logger`, который используется для вывода данных и управления уровнем вывода. Стандартные уровни: TRACE, DEBUG, INFO, WARN, ERROR, FATAL, OFF.

Appender – это то место, куда приходят сообщения о логировании (консоль, файл и т.д.). Есть ряд стандартных аппендеров, но существует возможность всегда создать свой аппендер. Все аппендеры реализуют интерфейс `org.apache.log4j.Appender`. У одного логгера может быть несколько аппендеров, а к одному аппендеру может быть привязано несколько логгеров.

Консольный аппендер: `org.apache.log4j.ConsoleAppender`. Основные-файловые аппендеры:

- `org.apache.log4j.FileAppender`;
- `org.apache.log4j.RollingFileAppender`;
- `org.apache.log4j.varia.ExternallyRolledFileAppender`;
- `org.apache.log4j.DailyRollingFileAppender`.

Компоновка (*layout*) – это унаследованный от `org.apache.log4j.Layout` объект для конфигурирования формата вывода. Стандартные компоновщики:

- `org.apache.log4j.SimpleLayout`;
- `org.apache.log4j.HTMLLayout`;
- `org.apache.log4j.xml.XMLLayout`;
- `org.apache.log4j.TTCCLayout`;
- `org.apache.log4j.PatternLayout`;
- `org.apache.log4j.EnhancedPatternLayout`.

Каждый компоновщик поддерживает позиционное форматирование. Это позволяет выводить данные в требуемом формате. Гибкие возможности по созданию логов в Java позволяют решать практически любые задачи по ведению статистики работы приложений.

Ю.М. Астапович (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **А.В. Ворухев**, канд. тех. наук, доцент

АВТОМАТИЗАЦИЯ РАСЧЕТА СМЕТ ПО ПРОЕКТНО-ОБСЛЕДОВАТЕЛЬСКИМ РАБОТАМ

Существует около тридцати действующих Справочников базовых цен на проектные работы и инжиниринговые услуги для строительства. Данные ценовые документы предназначены в качестве базы для формирования договорных цен между заказчиком и исполнителем. В каждом

сборнике приводятся базовые цены на выполнение определенных видов работ. Стоимость работ определяется применением повышающих и корректирующих коэффициентов (К). Они соответствуют определенным условиям. Некоторые коэффициенты могут иметь постоянное значение или варьирующее, зависит от показателей проектируемых объектов (мощность, протяженность, площадь, емкость и т.п.). В последнем случае вычисление будет происходить по определенной формуле, либо определяться по данным из дополнительных таблиц Справочников. ООО «ИтернаГрупп» довольно часто выполняет заказы на обследование и реконструкцию, поэтому было целесообразно для исполнения дипломного проекта использовать «Сборник цен на обследовательские работы по выявлению технического состояния, разработке мероприятий по ремонту и усилению строительных конструкций жилых, общественных и производственных зданий и сооружений».

Используя данные сборника составляется сводная смета. Сводные расчеты цены на ремонт или реконструкцию зданий и сооружений определяют лимит средств нужных для полного завершения строительных работ, служат основанием для начала финансирования. Смета предоставляет Заказчику информацию о всех используемых показателях (применяемых коэффициентах, их величине, затраты по отдельным видам работ и расчеты полной стоимости проекта).

Сметная стоимость на протяжении разработки проекта может быть неоднократно скорректирована, в случае уточнения методов и характера работ. Это приводит к определенным неудобствам и затратам. А данное разработанное приложение позволяет сократить процесс составления сметы, избавляет от нудных и рутинных расчетов, экономит время. При его использовании ошибки при подсчете на калькуляторе полностью исключаются.

Для создания проекта применяется Microsoft Access 2013 с использованием объектно-ориентированного языка Visual Basic.

Была осуществлена следующая работа:

- 1 Изучен сборник цен на обследовательские работы по выявлению технического состояния, разработке мероприятий по ремонту и усилению строительных конструкций жилых, общественных и производственных зданий и сооружений;

- 2 Определены методические указания, которыми следует руководствоваться при составлении сметы для определения общих коэффициентов, стоимости сбора исходных данных, стоимости обмерных работ, стоимости выполнения работ по обследованию, стоимости разработки решений по ремонту и усилению строительных конструкций, и конечных данных (итогов);

3 Дополнительно была изучена и проанализирована необходимая литература для реализации проекта;

4 В режиме конструктора таблиц разработаны таблицы «Proektir» (Проектировщики) и «Smety» (Сметы). Они содержат всю информацию созданной базы. Произведена нормализация базы данных для уменьшения избыточности информации в таблицах, построения оптимальной структуры таблиц и связей. Были установлены связи между полем Id таблицы «Proektir» и полями таблицы «Smety». Таким образом поле «proektirId» становится внешним ключом;

5 В режиме конструктора форм разработаны формы для отображения данных в базе MS Access:

– форма «ListSmet» (Список смет). Форма «ListSmet» (Список смет) позволяет создавать, редактировать, удалять и выводить сформированную смету на печать. Так же она предназначена прежде всего для хранения всех смет, которые будут созданы и не удалены;

– форма «Смета». Форма «Смета» упрощает ввод данных. В данной форме и реализуется расчёт смет. Флажки позволяют определить логическое значение. С помощью кнопок осуществляется быстрый переход к формам с графическими подсказками. Благодаря мастеру кнопок Access создаются процедуры обработки событий на языке Visual Basic. Эти процедуры связываются с событиями. Например, «Нажатие кнопки» (On Click) и выполняет выбранное действие. Можно открыть процедуру, ознакомиться с программой и при необходимости внести в программу изменения. Для открытия процедуры обработки события надо в режиме конструктора вызвать контекстно-зависимое меню кнопки и выбрать команду «Обработка событий» (Build Even). Наличие форм с графическими подсказками способствует ускорению и облегчению в поиске нужной информации для использования в расчётах проектных работ. Больше нет необходимости использовать бумажные носители и тратить время на перелистывание Сборника в поиске нужной информации (пункта, таблицы). Так же в форме «Смета» предоставляется возможность заполнения всех полей и в дальнейшем получения автоматического расчёта таких данных, как:

– K_1 (объём);

– $K_{\text{общ}}$ (общий коэффициент);

– итога по п.2 (конечный результат по этапу работ «Стоимость сбора исходных данных»);

– итога по п.3 (конечный результат по этапу работ «Стоимость обмерных работ»);

– итога по п.4 (конечный результат по этапу работ «Стоимость выполнения работ по обследованию»);

- итого по п. 5 (конечный результат по этапу работ «Стоимость разработки решений по ремонту и усилению строительных конструкций»;
- итого на 100м³;
- итого на объём здания;
- налог по упрощённой системе налогообложения 5%;
- итого по смете (для вычисления вышеперечисленных значений писался программный код процедуры на языке Visual Basic).

6 В режиме конструктора отчетов создан отчёт для форматированного представления данных, которые выводятся на экран с последующей печатью. (Для конвертации числа в текст на языке Visual Basic была создана функция «NumberToText».)

Результаты тестирования показали, что разработанное приложение корректно и может с успехом применяться на практике для решения задач в рассматриваемой области.

Проект действительно позволяет облегчить и ускорить расчет смет по проектно-обследовательским работам, тем самым, создавая возможность скорейшего сотрудничества между заказчиком и проектировщиком. А в условиях жесткой конкуренции очень важно быстро реагировать.

М.В. Бакунов (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **Е.А. Ружицкая**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА ФРЕЙМВОРКА XEMAX ENGINE ДЛЯ CORONA SDK

Рынок мобильных телефонов это самый быстроразвивающийся сегмент рынка. Обилие разнообразных моделей телефонов создает на этом рынке большую конкуренцию. Множество производителей стараются включить в телефоны как можно больше различных функций.

Фреймворк Xemax Engine спроектирован как вспомогательный механизм для построения приложений на кросс-платформенной среде разработки мобильных приложений Corona SDK. Сам фреймворк состоит из составных частей. Основными элементами являются программное обеспечение SceneCreator и программное ядро для работы с функционалом Corona SDK.

Ядро фреймворка представляет собой надстройку над стандартными библиотеками и настроечными файлами Corona SDK и так же имеет дополнительные функции, расширяющие возможности разработчика. Главными причинами разработки данного продукта были следующие:

1. Увеличения быстродействия приложения за счет более продуман-

ного распределения внутренней памяти. В фреймворк встроен memory manager, который следит за освобождением памяти от текстур и других элементов. Так же он позволяет установить места отображения заставок, чтобы загрузка элементов сцены была незаметным.

2. Уменьшение времени разработки приложения за счет использования более удобных и производительных функций

3. Уменьшения «порога вхождения» в технологию из-за интуитивно понятных функций и принципов разработки

4. Построение отдельных сцен в редакторе сцен и подключения ее в несколько строк вместо того, чтобы описывать каждый элемент в отдельности, что потребует более сотни строк для десятка элементов. В то время как загрузка сцен с помощью редактора сцен не изменяется от количества объектов на экране. Добавлением элементов занимается механизм фреймворка, который после окончания работы предоставляет полный доступ к отдельным элементам сцен, а так же позволяет безопасно для приложения выгрузить эти объекты из памяти.

В состав ядра вошли множество различных модулей, которые предоставляют широкий спектр возможностей для работы с функциями Corona SDK, сценами, экспортированными из SceneCreator, рекламными площадками, внутренними покупками, памятью, обработчиками, различными визуальными эффектами, масштабированием, музыкой и т.д.

С.Я. Бендерская (УО «ГГУ им. Ф. Скорины», Гомель)

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

ПРАКТИЧЕСКИЕ И ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ ПО МОДЕРНИЗАЦИИ ЛВС НА ПРИМЕРЕ РУП «ГОМЕЛЬЭНЕРГО»

Компьютерная сеть (вычислительная сеть, сеть передачи данных) – система связи компьютеров или вычислительного оборудования (серверы, маршрутизаторы и другое оборудование). Для передачи данных могут быть использованы различные физические явления, как правило – различные виды электрических сигналов, световых сигналов или электромагнитного излучения.

Локальная вычислительная сеть (ЛВС, локальная сеть; англ. Local Area Network, LAN) – компьютерная сеть, покрывающая обычно относительно небольшую территорию или небольшую группу зданий. Компьютеры, подключенные к сети, могут выполнять две функции: они могут быть рабочими станциями или серверами.

Рабочая станция (англ. workstation) – комплекс аппаратных и программных средств, предназначенных для решения определённого круга задач.

Сервер – выделенный из группы рабочих станций компьютер, который предназначен для выполнения конкретной сервисной задачи без участия людей. Участие человека и консоли (монитора, мышки, клавиатуры) необходимо только при настройке, при аппаратно-техническом обслуживании и управлении нештатными ситуациями. Серверы обеспечивают более безопасный общий доступ к файлам и ресурсам. К примеру, в серверной сети сотрудники одной компании могут одновременно иметь доступ к интернету и к электронной почте компании. Серверы размещаются в так называемых серверных комнатах, специально оборудованных помещений. Управлением серверами занимаются квалифицированные специалисты – системные администраторы.

Модель OSI состоит из 7 уровней:

- прикладной;
- представительский;
- сеансовый;
- транспортный;
- сетевой;
- канальный;
- физический.

Технологии локальных сетей реализуют, как правило, функции только двух нижних уровней модели OSI – физического и канального.

Филиал «Гомельские электрические сети», один из самых крупных в составе РУП «Гомельэнерго», образован 1 июля 1957 года на базе цеха сетей и подстанций Гомельской электростанции.

Основной задачей филиала «Гомельские электрические сети» является обеспечение надёжной и бесперебойной работы оборудования, установок, сооружений и передаточных устройств, гарантирующее выполнение главной задачи по удовлетворению потребителей в электрической энергии.

В ЛВС входит кабельная система, пассивное и активное оборудование.

Активное оборудование делится на:

- оборудование ядра ЛВС;
- оборудование рабочей группы;
- оборудование внешних магистральных линий.

В качестве оборудования ядра ЛВС используются коммутатор Catalyst 2950T-24 фирмы Cisco Systems и коммутатор Planet FNSW-2401 фирмы Planet Technology Corp.

Коммутатор Catalyst 2950T-24 24 портами 10/100BaseTX и двумя 10/100/1000BaseT портами для объединения в стек и подключения серверов.

Протокол CDP позволяет с помощью программного обеспечения CiscoWorks автоматически исследовать топологию сети.

Неблокируемая коммутационная матрица 8.8 Гб/с позволяет достичь максимальной пропускной скорости коммутатора 4.4 Гб/с. Максимальная производительность – 3.6 млн.пак/с? Максимальное количество MAC адресов сети – 8000 адресов. Технология FastEtherChannel позволяет объединить до 4 портов в единую линию. Коммутатор имеет 16 Мб ОЗУ, 8 Мб Flash и 8 Мб буферной памяти.

Управление коммутатором может осуществляться как через встроенный Web-интерфейс (Cisco Visual Switch Manager), так и с помощью интерфейса командной строки по Telnet, а также с помощью средств управления CiscoWorks 2000.

В качестве оборудования рабочей группы кроме оборудования Cisco System используется коммутатор компании Planet, Planet FNSW-2401 оснащенный 24 портами 10/100BaseTX с автоопределением скорости портов. Каждый порт MDI/MDI-X с автоопределением может быть использован для соединения с 8-ю компьютерами или каскадирования концентраторов/коммутаторов.

Для подключения используемого оборудования к ВОЛС используются преобразователи среды передачи данных McBasic Gigabit Ethernet Series с интерфейсами 1000BaseTX витая пара и 1000Base-FX многомодовый оптический кабель.

Для организации беспроводного оптического канала связи используется оборудование фирмы НПК «КАТАРСИС» – БОКС 100М-ТС1. Системы БОКС используют в качестве источников излучения только светодиоды. Их излучение, в отличие от лазерного, не наносит здоровью человека никакого вреда и не требует каких-либо аттестационно-согласовательных мер.

Для подключения удаленных пользователей используются модемы Zyxel Prestige 841C и Prestige 841. Prestige 841 – это настольный абонентский VDSL-модем с Ethernet-интерфейсом 10/100BaseT, который может использоваться в связке с головным модемом Prestige 841C (Central Office Equipment).

Практика показывает, что модернизацию локальных вычислительных сетей необходимо проводить приблизительно каждые 3–6 лет. Модернизацию необходимо проводить в свете уже существующих ИТ-решений.

Правильный подход к строительству сетей лежит в закладывании ресурсов для последующего развития, исходя из будущих возможных потребностей. Правильно построенная сеть изначально масштабируема.

Сеть должна быть наращиваемой для подключения, например более мощного сервера, для облегчения обслуживания компьютеров, новых приложений или систем контроля объектов. Все факторы необходимо учитывать, чтобы заложить некий потенциал роста.

Важной причиной для модернизации являются вопросы связанные с информационной безопасностью, как правило, необходимо поддерживать защиту сети на высоком уровне. Постоянно обнаруживается множество «дыр» в системах защиты, появляются более изощрённые методы атак, поэтому появляется более современное, а значит более эффективное оборудование позволяющее минимизировать риски несанкционированного доступа.

В конечном итоге хочется сделать вывод, что к задаче по модернизации сети надо подходить комплексно. Необходимо учитывать всевозможные факторы, часто лежащие в совершенно разных плоскостях, но являющиеся дополнениями друг другу и игнорирование одних может свести на нет эффективность других.

Литература

1. Столлингс, В. Беспроводные линии связи и сети. / Столлингс В.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 640 с.
2. Таненбаум, Э. Компьютерные сети. – 4-е издание/ Э. Таненбаум – СПб.: Питер, 2003 – 992 с.
3. Локальные сети. Модернизация и поиск неисправностей. – СПб.: БХВ – Петербург, 2006 – 640 с.
4. Олифер, В.Г. «Компьютерные сети» / В.Г. Олифер, Н.А. Олифер. – СПб: Издательство «Питер», 2004.

К.А. Богданчук (БНТУ, Минск)

Науч. рук. **Д.А. Степаненко**, канд. тех. наук, доцент

АНАЛИЗ КОНСТРУКЦИЙ ОПТИЧЕСКИХ ГОЛОВОК CD-ПРИВОДОВ И ИХ ПРИМЕНЕНИЕ В ИЗМЕРИТЕЛЬНЫХ СИСТЕМАХ

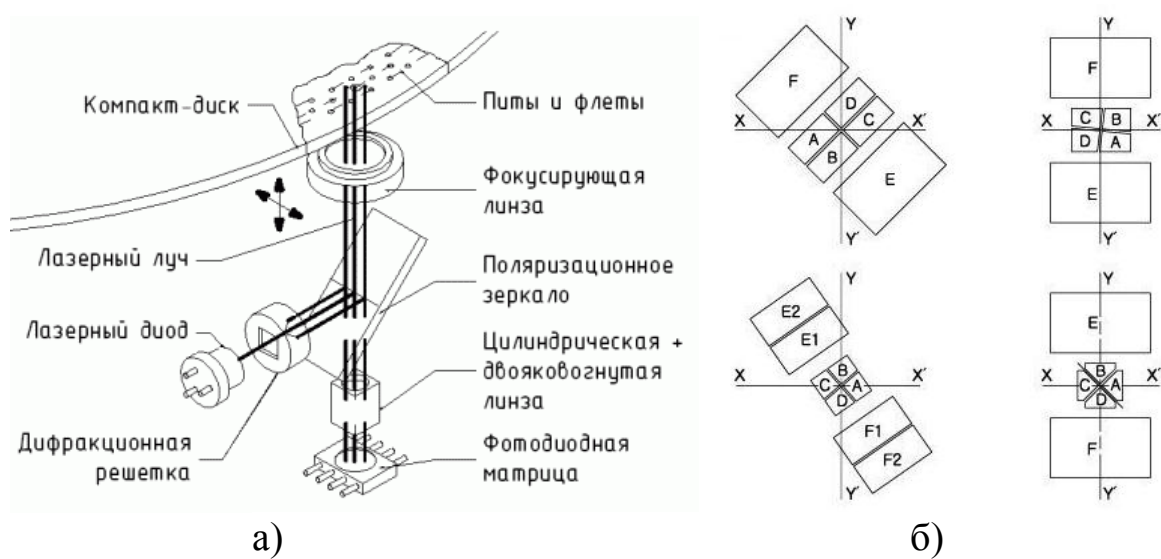
Оптические методы записи и воспроизведения информации с использованием CD-, DVD- и BD-носителей играют важную роль в современной вычислительной и бытовой технике. Аппаратная реализация этих методов основана на использовании оптических головок, представляющих собой прецизионные оптико-механические узлы, позволяющие производить запись и считывание информации с помощью лазерного излучения. В данной работе приводятся сравнительный анализ конструкций оптических головок CD-приводов и анализ

возможности их применения для построения измерительных систем, в частности, компактного виброметра.

Базовым принципом записи информации на CD-, DVD- и BD-носителях является формирование на поверхности диска углублений («питов») с помощью лазерного излучения высокой интенсивности. В CD-приводах для этой цели используется излучение инфракрасного лазерного диода с длиной волны 780 нм, в DVD-приводах – излучение красного лазерного диода с длиной волны 650 нм, а в BD-приводах – излучение синего лазерного диода с длиной волны 405 нм. Уменьшение длины волны позволяет повысить плотность записи информации. Питы используются для представления логических нулей, а плоские участки («лэнды» или «флэты») между питами – для представления логических единиц. Считывание информации производится с помощью лазерного излучения низкой интенсивности от того же лазерного диода, который используется при записи. Для регулировки интенсивности излучения лазерного диода (переключения из режима записи в режим чтения или обратно) оптическая головка снабжается контроллером лазерного диода. Для считывания информации излучение лазерного диода фокусируется на плоской отражающей поверхности диска в пятно диаметром около 1,7 мкм. При попадании излучения на лэнды происходит его зеркальное отражение, а при попадании на питы – диффузное рассеяние. Если зарегистрировать интенсивность отраженного (рассеянного) излучения с помощью фотодетектора, то при попадании луча на лэнды эта интенсивность будет максимальной, а при попадании на питы – минимальной. Таким образом, регистрация интенсивности отраженного (рассеянного) лазерного излучения позволяет считывать информацию, записанную на диске в виде последовательности питов и лэндов. Схема считывания информации с CD-диска приведена на рисунке 1.

Луч лазерного диода с помощью дифракционной решетки делится на 3 части (основной максимум и два побочных максимума 1-го порядка). Луч, соответствующий основному максимуму, после отражения от диска попадает на квадрантный фотодетектор, образованный фотодиодами A-D, а лучи, соответствующие побочным максимумам, – на боковые фотодиоды E и F. Сигнал $TE = E - F$ соответствует ошибке трекинга, характеризующей отклонение считывающего луча от информационной дорожки. Сигналы с фотодиодов квадрантного фотодетектора комбинируются между собой по формулам $RF = A + B + C + D$ и $FE = (A + C) - (B + D)$. Сигнал RF соответствует суммарной интенсивности света, попадающего на фотодетектор при отражении от диска, и несет информацию о последовательности питов и лэндов на

информационной дорожке. Сигнал FE соответствует ошибке фокусировки, характеризующей отклонение фокуса головки от поверхности диска. Неравномерное распределение интенсивности на фотодиодах A-D при отклонении фокуса создается за счет использования астигматической линзы. Головка работает в режиме автофокусировки, для чего сигнал FE подается на исполнительное устройство в виде электромагнитного привода, перемещающего фокусирующую линзу по координате z. Автофокусировка позволяет избежать негативных эффектов, связанных с торцовым биением поверхности диска. Отслеживание информационной дорожки производится с помощью второго привода, на который подается сигнал TEи который перемещает линзу по координате y.



а) ход лучей в оптической головке;
 б) схемы расположения фотодетекторов

Рисунок 1 – Схема считывания информации с CD-диска

Благодаря простоте конструкции, компактности и низкой стоимости оптических головок и широкому распространению лазерных измерительных систем имеется большое количество работ, посвященных применению оптических головок для построения различных измерительных приборов, например, профилометров и акселерометров. Учитывая, что зависимость сигнала FE от смещения фокуса по координате z является линейной в диапазоне шириной 10-20 мкм, оптические головки могут быть использованы для построения компактных виброметров для измерения малых колебательных смещений, например, амплитуды колебаний ультразвуковых волноводных систем. Рабочая частота фотодетекторов CD-привода со скоростью чтения X40 составляет около 55 МГц, что потенциально позволяет исследовать высокочастотные колебательные процессы.

Для выбора оптической головки, наиболее подходящей для построения виброметра, был проведен сравнительный анализ конструкций трех типов головок: KRS-202A (Sony), KSS-572A (Sony) и SF-P151 (Sanyo). В головке KRS-202A для переключения между режимами чтения и записи используется контроллер лазерного диода, что усложняет электронную схему головки. Так как в виброметре достаточно использовать лазерное излучение низкой интенсивности, то предпочтительным является применение головок, предназначенных для работы только в режиме чтения и не имеющих контроллера лазерного диода. Этим условиям удовлетворяют головки KSS-572A и SF-P151. Простота электронной схемы головки имеет большое значение по той причине, что многие используемые в головках интегральные микросхемы не имеют подробного технического описания. Дополнительным достоинством головок, работающих только в режиме чтения, является снижение риска повреждения зрения излучением лазерного диода. В качестве фотодетекторов в головках KRS-202A и SF-P151 используются PDIC-модули (PDIC = PhotoDetectorIntegratedCircuit), содержащие интегрированные операционные усилители сигналов с фотодиодов, что позволяет исключить необходимость использования внешних усилителей. В оптико-электронной схеме головки KSS-572A используются всего два элемента: фокусирующая линза и LDH-модуль (LDH = Laser-Detector-Hologram). LDH-модуль представляет собой оптико-электронную микросхему, содержащую лазерный диод, зеркало, фотодетекторы и голографический оптический элемент (дифракционную решетку). В головке SF-P151 используются отдельно установленные лазерный диод и PDIC-модуль. Окончательно в качестве наиболее подходящей для построения виброметра была выбрана головка SF-P151, имеющая наиболее простую электрическую схему включения.

М.Ю. Бокий (УО «ГГТУ им. П.О. Сухого», Гомель)

Науч. рук. **Е.Г. Стародубцев**, канд. физ.-мат. наук, доцент

АЛГОРИТМ ОТСЛЕЖИВАНИЯ ИЗМЕНЕНИЙ ДАННЫХ В УНИВЕРСАЛЬНОМ ХРАНИЛИЩЕ ФИНАНСОВОЙ ОТЧЕТНОСТИ

Пользователи корпоративного хранилища данных нуждаются в отслеживании изменений значений некоторых атрибутов. Разработчик универсального хранилища должен реализовать возможность ведения истории для таких атрибутов. Такая функциональность предоставляется интерфейсами консолидации финансовой отчетности.

Сохранение истории данных выполняется путем создания нескольких записей для одного натурального ключа, разделяя старые и новые записи с помощью суррогатных ключей, ключей версий и наборов колонок аудита.

Для реализации процесса перезаписи одних атрибутов и сохранения истории изменения других необходимо сгенерировать хэш-код на основании определенного набора атрибутов, который затем может быть использован для идентификации обновленных колонок.

Набор колонок аудита будет использован для отслеживания сделанных изменений (см. таблицу).

Таблица – Набор колонок для реализации процесса

Тип колонки		Комментарий
Surrogate Key		Суррогатный ключ
Version Key		Значение возрастает на единицу с каждым обновлением, является частью первичного ключа таблицы.
Natural Key		Натуральный ключ
NonHistory		Атрибуты, для которых не нужно сохранять историю
History		Атрибуты, для которых нужно сохранять историю
Колонки аудита	Current Indicator	Текущий индикатор. Если это текущая версия записи - 'Y' иначе 'N'
	Source System Name	Название системы-источника, из которой данные поступили
	Record Checksum Code	Хэш-код на основе натурального ключа и History атрибутов
	Row Status Code	Статус записи: вставка = 'I', обновление = 'U', удаление = 'D'
Колонки аудита	Insert Timestamp	Дата\время (timestamp) вставки записи
	Update Timestamp	Timestamp обновления записи
	Effective Date	Действующая дата: текущая дата при вставке записи.
	Expired Date	Дата истечения. Значение по умолчанию – '31-12-9999'. При вставке новой версии для старой устанавливается текущая дата.

Алгоритм процесса представлен на рисунке 1.

Значения колонок аудита для каждого действия:

1 Вставка записи в таблицу:

Current Indicator = 'Y'

Row Status Code = 'I'

Checksum Code = сгенерированный хэш-код

Insert Timestamp = текущий timestamp

Update Timestamp = текущий timestamp

Effective Date = текущая дата

ExpiredDate = '31-12-9999'

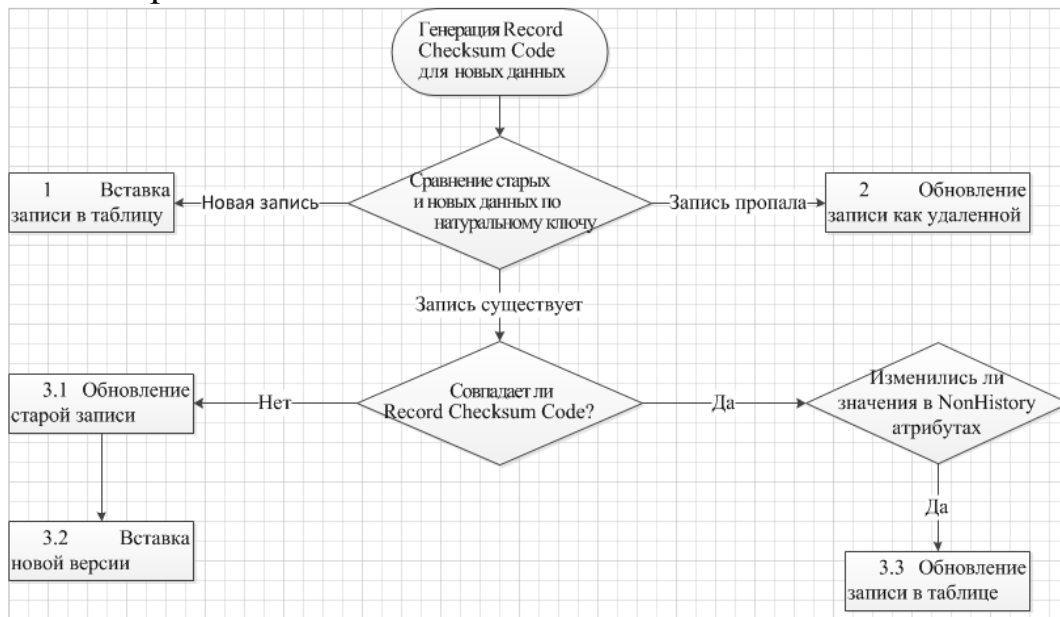


Рисунок 1 – Алгоритм отслеживания изменений данных

2 Обновление записи как удаленной:

Current Indicator = 'N'

Row Status Code = 'D'

Update Timestamp = текущий timestamp

ExpiredDate = текущая дата

3.1 Обновление старой записи:

Current Indicator = 'N'

Row Status Code = 'U'

Update Timestamp = текущий timestamp

ExpiredDate = текущая дата

3.2 Вставка новой версии:

Version key = version key старой записи + 1

Current Indicator = 'Y'

Row Status Code = 'I'

Checksum Code = сгенерированный checksum код

Update Timestamp = текущий timestamp

Insert Timestamp = текущий timestamp

Effective Date = текущая дата

ExpiredDate = '31-12-9999'

3.3 Обновление записи в таблице:

Применить изменения для обновления записи

Row Staus Code = 'U'

Update Timestamp = Текущий timestamp

Таким образом, предлагаемый алгоритм позволяет сохранять историю изменений значений атрибутов там, где это важно для аналитика, и в то же время не тратить ресурсы и не создавать излишних, ненужных данных там, где это не требуется. Использование контрольной суммы ускорит работу приложения, так как для каждой записи достаточно сравнивать только ее вместо множества атрибутов, требующих ведения истории. Набор колонок аудита предоставит аналитику исчерпывающую информацию, а также может быть использован для фильтрации данных различными интерфейсами.

И.А. Борихин (УО БрГТУ, Брест)

Науч. рук. **В.И. Хвещук**, канд. тех. наук, доцент

О МОДЕЛИРОВАНИИ ДЕЯТЕЛЬНОСТИ ИТ-СПЕЦИАЛИСТОВ

Одной из важных особенностей современных ИТ- изделий является то, что их производство носит единичный характер. Это требует от производителей каждый раз по-новому организовывать производственный цикл (подбор исполнителей, разработка модели производства, выбор средств для реализации и другие компоненты). В данной работе предложен модельный подход к организации планирования производства ИТ- изделий, который основан на использовании совокупности следующих моделей: модель ИТ- изделия; модель проекта; модель производства; модель ИТ – специалиста. Основные положения данного подхода следующие:

- Модель ИТ- изделия. Предложена типовая системная архитектура изделий. Она основана на классификации следующих компонентов: предметных областей для разработки изделий, видов разработки, архитектур изделий, элементов изделий.

- Модель ИТ- производства. Она представляет собой набор моделей технических процессов, необходимых для реализации заданного класса ИТ- изделий. Для каждого процесса определяется примерный набор работ и задач, а также спектр возможных средств для их реализации.

▪ Модель ИТ-специалиста. Предложена субъективная оценочная модель навыков, способностей и знаний сотрудника. Она определяется на основе резюме, собеседования, полученного образования, накопленного опыта, психологического тестирования и т.д., которые представляются в виде определенных значений коэффициентов.

▪ Модель проекта изделия. Она определяется в виде совокупности следующих компонентов: вида проекта, назначения проекта, сроков разработки, требуемых ресурсов и других компонентов.

В настоящее время разработан программный комплекс (ПК), состоящий из базы данных (БД) для хранения рассмотренных моделей, а также набор приложений, обеспечивающих работу с ними. Ведется накопления реальных данных в БД и адаптация предложенных моделей к реальным условиям производства ИТ-изделий.

На основе предложенного подхода и созданного ПК разрабатывается система поддержки принятия решений при управлении ИТ-проектами, которая позволит уменьшить следующие риски: при открытии новых ИТ-проектов, при подборе сотрудников для их реализации, при планировании производства ИТ-изделий и другие.

О.А. Васильев (УО «ГГТУ им. П.О. Сухого», Гомель)
Науч. рук. **Е.Г. Стародубцев**, канд. физ.-мат. наук, доцент

АНАЛИЗ РЕЗУЛЬТАТОВ ТЕСТИРОВАНИЯ С ПРИМЕНЕНИЕМ ГРАФОВ

Автоматизация анализа результатов тестирования позволяет минимизировать временные затраты на анализ проблем. Автоматизацию можно ввести путем разработки унифицированного формата тестовых сообщений и ошибок, к которым применяются средства анализа. В качестве основного средства для анализа ошибок тестов предлагается использовать систему, построенную на основе графов. Используется взвешенный граф [1], множество вершин которого представлено тестами, с имеющимися стеками вызовов и сообщениями об ошибках, данный граф не может иметь циклов.

Алгоритм можно описать следующим образом. В начале цикла имеется нуль-граф, для начала анализа выбирается произвольная вершина, однако предпочтительно выбирать первую вершину, т.к. для хранения графа используется списковая структура. Начиная с выбранной вершины, идет попарное сравнение с остальными вершинами графа по определенным правилам, которые определяют степень схожести проблем.

Функция, выполняющая роль правила, возвращает числовое значение, описывающее степень схожести проблем. При удовлетворении минимального барьера схожести между вершинами графа создается дуга с весом, равным значению вычисленной функции. Все найденные схожие вершины маркируются, и происходит переход к следующей немаркированной вершине. Цикл повторяется, пока все вершины не будут промаркированы.

На выходе будет получен граф, который представляет собой множество подграфов, отражающих найденные проблемы. Алгоритм предлагает высокую скорость обработки больших объемов данных, т.к. при каждой итерации скорость будет возрастать с исключением вершин. Граф может использоваться как для получения групп проблем по тестам, так и для дальнейшей визуализации, оценки точности сравнения и дальнейшей корректировки правил.

Литература

1. Graph (mathematics) [Электронный ресурс] / Wikipedia. – 2005. – Режим доступа: [http://en.wikipedia.org/wiki/Graph_\(mathematics\)](http://en.wikipedia.org/wiki/Graph_(mathematics)). – Дата доступа: 6.03.2014.

О.А. Васильев (УО «ГГТУ им. П.О. Сухого», Гомель)
Науч. рук. **Е.Г. Стародубцев**, канд. физ.-мат. наук, доцент

ВИЗУАЛИЗАЦИЯ ПРОЦЕССА ВЫПОЛНЕНИЯ АВТОМАТИЧЕСКИХ ПАРАЛЛЕЛЬНЫХ ТЕСТОВ

Распараллеливание автоматических тестов имеет как преимущества (увеличение быстродействия, возможность тестирования системы на единовременный доступ), так и недостатки (сложность поддержки подобных тестов и исправления ошибок). Один из способов решения данной проблемы – графическая визуализация программных потоков. В качестве удобного средства отображения параллельного выполнения тестов можно использовать график с временной линией. Вдоль ординат такого графика размещается временная линия с разметкой по часам и минутам; вдоль оси абсцисс – номера потоков, в которых запускаются тесты; на плоскости графика – названия тестов с отметками времени начала и конца.

В качестве платформы для реализации был выбран язык JavaScript (уже существуют реализации похожих графиков, но большинство из них поставляется в виде платных программных комплексов либо имеет

ограниченный функционал и сложны в использовании). Был разработан формат сообщения для передачи данных от сервера клиенту. Данные передаются в JSON, что обеспечивает легковесность данных и удобство обработки на клиенте. Сам формат сообщения содержит имя, а также время начала и конца теста (используются временные отметки UNIX в UTC для точности и простоты манипуляций). Использовалась бесплатная библиотека D3JS с широким набором средств, для создания интерактивных HTML-документов, включая средства для работы с SVG-графикой [1]. D3JS упрощает взаимодействие с SVG-примитивами, предлагая разработчику декларативный синтаксис, что позволяет быстро создавать сложные графики. Сам код графика представляет собой небольшой модуль на JavaScript, легко подключаемый на страницу, в который лишь надо передать данные в описанном выше формате. На основе данных будет построен интерактивный график, наглядно показывающий запуски параллельных тестов по времени и позволяющий быстро определить проблемные тесты и правильно их сгруппировать в дальнейшем.

Литература

1. Data-Driven Documents [Электронный ресурс] / D3JS. – 2013. – Режим доступа: <http://d3js.org/>. – Дата доступа: 19.02.2014.

А.Н. Веракшич (УО «ГГУ имени Ф. Скорины», Гомель)

Науч. рук. **М.А. Подалов**, ассистент

ИСПОЛЬЗОВАНИЯ МАШИННОГО ЗРЕНИЯ В УЧЕБНОМ ПРОЦЕССЕ

Основную часть информации о внешнем мире человек получает по зрительному каналу и далее весьма эффективно обрабатывает полученную информацию при помощи аппарата анализа и интерпретации визуальной информации. Поэтому встает вопрос о возможности машинной реализации данного процесса. [1]

За счет возрастания сложности решаемых научно-технических задач, автоматическая обработка и анализ визуальной информации становятся все более актуальными вопросами. Данные технологии используются в весьма востребованных областях науки и техники, таких как автоматизация процессов, повышение производительности, повышение качества выпускаемых изделий, контроль производственного оборудования, интеллектуальные робототехнические комплексы, системы управления движущимися аппаратами, биомедицинские исследования и множество других. [2]

Цель машинного зрения в распознавании положения – определение пространственного местоположения (местоположения объекта относительно внешней системы координат) или статического положения объекта (в каком положении находится объект относительно системы координат с началом отсчета в пределах самого объекта) и передача информации о положении и ориентации объекта в систему управления или контроллер.

Примером использования распознавания положения может служить физические опыты, когда наблюдатель используя машинное зрение может получить точные данные измерений, такие как: траектория движение наблюдаемого тела, определение количества колебаний, локализация центра тяжести тела и другие.[3]

Интерес к системам машинного зрения во многом поддерживается успехами машин в тех областях, где они намного превосходят человека по своим возможностям. Так, компьютер способен выявлять в изображении тысячи градаций серого и различать миллионы цветов, очень быстро решать типовые и хорошо формализуемые задачи распознавания и определять мелкие детали изображений.

В работе с помощью машинного зрения было исследовано движение физического маятника, а также определение ускорения свободного падения с помощью физического маятника. Задача заключалась в том, чтобы отследить количество колебаний маятника с помощью мобильного устройства, а также составить график отклонений и вывести все данные на экран.

Литература

3. Форсайт, Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. – М. : ИД «Вильямс», 2004. – 245 с.
4. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман. – М. : БИНОМ. Лаборатория знаний, 2006. – 116 с.
5. Hashimi, S.Y. Pro Android / Sayed Y.Hashimi, Satya Komatineni – М.: Издательство «Apress», 2009. – 437 с.

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

Науч. рук. **А.В. Воруев**, канд. тех. наук, доцент

ПРОБЛЕМЫ РАСПРЕДЕЛЕНИЯ IP-АДРЕСОВ В СЛОЖНЫХ СЕТЯХ

Для полноценного обеспечения работы любой сети необходимо разделять адресное пространство между всеми участниками сети. IP-адрес

(Internet Protocol Address) – уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP в сети. Благодаря IP-адресу информация доставляется нужному устройству в сети. Также адресация необходима для связи небольших сетей в более крупную.

Разумеется, возникает вопрос, как распределить IP-адреса между компьютерами и другими устройствами. IP-адрес можно назначить вручную каждому компьютеру. Это делается администратором. У этого способа есть ряд недостатков. Например, нужно отслеживать, чтобы в пределах одной сети IP-адреса не повторялись. Или если в сети слишком много устройств, настройка IP-адреса на каждом из них – достаточно трудоемкий процесс.

Второй способ – использовать одно из устройств в качестве так называемого «раздатчика» адресов. Такое устройство будет играть роль DHCP-сервера. Каждое устройство сети будет обращаться к серверу и получать от него динамический IP-адрес. Для следующей организации сети необходимо, чтобы в каждой подсети был свой DHCP-сервер, раздающий IP-адреса определенного диапазона. Это создаёт дополнительные проблемы. Так, каждый такой сервер необходимо настроить, администрировать, поддерживать в рабочем состоянии. Опять-таки, это тоже трудоемкий процесс.

Есть возможность настроить один DHCP-сервер, раздающий для нескольких сетей IP-адреса. При этом сервер располагается в одной из сетей. Рассмотрим этот вариант подробнее.

Для работы использовалась программа Cisco Packet Tracer. Была создана сетевая структура, состоящая из 17 устройств (рисунок 1). Для того чтобы распределить IP-адреса, необходимо настроить сервер DHCP, который сам будет присваивать эти адреса компьютерам сети. Основной сложностью поставленной цели является то, что сеть содержит в себе еще и подсети, компьютеры которых также должны получать адреса, а сервер для этого должен быть один и располагаться в одной подсети. Как видно из рисунка, DHCP-сервер располагается в сети роутера 0. Далее пошагово описана настройка сервера.

Шаг 1. Настроить получение IP-адреса компьютерами через DHCP. Для этого выбрать компьютер, выбрать порт и поставить переключатель на DHCP.

Шаг 2. Узнать IP-адреса всех роутеров сети.

Шаг 3. Настроить сервер на раздачу IP-адресов. Для этого войти в настройки DHCP сервера, затем прописать следующее (пример для сети 192.168.1.0): в строке Pool Name указать имя пула serverPool; в строке Default Gateway указать шлюз по умолчанию 192.168.1.1; указать начальный IP-адрес 192.168.1.3 т.к. предыдущие адреса – это адреса самого сервера и роутера. Нажать кнопку Add.

Шаг 4. Прodelать то же самое для остальных сетей, изменяя имя пула, начальные адреса и шлюзы по умолчанию (для этого и нужны были IP-адреса роутеров, ведь они являются шлюзами).

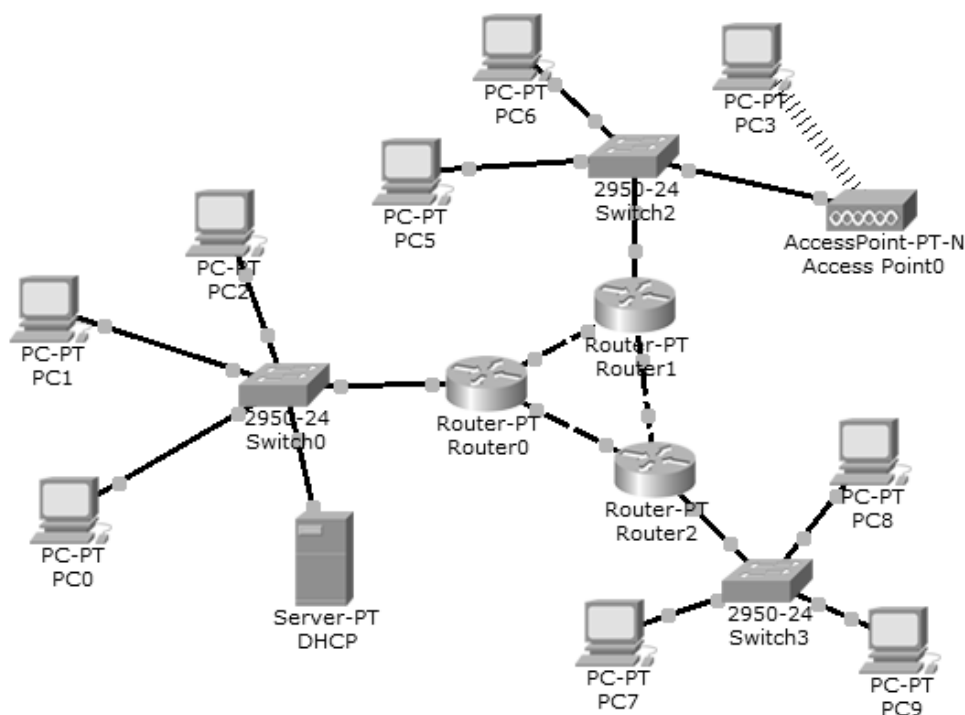


Рисунок 1 – Сложная сеть с одним DHCP-сервером

Приведенная выше сеть достаточно проста, не имеет лишних серверов. Все устройства могут обмениваться информацией.

С.В. Ворочай (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **В.Н. Леванцов**, ст. преподаватель

ЗАДАЧИ ВВЕДЕНИЯ НА ПРЕДПРИЯТИИ ДОПОЛНИТЕЛЬНЫХ РЫНКОВ СБЫТА ПРИ ПОМОЩИ ВЫСОКИХ ТЕХНОЛОГИЙ В ИНТЕРНЕТ

В современном мире все больше и больше торговых компаний не останавливаются на традиционном способе реализации своей продукции, а прибегают к торговле посредством различных интернет-магазинов и аукционов, в том числе и своих собственных. Интернет-магазин – это не магазин в привычном для нас сейчас понимании этого слова. Это Функция, которая окружает человека буквально везде, где есть доступ в Интернет. Перечислим основные преимущества интернет-магазина:

– Доступность в режиме 24/7: круглосуточно и семь дней в неделю – это способствует лучшей реализации товаров и услуг. Покупатель сам в праве выбирать, когда ему удобно заказать товар или услугу.

– Не нужна аренда больших производственных площадей и складов – достаточно небольшого офиса для обслуживания клиентов. Без дополнительных затрат на аренду помещений предприятие имеет возможность ставить цену на товар ниже, чем у конкурентов, но иметь эквивалентную прибыль с продаж.

– Интернет-магазин может быть как основным источником прибыли, так и дополнительным по отношению к основному бизнесу – это важный ответ на вопрос «Зачем нужен интернет-магазин?».

– Нет региональных ограничений – можно находить потенциальных покупателей как в своем городе или области, так и по всей стране. В последствии вполне можно расширить рынок сбыта от международного до мирового.

– Такие преимущества интернет-магазина как экономия времени и свобода выбора, играют важную роль и для покупателей.

– Нет никаких ограничений по количеству представленных на виртуальной витрине товаров. Мы можем показывать покупателям не только то, что имеется непосредственно на складе в данный момент времени, но и то, что предприятие еще не имеет в наличии, но имеет возможность достать данный товар в кратчайшие сроки. Т.е. мы получим экономию внутренних средств на товарах, которые не пользуются большим спросом, не покупая их на склад.

– Торговля и сам бизнес в целом в Интернете не требует большого количества обслуживающего персонала – можно обойтись одним консультантом, который сможет быстро реагировать на поступающие заявки от покупателей. В физическом магазине, как правило, потребность в персонале намного выше, а значит, большая часть дохода с продаж уходит на оплату труда работникам.

– В случае с раскруткой и продвижением интернет-магазина можно сфокусироваться только на потенциально заинтересованных в ваших товарах или услугах пользователях.

Интернет-магазин в первую очередь полезен владельцам стандартизированных товаров и услуг. Например, продавать с виртуальных полок сотовые телефоны и книги значительно проще, чем, скажем, одежду или обувь, которые требуется примерять. Другими словами, потенциальные владельцы онлайн-магазинов – это те компании, которые теоретически способны продавать свои товары по каким-то определенным каталогам, а так же с организацией доставки.

В представленной работе рассматривается разработка интернет-магазина для организации, которая занимается продажами в сфере IT технологий.

Мы раскрыли возможности интернет-магазинов для различных предприятий, но второй по важности вопрос состоит в том, чтобы правильно создать концепцию, структуру и наполнения сайта, чтобы он мог быть конкурентоспособным и привлекать целевых посетителей.

Во-первых, стоит обратить пристальное внимание на форму поисков товаров. Сделайте данный процесс выбора и покупки товара максимально удобным и комфортным для посетителей. Форма поиска должна располагаться на главной странице и быть максимально заметной, понятной и удобной.

Во-вторых, навигация самого сайта должна быть интуитивно понятной и функциональной. Меню сайта должно быть удобным и рассчитано конкретно на целевую аудиторию. Нестандартная панель навигации может отпугнуть потенциального покупателя.

В-третьих, дополнительные излишества, такие как музыкальное сопровождение, большие баннерные рекламы, сайт постоянно выдает ошибки или перекидывает на сторонние страницы и многое другое – это все то, что отпугнет покупателя.

В-четвертых, сайт должен быть максимально быстрым при загрузке. Высокоскоростной доступ в Интернет, предоставляемый современными провайдерами – это обычная ситуация на рынке. Но в этом есть свои минусы для разработчиков сайтов – разбалованный пользователь. Ведь нагрузка на сервер, как правило, большая – поэтому лишние запросы в БД на получение интересующих нас товаров в больших количествах только ухудшат скорость работы сервера. Исследования показали, что пользователь не испытывает неудобств если сайт грузиться за половину секунды, но вот большее время загрузки сайта начинает нервировать со временем даже самого отчаянного покупателя. Поэтому следует учесть этот факт и делать, с одной стороны, «легкие» страницы, а с другой стороны, максимально заполненные информацией. Лучше дать возможность пользователю открыть самостоятельно большую фотографию товара в отдельном фрейме, чем грузить лишнюю информацию, увеличивая при этом скорость загрузки сайта.

Для того чтобы потенциальный покупатель действительно заинтересовался товаром, необходимо разместить на сайте качественные фотографии этого товара. В интернет-магазине покупатель может отложить в свою корзину только изображение, поэтому сильно мутное или расплывчатое фото его только отпугнет. Лучше всего выкладывать

несколько качественных фотографий одного и того же товара, только в разных ракурсах.

Наличие новостной ленты, форума и возможности добавления комментариев к товару на сайте говорит о его надежности, честности и конкурентоспособности. Потенциальному клиенту будет интересно прочитать отзывы людей, которые уже делали заказ в этом интернет-магазине и остались довольны покупкой. Также на форуме можно ознакомиться с товаром более детально, не просто почитать подробное описание товара и его техническими характеристиками, а мнение людей, купивших его. Либо обсудить интересующие вопросы с администрацией. Также пользователь может выразить желание что-нибудь изменить или усовершенствовать на сайте. Внимательно прочтите критику и пожелания, а затем сделайте соответствующие выводы. Вы ведь создавали сайт именно для посетителей, поэтому должны делать все, чтобы им на сайте было легко, уютно и комфортно.

Так же на сайте должен быть уникальный дизайн, который соответствует логотипу или торговой марки фирмы, чей товар вы распространяете на сайте. Так же не стоит забывать о информации о заводах-изготовителях – зачастую прямая ссылка на официальный сайт завода сократит время пользователя на поиски нужного товара по официальным источникам и оградит попадания его на конкурентные сайты в процессе поиска.

Для того чтобы продвинуть интернет-магазина в сети Интернет есть много способов это сделать, но первое, что надо сделать – это разместить информацию о вашем сайте в поисковых машинах и каталогах Интернета. Самые популярные поисковые системы на нашем рынке – это www.google.com, www.yandex.ru, www.rambler.ru, www.tut.by. Таким образом вы даете возможность любому пользователю Интернета найти ваш сайт по ссылкам или запросам в поисковых машинах. После того, как вы разместите информацию о своем сайте в поисковых машинах, вам нужно будет регулярно обновлять свой сайт, добавлять новую информацию – новости, анонсы, статьи и другую информацию. Регулярность подразумевает, что, как минимум, раз в неделю у вас появляется новая информация на сайте.

Выполнение данных рекомендаций поможет сделать красивый, удобный и конкурентоспособный интернет-магазин, который в свою очередь принесет прибыль предприятию.

Д.В. Гетиков (УО «ГГУ им.Ф. Скорины», Гомель)
Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА WEB-ОРИЕНТИРОВАННОЙ БАЗЫ ДАННЫХ В КОРПОРАТИВНОЙ СРЕДЕ LOTUSNOTES

Современное производство немислимо без управляющих систем разной степени сложности. Но любой управляющей системе необходимо соответствующее информационное и программное обеспечение, иначе она не сможет продуктивно работать.

Выбор LotusNotes для разработки приложения обуславливается рядом причин. Разработчики смогут оценить улучшенные возможности интеграции Domino и Notes с Web-клиентами. Новая версия LotusNotes пополнилась несколькими новыми мощными программными объектами, среди которых страницы, структуры и наборы фреймов. Переориентация приложения LotusNotes под работу в Web – также довольно распространенное явление.

DominoDesigner позволяет работать со всеми основными технологиями, используемыми для верстки Web-документов, такими как HTML, CSS, JavaScript и т.д. Также присутствует возможность реализации серверной стороны обработки документов (при помощи Java и PHP), что позволяет создавать динамические интернет-страницы.

Создание пользовательского интерфейса обусловлено рядом используемых технологий: HTML – как многофункциональный язык разметки, очень хорошо подходит для создания простого и понятного конечному пользователю интерфейса, при этом поддерживается использование каскадных таблиц стилей, что позволяет гибко и быстро настраивать стили отображения документа. Так же, поскольку мы работаем с HTML-документом, есть возможность представления и обработки его как дерева элементов.

Благодаря богатым возможностям среды разработки Notes, можно быстро создать систему прототипов для конкретного заказчика, которая могла бы обеспечить следующее:

- Облегчение создания баз данных на основе эскизов
- Возможность устанавливать порядок, согласно которому базы данных применяют шаблон
- Возможность легко добавлять усовершенствованные атрибуты, которые позволяют расширять возможности уровня-поля
- Облегчение добавления динамики формы/поля с помощью логики сценариев
- Возможность наследования изменений в шаблонах и др.

При наличии большого количества заказчиков или же исполнении заказов, которые лежат в идентичных сферах бизнеса, может быть привлечено дополнительное преимущество работы с Notes. Это преимущество состоит в удобстве клонирования базы данных и форм.

При помощи данной технологии была решена задача по реализации web-ориентированной базы данных «Грузоперевозки» средствами универсального корпоративного средства управления данными LotusNotes при использовании основных технологий проектирования и верстки web-документов в специальной среде разработки – DominoDesigner.

Н.А. Глухов (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

РЕШЕНИЕ ОПТИМИЗАЦИОННЫХ ЗАДАЧ

Задача линейного программирования формулируется следующим образом: задана целевая линейная функция независимых переменных x_1, x_2, \dots, x_n : $a_1x_1 + a_2x_2 + \dots + a_nx_n$, где a_i – коэффициенты линейной функции, $i=1, 2, \dots, n$. Известно также несколько уравнений и неравенств (ограничений), независимыми переменными которых являются x_1, x_2, \dots, x_n :

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b(a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b)$$

$x_i \geq 0, i=1, 2, \dots, n$ и ряд других.

Среди множества вариантов, удовлетворяющих условиям ограничения, найти совокупность переменных x_1, x_2, \dots, x_n , при которых целевая функция получает максимальное (минимальное) значение.

Система Mathematica имеет встроенные функции решения математического программирования. Функция $ConstrainedMax[f, \{Q\}, \{x_1, x_2, \dots, x_n\}]$ ищет глобальный максимум, т.е. такое значение x_i , при которых выполняются все ограничения Q , а целевая функция имеет максимальное значение. Функция $ConstrainedMin[f, \{Q\}, \{x_1, x_2, \dots, x_n\}]$ ищет глобальный минимум.

Технология отыскания оптимального значения реализована на конкретных типовых примерах из техники, управления, экономики.

1. Задача о назначениях. Пять человек с номерами a_1, a_2, a_3, a_4, a_5 способны выполнить пять заданий t_1, t_2, t_3, t_4, t_5 . В силу разной квалификации на выполнение этих заданий им потребуется различное время. Как следует распределить людей по заданиям, чтобы минимизировать время выполнения? Время выполнения (в часах) приведено.

2. Транспортная задача. Имеется четыре склада товаров и три их потребителя. Известны также количество товаров на каждом складе, потребности каждого потребителя, а также стоимость доставки товара до потребителя. Необходимо составить оптимальный план перевозок, при котором суммарная стоимость перевозок будет минимальной.

3. Задача о профессиях. Пусть в некоторой организации берут на работу трех человек: Алексея, Петра и Николая. Им бала предложена серия тестов для определения их способностей к профессиям учителя, программиста и администратора. Полученные баллы известны. Отдел кадров должен решить, кого, куда направить на работу с максимальной пользой для организации.

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

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

BLINK: НОВЫЙ БРАУЗЕРНЫЙ ДВИЖОК ДЛЯ CHROMIUM

Blink – кросс-платформенный движок веб-браузеров, созданный сообществом Chromium и разработчиками из корпорации Google в частности. Blink стал ответвлением предыдущего Chromium-движка WebKit, созданного компанией Apple еще в 1998 году и переработанного для браузеров Chromium в 2008-2012 годах. Проект был объявлен Google в апреле 2013 года, и уже в мае вышел браузер Google Chrome 28 на основе нового движка. Сначала интернет-сообщество отреагировало на Blink скептически, а некоторая его часть – с долей возмущения.

Однако теперь множество причин заставляет думать, что создание Blink стало удачным решением:

1. До создания нового движка Google занимал первое место по количеству доработок и правок кода WebKit. Теперь почти все программисты Google оставили работу над этим движком и переключились на проект Blink, а это значит, что сообщество потеряло большое количество разработчиков. Так как новый движок основан на WebKit, часть введенных функций будет осуществима на обоих движках, однако со временем часть общего кода будет становиться все меньше и меньше. Потеря многочисленной команды кодеров существенно снизила темп разработки WebKit. Это не значит, что он не сможет держаться на одном уровне с Blink, но это будет сложно. Такое замедление в развитии может привести к отставанию браузера Safari (который тоже основан на движке WebKit) для MacOS и iOS от других браузеров.

2. Кроме очевидных вопросов о развитии движка, появление Blink может изменить способ создания веб-сайтов. Из-за быстрого взлета Chrome, iOS и Android, WebKit стал предпочтителен для многих веб-разработчиков. В результате значительная их часть при создании сайтов перестала беспокоиться о пользователях других браузеров (на основе движков Presto, Gecko и других). Это стало большой проблемой, поскольку WebKit имеет специфический CSS-фильтр. Разработка сайтов конкретно под WebKit привела бы к неверному отображению страниц на других браузерах. С развитием Blink сайты должны будут быть максимально совместимы с основными стандартами, поэтому описанная выше проблема перестанет существовать.

3. До появления Blink кроме WebKit наиболее популярны были три движка: Presto (Opera), Gecko (MozillaFirefox) и Trident (InternetExplorer). Работа над Presto была официально остановлена в 2012 году, а значит, Blink стал ему мощной заменой в браузерной конкуренции. Наличие стольких конкурирующих движков должно привести к активному развитию каждого из них, что не может не радовать интернет-пользователей (хотя для разработчиков сайтов может оказаться лишней проблемой).

Главной причиной, по которой Google решил разработать собственный движок, стали архитектурные различия Chromium и WebKit. В Chromium была реализована поддержка мультипроцессорности. Когда в WebKit появилась альтернативная реализация, Google продолжили использовать свой код, что существенно усложнило интеграцию WebKit и Chromium. С каждой новой версией движка разработчикам Chrome приходилось все больше и больше кода переделывать под себя. Теперь эта проблема исчезла. С переходом на Blink проект очистили от лишних файлов, а код был сокращен на 8,8 миллионов строк (вместо планируемых 4,5 миллионов).

С учетом добавленных функций, Chromium (а следовательно и GoogleChrome) стал легче, быстрее и безопаснее. Кроме того, на сегодняшний день число разработчиков Blink гораздо больше, чем у какого-либо другого движка. Как и ранее с WebKit, большинство этих разработчиков работает на Google, однако с переходом Opera и Яндекс.Браузер на Blink (с Presto и WebKit соответственно) к проекту примкнуло еще большее количество кодеров. Для обеспечения продукту полной безопасности Google выплачивает тысячи долларов отдельным разработчикам, которые находят уязвимости в коде. Таким образом, Blink приобретает все большую популярность, а вместе с ней – и большие перспективы для развития.

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

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

ИСПОЛЬЗОВАНИЕ ЯЗЫКА PYTHON ДЛЯ СОЗДАНИЯ КОМПЬЮТЕРНЫХ ИГР

Python – широко используемый высокоуровневый язык программирования общего назначения. Его принципы проектирования подчеркивают удобочитаемость кода, и его синтаксис позволяет программистам выражать понятия в меньшем количестве строк кода, чем было бы возможно на таких языках, как C/C++ и Java. За простоту и функциональность этот язык был оценен такими компаниями, как Google, IBM, Intel, Cisco, HP, Seagate. Даже NASA использует Python для некоторых математических вычислений. Но для обычных пользователей более привлекательно звучит такая возможность языка, как «Создание игр».

Реализацией этой возможности занимается инструмент языка PyGame. Он добавляет множество возможностей кроме основных функций SDL (SimpleDirectMediaLayer) – мультимедийной библиотеки, реализующей единый программный интерфейс для графической подсистемы, звуковых устройств и средств ввода. Это позволяет создавать полнофункциональные игры и мультимедийные программы. PyGame поддерживается почти каждой платформой и операционной системой.

Перечислим достоинства этого инструмента разработки. Первым в этом списке записано шутовское достоинство «Глупость встроена». Эту функцию разработчики описали так: «PyGame предназначается для того, чтобы сделать разработку ПО забавной. Каждые 3,1415 секунд добавляется новая глупость». Остальные достоинства перечислены ниже:

1 Не требует OpenGL. Эта библиотека порой повреждается при установке на системах Windows и Linux, поэтому требование конкретно OpenGL может сделать процесс разработки медленным или невозможным для некоторых пользователей. PyGame имеет поддержку DirectX, WINDIB, X11, кадрового буфера Linux и других различных бэкэндов.

2 Поддержка многоядерных процессоров. Использование двухъядерных и восьмиядерных процессоров позволяет реализовать гораздо больше игровых функций с помощью GIL (способ синхронизации потоков, используемый в Python).

3 Использование оптимизированного C-кода и ассемблерного кода для реализации базовых функций. Функции, написанные на языке C, часто отрабатывают в 10-20 раз быстрее аналогичных на Python. Ассемблерных код увеличивает это число еще в 5-10 раз.

4 Кроссплатформенность. PyGame, как и Python в целом, поддерживает множество операционных систем – Windows (95+, включая

доступные 64-bit ОС), WindowsCE, BeOS, MacOS, MacOSX, FreeBSD, NetBSD, OpenBSD, BSD/OS, Solaris, IRIX, и QNX. Код также содержит неофициальную поддержку для AmigaOS, Dreamcast, Atari, AIX, OSF/Tru64, RISC OS, SymbianOS, и OS/2. В результате, необходимое для разработки ПО может быть легко установлено на любую из этих систем.

5 Не требует от GUI использование всех функций. Благодаря этому использовать PyGame можно без монитора, если инструмент используется для обработки игровых звуков или движений манипуляторами.

6 Небольшое количество кода. Ядро PyGame оснащено только нужными инструментами для разработки игр, включая библиотеки GUI и некоторые эффекты, разработанными отдельно сторонними программистами.

Благодаря простоте синтаксиса Python, написание игр на PyGame представляет собой легкую задачу для опытных программистов. Для начинающих на официальном сайте расположены FAQ, документация и уроки по разработке. За все время существования проекта было создано более тысячи игр разных жанров. Сообществу свободного ПО, которое осуществляет поддержку и развитие PyGame, а также принимает от пользователей сообщения об ошибках и быстро исправляет их, сделало это возможным.

А.В. Гордиевич (УО БрГТУ, Брест)

Науч. рук. **В.И. Хвещук**, канд. тех. наук, доцент

ИНФОРМАЦИОННЫЙ САЙТ ДЛЯ «БРЕСТГОРТРАНС»

Одной из важных возможностей Интернета – это оказание различных информационных услуг разным категориям пользователей. В работе представлены результаты разработки и реализации информационной системы (ИС) «Брестгортранс» для пользователей Интернета. Предприятие «Брестгортранс» обеспечивает реализацию перевозок населения городским транспортом в рамках г. Брест.

ИС «Брестгортранс» является Интернет ресурсом, обеспечивающим его пользователей следующей информацией: об услугах предприятия; о маршрутах движения городского транспорта; о ценах на проезд и о расписании движения городского транспорта и другие. Поиск необходимой информации по ресурсу возможен, как по ключевым словам, так и по заданным значениям.

ИС реализована на основе клиент-серверной архитектуры. Клиентская часть ИС (сайт) реализована с помощью возможностей языка разметки HTML, а поиск необходимой информации осуществляется

средствами языка JavaScript. Пользовательский интерфейс ориентирован на неквалифицированных пользователей.

Серверная часть ИС в виде реляционной базы данных (БД) реализована в рамках MicrosoftSQLServer.

Созданная ИС установлена на сервере «Брестгортранс», доменное имя зарегистрировано на hoster.by.

В настоящее время данная ИС находится на стадии опытной эксплуатации. Для ИС разработан комплект необходимой эксплуатационной документации: инструкция по установке ИС, тексты и описание программ, инструкция пользователя, описание БД и другие. На основе результатов опытной эксплуатации будут учтены замечания и пожелания, как заказчиков, так и пользователей данного ресурса.

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

Науч. рук. **Е.Е. Пугачёва**, ассистент

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ ALV ДЛЯ ФОРМИРОВАНИЯ ОТЧЁТОВ В СИСТЕМЕ SAP R3

Технология ALV – это интегрированный элемент среды разработки ABAPObjects. Она позволяет разработчикам приложений для системы SAPR3 быстро реализовать показ структурированных множеств данных с помощью трёх различных ALV инструментов, по одному для отображения:

- простые и двумерные таблицы;
- иерархические-последовательные списки;
- древовидные структуры.

Одним из наиболее используемых инструментов технологии ALV является ALV Grid Control – гибкий инструмент для отображения списков.

К функциям, предоставляемым ALV Grid Control, относятся:

- отображение неиерархических списков последовательно с современным дизайном;
- использование типичных функции списка – таких, как сортировка и фильтрация – без дополнительной работы по программированию;
- адаптивное определение функций списка и их расширение;
- одинаковая реакция программы на пользовательские действия (типа двойного нажатия на строке);
- связь с интерфейсом отчет/отчет.

ALV Grid Control не позволяет отобразить блок или иерархические списки. В настоящее время, простые списки могут быть отображены только в однострочном формате.

Для отображения данных как минимум должны быть предоставлены следующие два типа информации, чтобы отобразить данные:

- внутренняя таблица с данными для отображения – называемая таблицей выходных данных;
- описание структуры этих данных объявленной в ALV Grid Control через каталог полей или через соответствующую структуру словаря данных.

Каталог полей – таблица, которая содержит информацию относительно полей, которые будут отображены. Например, ALV использует эту таблицу, чтобы идентифицировать тип поля.

В основном, необходим каталог полей для каждого списка отображаемого в ALV.

Литература

1 Кречмер, Р. Разработка приложений SAPR/3 на языке ABAP/4 / Р. Кречмер, В. Вейс. – М.: Лори, 1998. – 348 с.

2 SAP Help Portal – The central place for SAP documentation [Электронный ресурс] / SAP. Документация. – Режим доступа: <http://help.sap.com>. – Дата доступа: 15.02.2014.

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

Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА И СОЗДАНИЕ ОБЪЕКТОВ КОНФИГУРАЦИИ ДЛЯ ПРЕДПРИЯТИЙ ТОРГОВЛИ

В работе используется платформа 1С:Предприятие 8.2 со встроенным языком «1С:Предприятие». Описываются возможности встроенного языка и работа в системе от запуска Конфигуратора и Отладки до настраивания панелей инструментов «под себя».

Одним из объектов конфигурации является *Справочник*. Он предназначен для работы со списками данных. Как правило, в работе любого предприятия торговли используются списки сотрудников, списки товаров, списки клиентов, поставщиков и т. д. Свойства и структура этих списков описываются в объектах конфигурации Справочник. Справочник состоит из *элементов*. Например, для справочника сотрудники элементом является сотрудник, для справочника товаров – товар и т.д. Существует пять форм Справочника (форма объекта, форма группы, форма списка, форма для выбора, форма для выбора группы).

Объект конфигурации *Документ* предназначен для описания информации о совершенных хозяйственных операциях или о событиях, произошедших в жизни организации вообще. В работе любой фирмы

используются такие документы, как приходные накладные, приказы о приеме на работу, платежные поручения, счета и т.д. Свойства и структура этих документов описываются в объектах конфигурации Документ. Существует три формы документа (форма объекта, форма списка, форма для выбора), наполнение которых было реализовано. Ещё одним объектом конфигурации является *Перечисление*. Он предназначен для описания структуры хранения постоянных наборов значений, не изменяемых в процессе работы конфигурации. В предлагаемой работе создается новый объект конфигурации Перечисление с именем ВидНоменклатуры. Следующим объектом конфигурации является *Регистр накопления*, который предназначен для описания структуры накопления данных. Этот объект конфигурации сформирован под именем ОстаткиМатериалов.

В результате работы с системой 1С:Предприятие были созданы наиболее часто используемые объекты конфигурации, практически реализуемые на любом предприятии торговли.

А.В. Грудовенко (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **В.Н. Леванцов**, ст. преподаватель

РАЗРАБОТКА ЖУРНАЛА УЧЕТА ТОВАРНО-МАТЕРИАЛЬНЫХ ЦЕННОСТЕЙ ДЛЯ ГМК «ХАТЫНЬ»

В процессе работы современных организаций часто встаёт проблема необходимости учета товарно-материальных ценностей. При этом необходимо учитывать множество факторов, таких как перемещение и выбытие, хозяйственные события поступления товарно-материальных ценностей (ТМЦ) и многие другие. Несмотря на то, что задачи учета ТМЦ достаточно глубоко изучены на практике же система учета имеет различия.

В свое время был сформулирован класс так называемых слабо формализованных задач (задачи, которые нельзя решить без участия эксперта в данной области). 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 складского учета. Для реализации автоматизированной системы была применена одноуровневая архитектура (СУБД MS Visual Fox Pro) сбора и анализа данных. Применение данной архитектуры позволило обеспечить быстрое внедрение системы, а также, при необходимости,

позволит 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, перемещения и списания, что, несомненно, повышает экономическую эффективность работы отдела бухгалтерского учета. Данная автоматизированная система может быть применена в соответствующих отделах организации любого уровня.

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

Науч. рук. **А.И. Кучеров**, ст. преподаватель

ВИРТУАЛИЗАЦИЯ РАБОЧИХ МЕСТ

Тема проекта актуальная, и ее развивают достаточно давно, как технологически (программно), так и с точки зрения аппаратной части. Можно услышать таких слова, как «тонкие клиенты», «виртуализация», «bring your own device», «удаленный доступ к данным», то, скорее всего речь идет о какой-либо из частей описывающих понятие виртуализация рабочих мест.

И, прежде чем углубляться в изучение данной темы, стоит ответить на самые простые вопросы, чтобы упорядочить те обрывки информации, которые уже есть у многих образованных людей в головах. Ну а тем, кто уже понимает о чем идет речь будет интересно узнать что-то новое или увидите проблему под другим углом.

Итак, виртуализация рабочего места. Что она из себя представляет? Прежде чем ответить на данный вопрос, предлагаем вам задуматься вообще о самом термине «рабочее пространство» и такой фразе, как «виртуализация рабочего места».

С точки зрения информационных технологий, конечно же, первые ассоциации будут с рабочим компьютером (и рабочим столом, как физическим, так и «виртуальным», который вам предлагает ваша операционная система). И именно виртуальный рабочий стол (системы) является отличным примером того, как виртуализируется рабочее место. На рабочем столе находятся активные (текущие) документы, программы, с

которыми вы работаете в данный момент, а многозадачность ОС и удобная файловая система упрощают вашу работу.

Несколько десятков лет назад вместо текстового редактора Microsoft Word (установленного на персональном компьютере) на рабочем столе (реальном) стояла печатающая машинка. Вместо приложения «калькулятор» – его «десктопный» аналог, подключенный к розетке или автономный. Вместо поиска от Google – толстая стопка папок, пронумерованная по годам (или буквам алфавита). Вместо адресной книги почтового клиента – телефонный справочник и стопка конвертов. Вместо систем CAD/CAE – набор чертежных принадлежностей и технической документации.

Сейчас все эти «бумажно-аппаратные» комплексы заменил персональный компьютер, а сами компьютеры (и мобильные телефоны) имеют не высокую цену. Сравните сложность производства стула, и, скажем, центрального процессора по техпроцессу в 45 нанометров. А ведь их стоимость сопоставима, и, порой, отличается не в 10–20 раз, а в 2–3.

Что отличает процесс виртуализации рабочих мест от обустройства персональных рабочих мест – масштабы и скорость. Виртуализация рабочего места сегодня – это перенос возможностей персонального компьютера (и всех его приложений) в «виртуальное» состояние, и выполняется оно подобно тому, как в прошлом «аппаратные» помощники офисных служащих получили свои цифровые аналоги, после чего оказались в музеях.

Если раньше виртуализация стремилась отвязать вас от множества конкретных вещей, то в наше время виртуализация рабочих мест – способ отвязать вас от конкретного «железа» компьютера. Вместо большого, занимающего место системного блока – маленькая коробочка, размером с домашний роутер, или монитор-моноблок. Вместо блока питания на 350 ватт – небольшой и экономичный БП, как у ноутбуков, не требующий активного охлаждения. Вместо отдельных мест для бухгалтеров, инженеров и менеджеров – универсальные рабочие места, готовые к работе в любое время. Все программы, данные и настройки хранятся на сервере, «железо» которого рассчитано на постоянные нагрузки, имеет высокий коэффициент надежности, а ПО грамотно распределяет доступные ресурсы между активными пользователями. Простаивающий компьютер инженера никак не поможет экономить на электроэнергии или ускорить работу его соседа. Виртуализация рабочих мест (в актуальном прочтении) решает эти и многие другие проблемы. Причем развертывание новых рабочих станций и расширение занимает минуты против часов в «классическом» случае. А вместо «неудобного» компьютера вы можете использовать ваш привычный ноутбук или планшет с док-станцией и внешним монитором. А самое главное –

внутри компании практически с любого компьютера вы можете получить доступ к своим документам, файлам и приложениям за считанные секунды, и при этом не бояться ни потенциальной утечки данных, ни нарушений уровня допуска, ни потенциальных дыр в защите чужого компьютера. Ни файлов, ни данных, ни чего-либо хоть сколько-нибудь ценного на клиенте не хранится.

Виртуализация десктопов (Virtual Desktop Infrastructure – VDI) – это создание рабочих столов в виртуальной среде. С помощью технологии виртуализации рабочих мест сотрудник, имея любое устройство с доступом в Интернет – смартфон, планшетный компьютер, тонкий клиент, – может получить доступ к персональному рабочему столу и корпоративным информационным ресурсам. Внедрение VDI позволяет компании упростить создание и администрирование рабочих мест пользователей, обеспечить гибкость своей IT-инфраструктуры.

В.В. Гузова (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА СЕТЕВОГО ПРИЛОЖЕНИЯ ПОД ОПЕРАЦИОННУЮ СИСТЕМУ ANDROID

Android – операционная система для мобильных устройств, основана на ядре Linux. На сегодняшний день система поддерживается компанией Google. Android включает в себя операционную систему, набор библиотек, виртуальную машину Dalvik и стандартный набор пользовательских приложений. Приложения для Android пишутся на популярном языке программирования Java. ОС может быть установлена на платформах ARM, MIPS, PowerArchitecture. Сегодня создание приложений под Android позволяет разработчикам использовать язык программирования Java, а недавно – и C/C++, благодаря пакету инструментов и библиотек Android Native Development Kit, который позволяет для системы писать собственные низкоуровневые модули.

Появление данной системы на рынке оказалось знаковым, ведь именно она была избрана подавляющим большинством производителей смартфонов, планшетов и т. д. Это сказалось и на популярности среди пользователей. Выбирая мобильное устройство, они приобретают и операционную систему, которая в нем установлена. Статистика показывает, что разработка приложений под Android постоянно увеличивает свои обороты, ведь эта платформа открывает широчайшие возможности перед создателями софта и игр. ОС Android весьма часто используется при создании программ для работы с клиентами. При умелом использовании

ОС Android оказывается действенным маркетинговым инструментом.

В настоящей работе было реализовано сетевое приложение для создания Android-приложений. Данное приложение позволяет пользователю самостоятельно заполнить контентом и упаковать в арк-файл и загрузить его себе на устройство. Контентом для приложения являются аудио файлы. Разработанное Android-приложение дает возможность работать с аудио файлами следующим образом: прослушать файл (с возможностью перемотки, установки на паузу), установить аудио файл на вызов/уведомление.

Серверная часть разработки позволяет полностью заполнить, в зависимости от фантазии пользователя, будущее Android-приложение.

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

Научн. рук. **В.Н. Леванцов**, ст. преподаватель

РАЗРАБОТКА ПРОЕКТА ОПЕРАТИВНОЙ ВИЗУАЛИЗАЦИИ РЕМОНТНЫХ РАБОТ НА ТРАМВАЙНЫХ ПУТЯХ

В представленной работе рассматривается разработка информационного и программного обеспечения для ввода, автоматизации хранения и обработки информации о состоянии трамвайных путей, запланированных ремонтных работах, а так же протекающих в данный момент ремонтных работах на трамвайных путях.

Простой и удобный способ добавления информации по ремонтным работам можно осуществлять через панель администратора, специально разработанную для этих целей.

Разработанное программное обеспечение должно решить основные вопросы, связанные с хранением и обработкой данных по ремонтным работам. Интерфейс должен быть ориентирован на пользователя, не имеющего специальной подготовки.

Разработанное приложение будет добавлено в разрабатываемое приложение «Е-сервисы г. Москва», как один из сервисов для просмотра информации по ситуации на трамвайных линиях, которое в свою очередь будет добавлено в приложение «ГУЦОДД», посредством которого осуществляется информационная поддержка населения.

Основные задачи подсистемы: добавление информации по работам на трамвайной линии или ее участке, включая статус проведенных работ, фотографии с места проведения работ с привязкой ко времени, а так же ее положение на карте. Система должна наглядно показывать состояние трамвайных путей на выбранную дату. Система должна осу-

шествовать аудит всех операций, таких как добавление, редактирование, удаление.

Основные операции: добавление, удаление, редактирование информации о трамвайных путях, включая фотографии с мест проведения работ с привязкой ко времени, редактирование положения трамвайных линий, работ на них на карте.

Подсистема позволяет структурировать и визуализировать собираемую информацию о состоянии трамвайных путей для последующего анализа и планирования ремонтных работ.

П.В. Дементьев (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **А.В. Воружев**, канд. тех. наук, доцент

ОРГАНИЗАЦИЯ ВЕЩАНИЯ МЕДИАПОТОКА В ГЕТЕРОГЕННОЙ ЛОКАЛЬНОЙ СЕТИ МОБИЛЬНЫХ УСТРОЙСТВ

Зачастую, возникает потребность продемонстрировать на экране устройства конечного пользователя (client* рисунок 1) в пределах одной сети рабочий стол одного из компьютеров (Admin рисунок 1). Необходимость проводить лекции посредством компьютерных технологий заставляет создавать определенные программные продукты. Одним из таких продуктов является OpenMeetings.

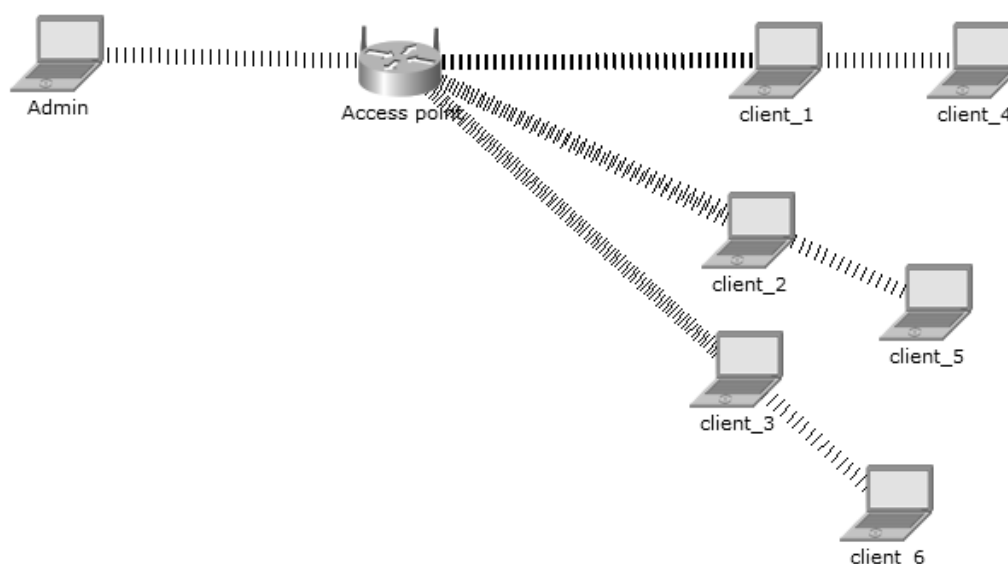


Рисунок 1 – Топология сети

Данная программная среда позволяет без клиентского программного обеспечения, используя только браузер, проводить конференции любого

формата с возможностью демонстрации рабочего стола. Интерфейс программы показан на рисунке 2.

Особенности и преимущества:

- Интеграция с СДО Moodle и другими СДО: возможность создавать вебинары прямо из курсов Moodle, переходить из Moodle в вебинары без повторной регистрации и авторизации.

- Свободное ПО: не требует ни разовых ни регулярных лицензионных платежей.

- Устанавливается на собственный сервер заказчика.

- Поддерживает режимы аудио и видео конференции нескольких участников или выступления одного участника.

- Белая доска: позволяет делать пометки и зарисовки, отображать картинки, видеозаписи и текстовые документы для участников вебинара.

- Демонстрация слайдов и презентаций.

- Модерация и управление доступом к вебинару.

- Функция записи видеовстречи.

- Функция демонстрации рабочего стола докладчика.

- Обмен файлами.

- Голосования и опросы.

- Предусмотрен импорт в конференцию документов в различных форматах (.tga, .xcf, .wpg, .txt, .ico, .ttf, .pcd, .pcds, .ps, .psd, .tiff, .bmp, .svg, .dpx, .exr, .jpg, .jpeg, .gif, .png, .ppt, .odp, .odt, .sxw, .wpd, .doc, .rtf, .txt, .ods, .sxc, .xls, .xsi, .pdf). После импорта они будут доступны другим участникам текущей конференции без ограничений. Оригинальный файл и его pdf-версия доступны для скачивания всем участникам. С целью просмотра и редактирования на доске конференции файлы конвертируются в форматы png и pdf. Участники могут редактировать загруженный документ, вводя текст поверх оригинала, рисовать, отмечать нужные места стрелками. Реализовано два типа опросов («Да/Нет» и ввод числовой оценки 1–10).

Минимальные системные требования к серверу:

- Процессор: 1GHz

- ОЗУ: 1Gb

- Место на жестком диске: 2 Gb

- Требования к пропускной способности определены более четко, каждое подключение к серверу требует 256 Кбит/сек. Хотя клиент может выбрать подключение с меньшим качеством, которое потребует меньшую ширину канала – 160 Кбит/сек.

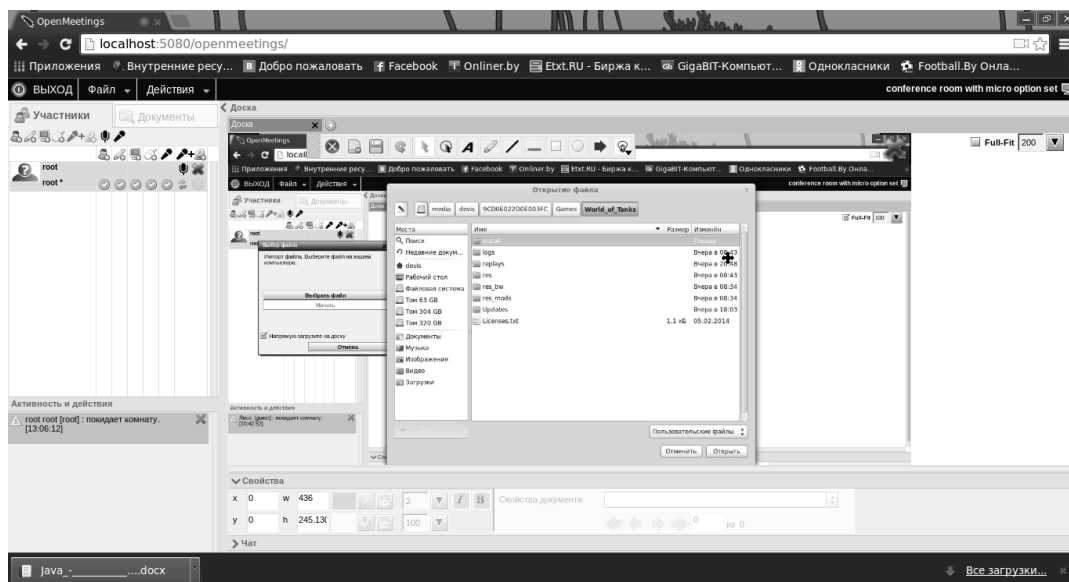


Рисунок 2 – Интерфейс программы в режиме конференции

В итоге для сервера нужно обеспечить (где N – количество участников):

- входящий канал – $256 \cdot N$ Кбит/с;
- исходящий канал – $256 \cdot N \cdot (N-1)$ Кбит/с.

Для клиентской системы:

- входящий канал – $256 \cdot (N-1)$ Кбит/с;
- исходящий канал – 256 Кбит/с.

И еще немаловажно, что для подключения клиентов к серверу не требуется установка дополнительного ПО, для этого используется веб-браузер с плагином для поддержки технологии Flash.

В рамках эксперимента было проведено пробное учебное занятие на котором к конференции было подключено 16 устройств. Были выявлены существенные недостатки, связанные с демонстрацией рабочего стола на Android устройствах – вещание видеозахвата экрана работает на них некорректно. В связи с этим следует выполнить дополнительную настройку кодеков либо изменить параметры вещания, заменив видео на серии скриншотов.

Литература

1. Установка и настройка Openmeetings 1.1 под Windows. – Режим доступа: <http://sysadminblog.ru/windows/2010/03/15/ustanovka-i-nastroyka-openmeetings-11-pod-windows.html>, свободный – Дата доступа: 25.02.2014.

2. Установка OpenMeetings v2.0 на Ubuntu 12.04. - Режим доступа: <http://proshin-roman.blogspot.com/2012/08/openmeetings-v20-ubuntu-1204.html>, свободный – Дата доступа: 25.02.2014.

С.С. Демиденко (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **Н.А. Шаповалова**, ст. преподаватель

ПРОЕКТ АВТОМАТИЗАЦИИ РАБОЧЕГО МЕСТА СОТРУДНИКА ЗЕМЛЕУСТРОИТЕЛЬНОЙ СЛУЖБЫ ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

Землеустроительная служба – это организация эффективного использования земельных ресурсов, составление проектов землеустройства, описание земельных участков, организация работ по улучшению земель, контроль процесса соблюдения земельного законодательства и охраны окружающей среды.

Проведя анализ существующих на сегодняшний день программных средств, было решено, что для автоматизации такого разностороннего перечня задач лучше всего подойдет программный продукт 1С: Предприятие от компании 1С версия 8.2. Встроенные прикладные объекты данного программного продукта отлично подходят для поставленной задачи.

База данных землеустроительной службы содержит весь перечень документов и отчетов, с которыми сталкиваются сотрудники землеустроительной службы.

Так как вся конфигурация создана на основе «Управляемого приложения», то при размещении конфигурации на веб-сервере, в ней можно будет находиться удаленно. Граждане смогут подавать заявления, не выходя из дома, а сотрудники получают значительную мобильность.

В ходе выполнения проекта было решено множество проблем, связанных с созданием документов, справочников, перечислений и регистров, а также печатных форм, соответствующих государственным нормам. Был спроектирован и реализован весь перечень отчетов, с которыми постоянно сталкиваются землеустроители. Все отчеты имеют параметры, что позволяет формировать их за любой промежуток времени.

Для удобства пользователя был реализован удобный, наглядный и интуитивно понятный интерфейс. Всего было разработано четыре подсистемы, в которых отображается весь рабочий процесс землеустроителя.

Для запуска приложения пользователю понадобится пройти аутентификацию. Это позволит уменьшить количество рисков, связанных с искажением и кражей корпоративных данных злоумышленниками. Причем, все пользователи имеют различный спектр возможностей по манипуляции над данными. По полномочиям сотрудников за каждым пользователем закреплены определенные роли на создание, чтение, редактирование и удаление данных из конфигурации.

Приложение «Землеустроительная служба» полностью функционирует и готово к использованию. Разработанные объекты конфигурации позволяют легко обрабатывать необходимый спектр данных. Программный комплекс правильно выполняет запросы пользователя и соответствует предъявляемым требованиям.

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

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

ОСОБЕННОСТИ АВТОМАТИЗАЦИИ РАБОЧЕГО МЕСТА СОТРУДНИКА ЗЕМЛЕУСТРОИТЕЛЬНОЙ СЛУЖБЫ ИСПОЛНИТЕЛЬНОГО КОМИТЕТА

В основе данного проекта лежит задача создания программного комплекса, охватывающего весь перечень прикладных задач, с которыми постоянно сталкиваются сотрудники землеустроительных служб, а также построение системы, которая позволит эффективно использовать рабочее время, формировать документацию в приемлемом виде, снизить количество бумажного архива документов, а также ускорить и облегчить рабочий процесс.

Сотрудники землеустроительной службы занимаются организацией эффективного использования земельных ресурсов, составлением проектов землеустройства, описанием земельных участков, организацией работ по улучшению земель, контролем процесса соблюдения земельного законодательства, охраной окружающей среды.

Проведя анализ существующих на сегодняшний день программных средств, было решено, что для автоматизации такого разностороннего проекта лучше всего подойдет программный продукт 1С: Предприятие от компании 1С версии 8.2. Данный продукт предназначен для автоматизации деятельности на предприятии. Первоначально 1С: Предприятие было предназначено для автоматизации бухгалтерского и управленческого учета (включая начисление зарплаты и управление кадрами), но сегодня этот продукт находит свое применение в областях, далеких от собственно бухгалтерских задач. Технологическая платформа 1С: Предприятия представляет собой программную оболочку над базой данных.

Главной отличительной особенностью 1С: Предприятия версии 8.2 от ее предыдущих версий является режим «управляемое приложение», при котором пользовательский интерфейс описывается декларативно, а его вид зависит от типа клиентского программного обеспечения.

Встроенные прикладные объекты данного программного продукта отлично подходят для поставленной задачи.

База данных «Землеустроительная служба» хранит большой и разносторонний спектр информации:

- весь перечень данных связанный с физическими и юридическими лицами, а также частными предпринимателями;
- информацию о земельных участках;
- данные о кадастровых стоимостях земельных участков;
- заявления на приватизацию земельных участков;
- документы на приобретение участков;
- данные о сотрудниках исполнительных комитетов;
- данные о целевых назначениях, ограничениях и правах на земельные участки и иной справочной информации.

База данных землеустроительной службы содержит весь перечень документов и отчетов, с которыми сталкиваются сотрудники землеустроительной службы.

Пользовательский интерфейс обеспечивает отображения хранимых данных в удобном и наглядном для пользователя виде, что позволит с легкостью проводить над ними различные манипуляции.

Весь перечень метаданных велик, поэтому остановимся только на некоторых, более интересных объектах конфигурации. Одним из таких объектов является справочник Адреса. Обычно для заполнения адресной информации в документах, отчетах и иных справочниках применяются довольно сложные и громоздкие адресные классификаторы, состоящие из достаточно весомого количества объектов метаданных (регистры сведений, справочники, перечисления). При использовании таких классификаторов пользователю придется поочередно выбирать в форме каждый элемент (страна, область, город и т.д.) из огромного количества данных, которые не всегда наглядны и плохо структурируемы. Данный подход, на мой взгляд, занимает достаточно много времени для заполнения всего лишь одного адреса, а сотрудники землеустроительных служб постоянно имеют дело с адресами. Поэтому было решено создать один справочник, который заменит целый классификатор. Главной особенностью данного справочника является его иерархическая структура. Иерархия распространяется только на элементы справочника и имеет шесть иерархических ступеней. Верхняя ступень иерархии - это страна, далее область, район, сельский совет, населенный пункт. Последней ступенью является улица или иное наименование объекта (например, карьер). Такая зависимость является хорошо структурируемой, более наглядной и уменьшит время заполнения адресных данных. По такой адресной структуре хорошо формируются различные отчеты в разрезе любой ступени иерархии.

Вторым по массивности является справочник Физические лица, который хранит паспортные, контактные данные о физических лицах, а также ксерокопии паспорта и данные о месте проживания, которые могут заполняться на основании прописки. Остальные справочники хранят весь перечень постоянной (базовой) информации, что позволяет формировать документы практически декларативно (без использования клавиатуры).

Что касается документов, то чаще всего сотрудники землеустроительных служб сталкиваются с двумя типами документов: заявлениями от граждан (физических лиц) и формированием землеустроительных дел по данным заявлениям. До создания конфигурации сотрудники формировали документы с использованием текстового редактора MS Word, либо «от руки», на что уходило большое количество времени (заявление до получаса, землеустроительное дело от нескольких дней до недель). С помощью созданного приложения можно формировать данные документы за считанные минуты. Для этого был создан шаблон, включающий в себя весь перечень видов заявлений и землеустроительных дел, а также весь перечень данных входящих в эти документы, что свело огромную разнообразность подобных документов к двум видам: Заявления физических лиц и Приобретение земельного участка (формирование землеустроительного дела). Пользователю необходимо лишь правильно заполнить форму документа, после чего система сама сформирует необходимую форму для печати. Практически все реквизиты документа заполняются на основании справочников и перечислений, т.е. пользователь выбирает данные по клику мыши. На основании хранимых документов можно с легкостью строить отчеты различной функциональности и сложности.

В конфигурацию включен весь перечень отчетов, с которыми сталкиваются землеустроители. Например, отчет «О наличии земель граждан» передавался ежегодно из землеустроительной службы в налоговую службу, и на его формирование уходили большие трудозатраты и большое количество времени, т.к. он содержит расчет кадастровой стоимости по каждому земельному участку. Отчет создавался в табличном редакторе MS Excel. Благодаря системе 1С данный отчет формируется за считанные секунды, после чего его можно распечатать или выгрузить в MS Excel.

Т.к. вся конфигурация создана на основе «Управляемого приложения», то при размещении конфигурации на веб-сервере, в ней можно будет находиться удаленно. Граждане смогут подавать заявления, не выходя из дома, а сотрудники получат значительную мобильность (смогут просматривать, добавлять, изменять данные, находясь на выезде или в командировках).

В ходе выполнения данного проекта была проделана работа по проектированию и разработке конфигурации для землеустроительной службы в среде 1С: Предприятие версии 8.2. Было решено множество проблем, связанных с созданием документов и реализацией печатных форм соответствующих государственным нормам, созданием справочников для ввода первичной информации хранимой в базе данных, разработкой необходимых перечислений, созданием регистров для проводки документов. Был спроектирован и реализован весь перечень отчетов, с которыми постоянно сталкиваются землеустроители. Все отчеты имеют параметры, которые позволяют формировать их за любой промежуток времени.

Большая часть реквизитов документов заполняется на основании справочников и перечислений, что позволяет сэкономить время на заполнение документов.

Для удобства пользователя был реализован удобный, наглядный и интуитивно понятный интерфейс. Всего было разработано четыре подсистемы, в которых отображается весь рабочий процесс землеустроителя.

Для запуска приложения пользователю понадобится пройти аутентификацию. Это позволит уменьшить количество рисков, связанных с искажением и кражей корпоративных данных злоумышленниками. Причем, все пользователи имеют различный спектр возможностей по манипуляции над данными. По полномочиям сотрудников за каждым пользователем закреплены определенные роли на создание, чтение, редактирование и удаление данных из конфигурации.

Главным результатом проведённой работы является создание полноценной конфигурации «Землеустроительной службы». Приложение полностью функционирует и готово к использованию. Разработанные объекты конфигурации позволят легко обрабатывать необходимые данные. Данный программный комплекс правильно выполняет запросы пользователя и соответствует предъявляемым требованиям.

К.С. Демченко (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **В.Н. Леванцов**, ст. преподаватель

ВНЕШНИЕ ОБРАБОТКИ ОТЧЕТОВ В СРЕДЕ 1С: ПРЕДПРИЯТИЕ 8.2

Сегодня одним из весомых условий успешной профессиональной деятельности на современном рынке труда среди экономистов, работников финансовых служб, бухгалтеров и менеджеров является знание и свободное владение навыками работы с системой «1С: Предприятие

8.2». Корпоративная информационная система «1С: Предприятие 8.2» является гибкой настраиваемой системой, с помощью которой можно решать широкий круг задач в сфере автоматизации деятельности предприятий.

За счет своей универсальности система «1С: Предприятие» может быть использована для автоматизации самых разных участков экономической деятельности предприятия: учета товарных и материальных средств взаиморасчетов с контрагентами, расчета заработной платы и т.д. Встроенный макроязык системы 1С позволяет реорганизовать самостоятельно практически все виды учета и отчетности по желанию пользователя.

«1С: Предприятие» – система автоматизации экономической и организационной деятельности предприятия. Поскольку такая деятельность может быть довольно разнообразной, система «1С: Предприятие» может «приспосабливаться» к особенностям конкретной области деятельности, в которой она применяется. Для обозначения такой способности используется термин конфигурируемость, то есть возможность настройки системы на особенности конкретного предприятия и класса решаемых задач.

Инструменты платформы 1С:Предприятие 8, позволяют создавать дополнительные отчеты и механизмы обработки данных под специфические задачи конкретной организации или пользователя. Внешние отчеты и обработки расширяют возможности прикладного решения, позволяют получать информацию в тех разрезах и в том виде, которые не заложены в типовом программном продукте. Такая возможность позволяет автоматизировать специфические задачи и процессы компании, что особенно важно в области управления персоналом, где уникальные технологии управления являются основой преимущества компании на рынке труда.

Внешние отчеты представляют собой отчеты, которые не входят в состав прикладного решения и хранятся в отдельных файлах с расширением *.erf.

Основное их преимущество заключается в том, что такие отчеты можно использовать в различных прикладных решениях без изменения структуры самих решений. Кроме того, важным преимуществом внешних отчетов является возможность проектировать и отлаживать их в процессе работы 1С:Предприятия, без необходимости сохранения каждый раз конфигурации прикладного решения.

В режиме 1С:Предприятие внешний отчет можно запустить на выполнение, открыв его как любой другой файл, хранящийся на диске. Такой отчет будет работать точно так же, как если бы он входил в состав

прикладного решения.

Любой существующий в конфигурации отчет или обработка может быть преобразован во внешний, и наоборот. Кроме этого внешние обработки и отчеты могут быть добавлены в структуру прикладного решения как новые объекты.

Конфигуратор поддерживает также возможность сравнения и объединения существующей в конфигурации обработки или отчета с внешней обработкой и отчетом, а также сравнение и объединение двух внешних отчетов или обработок.

Д.В. Деревянко (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **А.И. Кучеров**, ст. преподаватель

АНАЛИЗ ШАБЛОНОВ ПРОЕКТИРОВАНИЯ

Шаблоном проектирования или паттернами называют решения часто встречающихся проблем в области разработки программного обеспечения.

Паттерны проектирования ООП определяют только подход к проектированию и не зависят от языка программирования. Создаются эти решения на основе жизненных ситуаций моделируемых реальных объектов нашего мира.

Разрабатывая классы, определяя взаимодействия и обязанности между ними, перед программистом возникает проблема моделирования объектов. Существует множество таких шаблонов, и важным является правильный выбор. В случае если выбор был сделан неверный, через определённый момент времени разрабатываемое приложение невозможно будет воспринимать, поддерживать, модернизировать, не говоря уже о повторном использовании отдельных компонентов. Основное внимание следует уделять назначению шаблона и последствиям его применения.

Существуют несколько типов паттернов проектирования, каждый из которых решает свой круг задач:

4. Порождающие паттерны.
5. Структурные паттерны.
6. Паттерны поведения.

К первому типу относятся шаблоны, которые создают новые объекты или предоставляют доступ к существующим объектам. Самой распространённой задачей при разработке программных систем является создание новых объектов. Шаблоны данного вносят в программную систему независимость от процесса порождения и от типов объектов. В

качестве примера можно привести шаблон «Фабрика». Он используется, когда требуется получать какие-либо объекты и совершенно не важно, как они создаются. Как создавать объект решает фабрика, например, создавать новый или копировать из существующего объекта. С его помощью можно создавать объекты нужных классов, не указывая напрямую их типы. Паттерн «Одиночка» контролирует создание единственного экземпляра некоторого класса и предоставляет доступ к нему

Второй тип включает паттерны, которые решают проблемы компоновки системы на основе классов и объектов. Они вносят порядок и учат разные объекты более правильно взаимодействовать друг с другом. Существуют два основных механизма компоновки: наследование и композиция. Наследование используется тогда, когда базовый класс определяет интерфейс, а подклассы – реализацию. Структуры на основе наследования получаются статичными. А композиция, когда структуры строятся путем объединения объектов некоторых классов. Композиция позволяет получать структуры, которые можно изменять во время выполнения. Например, паттерн «Адаптер» представляет собой программную обертку над уже существующими классами и предназначен для преобразования их интерфейсов к виду, пригодному для последующего использования в новом программном проекте. Шаблон «Компоновщик» минимизирует различия в управлении как группами объектов, так и индивидуальными объектами.

Последний тип шаблонов предназначен для распределения обязанностей между объектами в системе. Эта группа паттернов позволяет структурировать подходы к обработке поведения и взаимодействия объектов. Проще говоря, как должны проходить процессы, в которых существует несколько вариантов протекания событий. Например, паттерн «Состояние» позволяет изменять свое поведение в зависимости от внутреннего состояния. Создается впечатление, что объект изменил свой класс, а шаблон «Наблюдатель» определяет зависимость «один-ко-многим» между объектами так, что при изменении состояния одного объекта все зависящие от него объекты уведомляются и обновляются автоматически.

Задача каждого паттерна – дать четкое описание проблемы и ее решения в соответствующей области. Для этого могут использоваться разные форматы описаний от художественно-описательного до строгого, академического. Правильно выбранные шаблоны проектирования позволяют сделать программную систему более гибкой, ее легче поддерживать и модифицировать, а код такой системы в большей степени соответствует концепции повторного использования.

Литература

1. Эккель, Б. Философия Java/ Б. Эккель – М.: Питер, 2010.– 640 с.
2. Шилдт, Г. Полный справочник по Java/ Г. Шилдт. – М.: Вильямс, 2009 – 1040 с.
3. Блинов, И.Н. Java. Промышленное программирование/ И.Н. Блинов, В.С. Романчик – М.: УниверсалПресс, 2007. – 704 с.
4. Гамма, Э. Приемы объектно-ориентированного проектирования. Патерны проектирования./ Э. Гамма, Р. Хелм, Р. Джонсон, Дж Влссидес – СПб.: Питер, 2001. – 368 с.

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

Науч. рук. **А.И. Кучеров**, ст. преподаватель

ЭТАПЫ РАЗРАБОТКИ ВЕБ-ПОРТАЛА НА БАЗЕ IBM PORTAL SERVER

Программный продукт WebSphere Portal фирмы IBM является одним из признанных лидеров на рынке платформ для построения Интернет-порталов. Одной из сильных сторон платформы WebSphere Portal является возможность расширения функциональности за счет интеграции с внешними приложениями, такими как Lotus Sametime, Lotus Mail и др., посредством специально разработанных портлетов и сервлетов. Это позволяет использовать уже существующие базы данных и системы документооборота.

Средства персонализации WebSphere Portal позволяют каждому пользователю портала самостоятельно создавать и настраивать личные рабочие страницы в соответствии с индивидуальными требованиями.

Основные этапы создания портала на платформе WebSphere Portal:

1. Установка портал-сервера и настройка рабочих мест разработчиков.
2. Установка и настройка готовых портлетов.
3. Настройка совместной работы портала с внешними приложениями.
4. Разработка и установка новых портлетов для расширения функциональности.
5. Распределение прав пользователей.
6. Настройка дизайна страниц и портлетов.

Следует уделить внимание возможности использования фреймворков и шаблонам проектирования, так как они значительно упрощают разработку приложений. На самой начальной стадии проектирования нужно выделить основные требования к приложению, а затем исходя из этих требований, отдать предпочтение наиболее подходящему фрейм-

ворку. Например, в качестве шаблона использовать Spring Portlet MVC Framework. Он является общепринятым подходом к построению веб-приложений таким образом, чтобы пользовательский интерфейс был отделён от логики приложения. Приложения, основанные на портлетах, сосредотачивают несколько частей функциональности в одну веб-страницу.

Разработка приложения включает в себя проектирование базы данных. Много времени тратится на проектирование таблиц, связей и написание sql-запросов. Для экономии времени можно использовать сторонние библиотеки, например Hibernate. Она предназначена для решения задач объектно-реляционного отображения, позволяет разработчику отказаться от значительного объёма сравнительно низкоуровневого программирования по обеспечению хранения объектов в реляционной базе данных. Фреймворк не только решает задачу связи классов Java с таблицами базы данных, но также предоставляет средства для автоматической генерации и обновления набора таблиц, построения запросов и обработки полученных данных и может значительно уменьшить время разработки, которое обычно тратится на ручное написание SQL и JDBC-кода.

Mapping Java классов с таблицами базы данных осуществляется с помощью конфигурационных Java-аннотаций. Обеспечиваются возможности по организации отношения между классами «один-ко-многим» и «многие-ко-многим». В дополнение к управлению связями между объектами, Hibernate также может управлять рефлексивными отношениями, где объект имеет связь «один-ко-многим» с другими экземплярами своего собственного типа данных.

С целью упорядочения, систематизации и обмена созданными портлетами, можно сформировать библиотека портлетов. Каждый портлет данной библиотеки может использоваться независимо от других на любом сервере WebSphere Portal Server.

Основные достоинства платформы IBM WebSphere Portal:

- Практически неограниченные возможности по настройке по масштабированию.
- Мощная система администрирования.
- Интеграция с внешними приложениями.
- Использование стандартных способов разработки веб-приложений на Java.
- Высокий уровень технической поддержки.

К основным недостаткам платформы можно отнести:

- Сложность поиска необходимой информации в большом объеме документации.

- Небольшой изначальный объем библиотеки портлетов (которая, однако, постоянно расширяется сторонними разработчиками).
- Высокие требования к аппаратным ресурсам.
- Недоработки в некоторых программных модулях.

В целом, можно сделать вывод, что использование платформы WebSphere Portal позволяет весьма эффективно работать над масштабными и постоянно развивающимися проектами.

Литература

1. Эккель, Б. Философия Java/ Б. Эккель – М.: Питер, 2010.– 640 с.
2. Sarin, A. Portlets in action/ A. Sarin, E. Macel, J. Bleiel – М.: Manning Publications Co, 2008. – 614 с.
3. Шилдт, Г. Полный справочник по Java/ Г. Шилдт. – М.: Вильямс, 2009 – 1040 с.
4. Блинов, И.Н., Java. Промышленное программирование / И.Н. Блинов, В.С. Романчик. – М.: УниверсалПресс, 2007. – 704 с.

Н.Н. Диваков (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **А.В. Воруев**, канд. тех. наук, доцент

ИСЧЕРПАНИЕ IPv4-АДРЕСОВ

Исчерпание IPv4-адресов – истощение запаса нераспределённых адресов протокола IPv4. Адресное пространство глобально управляется американской некоммерческой организацией IANA, а также пятью региональными интернет-регистраторами, ответственными за назначение IP-адресов конечным пользователям на определённых территориях, и локальными интернет-регистраторами, такими как интернет-провайдеры.

IPv4 позволяет использовать около 4,23 миллиарда адресов, и часть из них была распределена IANA региональным интернет-регистраторам блоками примерно по 16,8 миллионов адресов (с учётом использования CIDR). В феврале 2011 года IANA выделила региональным интернет-регистраторам последние пять оставшихся блоков /8 из своего адресного пространства APNIC является первым региональным интернет-регистратором, выделенные IP-адреса которого закончились. Это произошло 15 апреля 2011 года некоторых частях мира уже закончились распределённые IPv4-адреса, и истощение адресов оставшихся региональных интернет-регистраторов ожидается в течение пяти лет.

Исчерпание запаса IP-адресов региональных интернет-регистраторов в 2011 году 31 января 2011 года последние два незаре-

зервированных IANA блока адресов были выделены APNIC в соответствии со стандартными процедурами выделения адресов региональным интернет-регистраторам. Осталось пять зарезервированных и потому нераспределённых блоков /8. В соответствии с правилами ICANN, IANA приступила к выделению каждого из этих блоков каждому из региональных интернет-регистраторов после пресс-конференции 3 февраля 2011 года, что привело к окончанию запаса адресов IANA (рисунок 1).

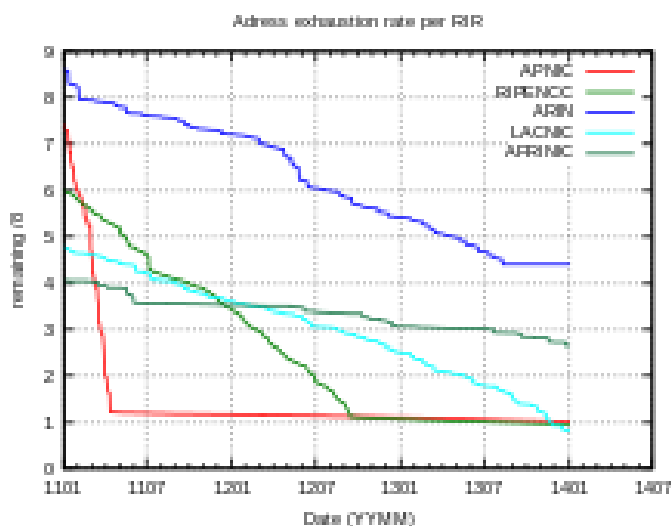


Рисунок 1 – Хронология исчерпания

Различные самостоятельные адресные блоки, исторически используемые отдельно от региональных интернет-регистраторов, были распределены между региональным интернет-регистраторам в феврале 2011 года.

Ряд технологий уменьшают потребность в IP-адресах:

- NAVING – (имея) в результате группы, которые удовлетворяют указанному перечню условий отбора групп;
- NAT, прокси-серверы и внутрисетевая адресация.

Технология NAT (Network address translation) позволяет нескольким компьютерам иметь один внешний IP-адрес. Находящиеся за NAT компьютеры могут подключаться друг к другу, используя внутрисетевые IP-адреса, но извне с такими компьютерами без специальной настройки соединиться невозможно.

Виртуальный хостинг веб-сайтов с доступом по доменному имени.

Несколько сайтов имеют общий IP-адрес, сервер отличает один от другого по доменному имени (поле Host HTTP/1.1).

Тщательный контроль региональных интернет-регистраторов за присвоением IP-адресов локальным интернет-регистраторам.

Перераспределение адресного пространства .

В первые годы существования Интернета использовалась неэффективная система классовой адресации. Большие блоки IP-адресов, распределённые в те времена, возвращаются в оборот.

Н.Н. Диваков (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **А.В. Воруев**, канд. тех. наук, доцент

ИСПОЛЬЗОВАНИЕ ПРОТОКОЛА ДНСР

Стандарт протокола ДНСР был принят в октябре 1993 года. Действующая версия протокола (март 1997 года) описана в RFC 2131. Новая версия ДНСР, предназначенная для использования в среде IPv6, носит название ДНСРv6 и определена в RFC 3315 (июль 2003 года).

ДНСР (англ. Dynamic Host Configuration Protocol – протокол динамической конфигурации узла) – это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому серверу ДНСР, и получает от него нужные параметры. Сетевой администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол ДНСР используется в большинстве сетей TCP/IP.

ДНСР является расширением протокола BOOTP, использовавшегося ранее для обеспечения бездисковых рабочих станций IP-адресами при их загрузке. ДНСР сохраняет обратную совместимость с BOOTP.

Помимо IP-адреса, ДНСР также может сообщать клиенту дополнительные параметры, необходимые для нормальной работы в сети. Эти параметры называются опциями ДНСР. Список стандартных опций можно найти в RFC 2132.

Некоторыми из наиболее часто используемых опций являются:

- IP-адрес маршрутизатора по умолчанию;
- маска подсети;
- адреса серверов DNS;
- имя домена DNS.

Протокол ДНСР поддерживает три механизма выделения адресов: автоматический, динамический и ручной. В первом случае клиент получает постоянный IP-адрес, в последнем ДНСР используется только для уведомления клиента об адресе, который администратор присвоил ему вручную. Оба эти варианта не таят в себе чего-либо принципиально нового, а вот динамический механизм заслуживает детального рассмотрения. Вы-

дача адреса в аренду производится по запросу клиента. DHCP-сервер (или группа серверов) гарантирует, что выделенный адрес до истечения срока его аренды не будет выдан другому клиенту; при повторных обращениях сервер старается предложить клиенту адрес, которым тот пользовался ранее. Со своей стороны, клиент может запросить пролонгацию срока аренды адреса либо, наоборот, досрочно отказаться от него. Протоколом предусмотрена также выдача IP-адреса в неограниченное пользование. При острой нехватке адресов сервер может сократить срок аренды адреса по сравнению с запрошенным. Выдача нового адреса. Последовательность событий в этом случае такова (рисунок 1).

Хранение параметров сетевой конфигурации станций-клиентов является второй услугой, предоставляемой DHCP-сервером. В создаваемой базе данных на каждого клиента заводится отдельная запись с уникальным ключом-идентификатором и строкой конфигурационных параметров.

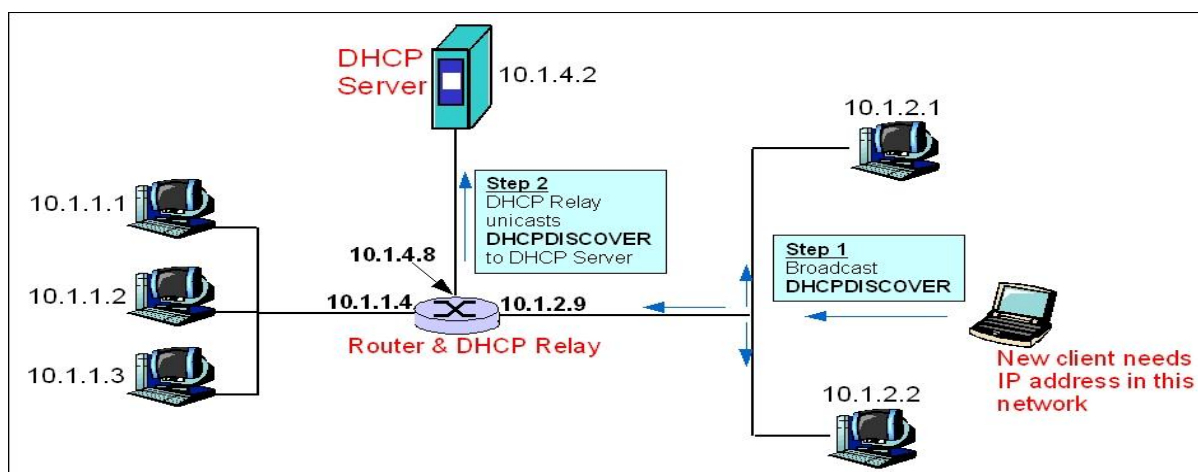


Рисунок 1– Последовательность событий при выделении IP-адреса

Н.Н. Диваков (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **А.В. Воруев**, канд. тех. наук, доцент

IPv6 ПРОТОКОЛЫ

ICMPv6. ICMP в IPv6 был заменён на ICMPv6. О ICMPv6 можно прочитать в RFC4443 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. Сам по себе ICMPv6 довольно прост, однако на его основе сделано множество довольно не тривиальных протоколов, о которых мы поговорим чуть ниже. MLD. Multicast Listener Discovery – протокол на основе ICMPv6. Аналог IGMP для IPv6. Подробнее в RFC3810 – Multicast Listener Discovery Version 2 (MLDv2) for IPv6.

NDP. Описание Neighbor Discovery Protocol, заменившего протокол ARP, доступно в RFC4861. Теперь это не отдельный протокол, а надстройка над ICMPv6 добавляющая несколько новых типов сообщений. Основное предназначение NDP – производить мапинг между link-layer и IPv6 адресами, однако это лишь небольшая часть функциональности.

Neighbour UnreachabilityDetection, позволяющая определить доступность хоста одним из двух методов:

с помощью протокола более высокого уровня:

- с помощью unicast solicitation, на который хост должен ответить Neighbor Advertisement сообщением;

- Duplicate Address Detection;

- отсылка и обработка redirect'ов;

- определение параметров линка (например MTU);

- определение сервисов в сети (например DNS: RFC6106 IPv6 Router Advertisement Options for DNS Configuration);

- Stateless автоконфигурация (об этом чуть ниже);

- Также есть криптографически защищённый аналог NDP – SEcure Neighbor Discovery (SEND) описанный в RFC3971. SEND, в свою очередь, использует Cryptographically Generated Address описанные рядом в RFC3972.

- Автоконфигурация. Как уже было упомянуто выше, хосты умеют автоматически генерировать себе IPv6 link-local адрес из адреса канального уровня. Так что без какой либо настройки любой IPv6-enabled хост подключённый к сети выдаёт сам себе адрес сетевого уровня. В IPv4 эта технология использует зарезервированный IPv4 диапазон 169.254/16. Подробно технология описана в RFC3927 Dynamic Configuration of IPv4 Link-Local Addresses (Заметьте, что этот RFC вышел после IPv6'ого 2462). В IPv4 автоконфигурация возможна только с использованием DHCP сервера. В IPv6 эту возможность оставили: можно конфигурировать сеть с помощью DHCPv6 сервера и клиента. Однако, поддержка со стороны вендоров DHCPv6 пока не блещет, так например, dhclient во FreeBSD из коробки не умеет. IPv6. Кроме stateful-конфигурации в IPv6 имеется также stateless автоконфигурация, которая позволяет хостам автоматически получать IPv6 адреса в сети без DHCP сервера через использование NDP. Роутеру достаточно задать префикс который он будет анонсировать через Router Advertisement всей остальной сети. Хосты будут получать RA и формировать на основе префикса и своего link-layer адреса IPv6 адрес.

- Пара замечаний:

- stateless определение адреса возможно только при наличии роутера/ротеров рассылающего RA;

- каждый роутер имеет приоритет: high/medium/low. Операционная система должна его учитывать при выборе default route;
- RFC6106 – IPv6 Router Advertisement Options for DNS Configuration объясняет как встраивать адреса DNS-серверов прямо в RA, что позволяет избавиться от использования DHCPv6 для этого дела. Однако это поддерживается не всеми вендорами.

Н.Г. Дикун (УО «ГГУ им. Ф. Скорины», Гомель)

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

СИСТЕМА КОМПОНОВКИ ДАННЫХ 1С: ПРЕДПРИЯТИЯ 8.2

Итоги любой деятельности организации отображаются в отчетности, которая представляет собой совокупность показателей, характеризующих деятельность организации за определенный период времени.

Платформа «1С: Предприятие» содержит мощный и гибкий механизм для формирования сложных экономических и аналитических отчетов – систему компоновки данных. Отчеты строятся на основе схемы компоновки данных – специального макета отчета, содержащего его декларативное описание.

Использование декларативного описания отчетов позволяет реализовать следующие возможности:

- создание отчета без программирования;
- возможность создания различных вариантов отчета;
- возможность задания различных вариантов пользовательских настроек;
- использование автоматически генерируемых форм просмотра и настройки отчета;
- разбиение исполнения отчета на этапы;
- исполнение отдельных этапов построения отчета на различных компьютерах;
- независимое использование отдельных частей системы компоновки данных;
- программное влияние на процесс выполнения отчета;
- настройки структуры отчета;
- совмещение в отчете нескольких таблиц;
- создание вложенных отчетов и др.

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.

Разработчик создаёт схему компоновки данных, в которой описывает текст запроса, наборы данных, связи между ними, доступные поля, параметры получения данных, и задаёт первоначальные настройки компоновки – структуру отчёта, макет оформления данных и др.

Отчёт системы компоновки имеет сложную иерархическую структуру и может состоять из различных элементов, таких как группировки, таблицы и диаграммы. При этом пользователь может изменить существующую структуру отчёта или вообще создать совершенно новую структуру отчёта.

При формировании и исполнении отчёта происходит последовательная передача данных от одного объекта системы компоновки данных к другому, до получения конечного результата – документа, показанного пользователю.

Алгоритм взаимодействия этих объектов выглядит следующим образом:

Разработчик создаёт схему компоновки данных и настройки по умолчанию. В общем случае, на основе одной схемы компоновки данных может быть создано большое количество различных отчётов. Настройки компоновки данных, создаваемые разработчиком или изменяемые пользователем, определяют, какой именно отчёт будет получен в конкретном случае.

На основе схемы компоновки и имеющихся настроек компоновщик макета создаёт макет. Это этап подготовки к исполнению отчёта. Макет компоновки данных является уже готовым заданием для выполнения процессором компоновки. Он содержит необходимые запросы, макеты областей отчёта и др.

Процессор компоновки данных выбирает данные из информационной базы согласно макету компоновки, агрегирует и оформляет эти данные.

Результат компоновки обрабатывается процессором вывода, и в итоге пользователь получает результирующий табличный документ.

О.В. Дробышевская (УО «ГГУ им. Ф.Скорины», Гомель)

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

SELF-ORGANIZING INCREMENTAL NEURAL NETWORK – АЛГОРИТМ ОБУЧЕНИЯ БЕЗ УЧИТЕЛЯ

Одной из задач обучения без учителя является задача нахождения топологической структуры, которая наиболее точно отражает

топологию распределения входных данных. Фундаментальная проблема для таких задач – это как система может приспособиться к новой информации без повреждения или уничтожения уже известной. Алгоритм SOINN позволяет частично избежать этого.

SOINN представляет собой нейронную сеть с двумя слоями. Первый слой используется для определения топологической структуры кластеров, а второй – для определения числа кластеров и выявления узлов-прототипов для них. Сначала обучается первый слой сети, а затем, используя данные полученные при обучении первого слоя, обучается второй слой сети.

Для решения задачи обучения без учителя надо определить, относится ли новый образ к какому-нибудь кластеру (классу, группе) или он представляет собой абсолютно новую группу. Предположим, что два образа относятся к одному кластеру в том случае, если расстояние между ними меньше длины расстояния T . Если T задается большим, то все образы будут попадать в группы, если маленьким, то каждый образ будет создавать новый кластер.

Задача состоит в том, чтобы выяснить это T . T для первого слоя подбирается, основываясь на некоторых знаниях построенной сети и текущего входного образа. А вот при построении второго слоя возможно посчитать внутриклассовое и межклассовое расстояние, чтобы подобрать значение порога T .

Для первого слоя порог подобия T вычисляется адаптивно по следующему алгоритму.

Сначала инициализируется порог подобия для новых узлов равным $+\infty$. Когда узел является первым или вторым победителем, обновляется значение порога подобия. Если узел не имеет прямых топологических соседей ($L_i > 0$), значение T_i обновляется как максимальное расстояние между узлом и его соседями: $T_i = \max \| W_i - W_j \|$

Если узел не имеет соседей, T_i устанавливается как минимальное расстояние между узлом и другими узлами в множестве A :

$$T_i = \min \| W_i - W_c \|$$

Для второго слоя порог подобия постоянный и вычисляется следующим образом:

$$\text{Внутриклассовое расстояние: } d_w = 1 / N_c * \sum_{\{(i,j) \text{ in } C\}} \| W_i - W_j \|$$

$$\text{Межклассовые расстояния (между } C_i \text{ и } C_j): d_b(C_i, C_j) = \max_{\{i \text{ in } C_i, j \text{ in } C_j\}} \| W_i - W_j \|$$

В качестве порога подобия T_c берется минимальное межклассовое расстояние, которое превышает внутриклассовое.

Для добавления узлов SOINN использует схему, при которой вставка происходит в регионе с максимальной ошибкой. А для оценки

полезности вставки используется «радиус ошибки» являющийся оценкой полезности вставки. Эта оценка гарантирует, что вставка приведет к уменьшению ошибки, и контролирует прирост узлов, а в конце концов стабилизирует их количество.

Для проведения экспериментов с рассматриваемым алгоритмом, он был реализован на языке C++ с применением библиотеки Boost Graph Library. В качестве тестовых данных было использовано следующее изображение, представляющее два класса(кольцо и круг). После подачи на вход сети SOINN 20000 случайно взятых точек из этих классов, был сформирован первый слой сети, вид которого показан на рисунке 1.

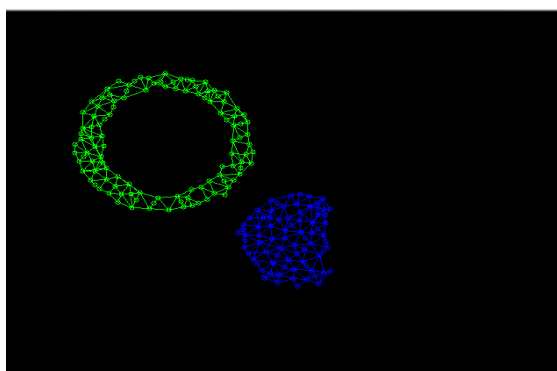


Рисунок 1 – Первый слой сети

На основе данных, полученных на первом слое, был сформирован второй слой, изображенный на рисунке 2.

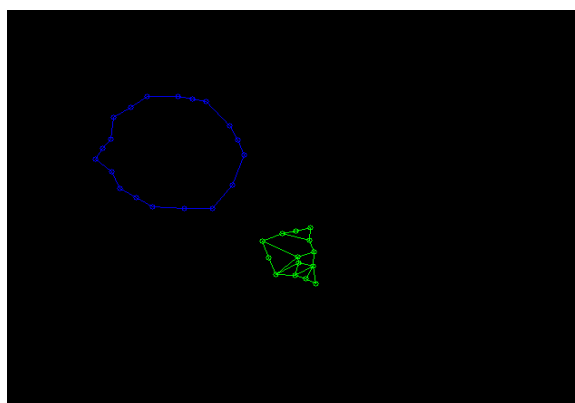


Рисунок 2 – Второй слой сети

На рисунке 3 представлена принципиальная схема работы алгоритма, используемого для обучения и первого и второго слоев сети.

Из результатов алгоритма обучения без учителя, основанного на топологизации данных, полученных экспериментальным путем, можно сделать вывод о справедливости применения данного метода.

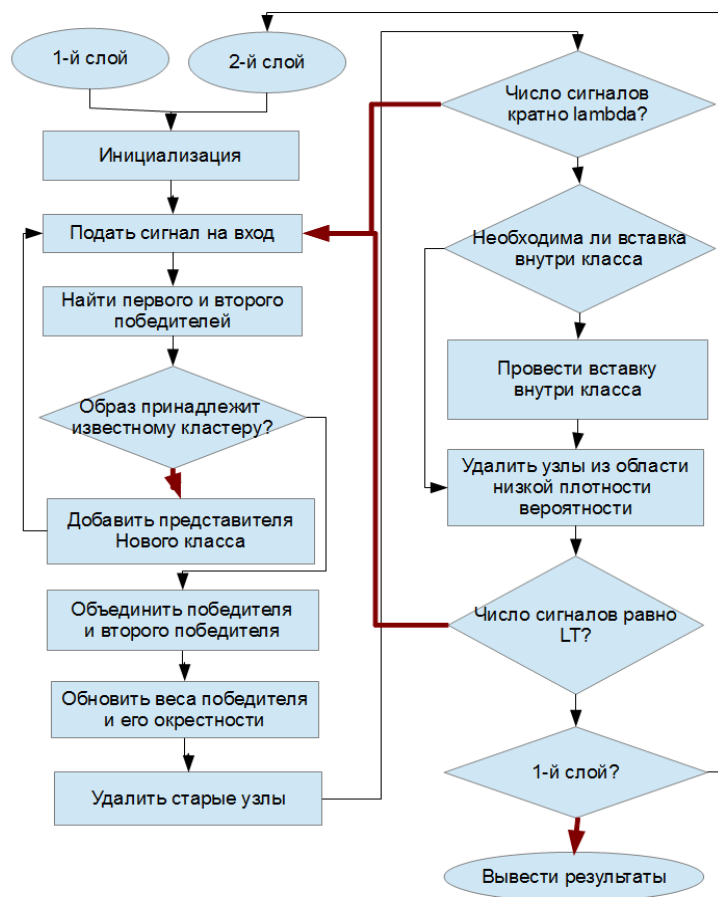


Рисунок 3 – Принципиальная схема работы алгоритма

Но все же стоит отметить недостатки алгоритма SOINN:

- Поскольку слои сети обучаются последовательно друг за другом, то сложно определить момент, когда стоит остановить обучение первого слоя и приступить к обучению второго слоя.
- При изменениях первого слоя сети необходимо полностью переобучать второй слой. В связи с этим не решена проблема онлайн-обучения.
- В сети есть много параметров, которые необходимо подбирать вручную.

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

НОВЫЕ СИСТЕМЫ ХРАНЕНИЯ FUJITSU ETERNUS DX

Новые системы Fujitsu Eternus DX S3, которые можно увидеть на рисунке 1, представляют собой сочетание архитектуры и автоматизированного регулирования качества обслуживания. Решения позволяют

обеспечить соответствие уровня хранения информации современным бизнес-потребностям, повышая коэффициент использования ресурсов системы до 90%, а также увеличивая возможности их консолидации и ускоряя окупаемость инвестиций.



Рисунок 1 – Fujitsu Eternus DX S3

Линейка систем Eternus DX третьего поколения включает в себя четыре устройства: масштабируемые модели начального уровня Eternus DX100 S3 и DX200 S3, а также модели среднего класса Eternus DX500 S3 и DX600 S3. Все устройства модельного ряда представляют собой унифицированные системы с единым пакетом управляющего программного обеспечения Eternus SF. Это позволяет реализовать концепцию единого семейства, тем самым снижая эксплуатационные расходы и затраты на миграцию систем.

Такой подход дает возможность гарантировать высокую защищенность инвестиций. Так, когда при использовании в качестве консолидированного хранилища системы начального уровня масштабируемости и производительности становится недостаточно, можно реализовать массив старшего уровня на тех же дисках без миграции данных, просто заменив контроллер.

Стоит также отметить, что разработчики приложили максимум усилий, чтобы новые системы стали полностью унифицированным хранилищем. Доступ к данным в устройствах обеспечивается как по блочному протоколу через Fibre-Channel, FCoE, iSCSI, SAS и Infiniband, так и через NFS. Нет необходимости в установке дополнительных шлюзов, как это делают некоторые производители. В случае реализации в дальнейшем различных систем репликации или повышения надежности уже существующие лицензии для блочного доступа будут работать и для файлового механизма. Помимо этого, Eternus DX имеют возможность подключения к сетям SAN и NAS.

Важным аспектом в новой линейке систем хранения Fujitsu Eternus DX является использование для подключения дисковых полок интерфейса SAS3. SAS3 вдвое производительней, чем его предыдущая версия, что в итоге существенно повышает общую производительность системы, многократно увеличивая скорость обращения к дискам.

Также стоит отметить и использование в новых моделях дисковых полок более высокой плотности, вмещающих до 60 жестких дисков. Такие полки и переход на более скоростной интерфейс подключения позволили снять ограничения на количество подключаемых SSD-дисков. При этом были оставлены дисковые полки на 2,5 и 3,5 дюйма. Все это дает возможность в любой последовательности подключать диски SAS, NL-SAS и SSD.

Ряд серьезных улучшений реализован и в области кэш-памяти новых систем хранения. Так, в качестве дополнительно кэша используется дополнительный твердотельный накопитель объемом более 5,5 Тб. При работе с большими объемами данных такой подход позволяет «закрыть» около 80 % информации за счет одной лишь кэш-памяти. Фактически, такой объем кэша позволяет целиком поместить в него, например, базу данных.

Еще одной изюминкой, характерной для семейства систем хранения Fujitsu Eternus DX S3, является использование ОС VxWorks, работающей внутри дисковых массивов. Она представляет собой операционную систему жесткого реального времени, применяемую в устройствах с повышенными требованиями к производительности и безопасности. Это современная многопоточная 64-разрядная архитектура, которая в совокупности с новыми многоядерными серверными процессорами Intel позволяет реализовать высокие показатели производительности системы.

Операционная система VxWorks имеет ряд ключевых преимуществ:

- ресурсоемкость;
- поддержка многопроцессорности и многоядерных процессоров;
- расширенная поддержка сетей TCP/IP (IPv4, IPv6);
- интеграция со SCADA-приложениями на базе Windows и промышленными сетями;
- поддержка веб-сервисов (XML, SOAP, WSDL);
- функции управления энергопотреблением;
- поддержка баз данных.

Использование в новых системах хранения Fujitsu Eternus DX новых процессоров Intel, оптимизированных для многопроцессорной и многопоточной обработки, контроллеров с большим объемом кэш-памяти и дополнительным кэшем на базе флэш-накопителей, а также интерфейса SAS3 для подключения жестких дисков и сетевых подключений Fibre

Channel со скоростью 16 Гб/с позволило обеспечить пятикратный рост производительности устройств по сравнению с моделями предыдущей серии. Стоит отметить и возможность интеллектуального управления данными по приоритетам – системы могут сами определять последовательность создания резервных копий.

Помимо увеличения в пять раз числа выполняемых операций ввода-вывода, также отмечается трехкратный рост полосы пропускания и возросшая в два раза пропускная способность шины. Также пятикратно увеличилась плотность виртуализации. В частности, теперь задачи по созданию моментальных снимков можно просто перенести на дисковую систему.

Какие можно сделать выводы об системах Fujitsu Eternus DX? Главное:

1. Революционная архитектура обеспечивает пятикратное увеличение производительности и повышает коэффициент использования ресурсов до 90%;

2. Автоматизация процессов управления уровнем услуг позволяет увязать производительность системы хранения с приоритетными задачами бизнеса;

3. Органично интегрированный доступ к блокам/файлам повышает возможности консолидации.

И.С. Жук (УО БрГТУ, Брест)

Науч. рук. **В.И. Хвещук**, канд. тех. наук, доцент

СРЕДСТВА АВТОМАТИЗАЦИИ ДОКУМЕНТООБОРОТА ДЛЯ КОНФЕРЕНЦИЙ

Одной из важнейших задач применения Интернет-технологий является автоматизация документооборота при организации и проведении таких мероприятий как конференция. Это мероприятие связано с достаточно трудоемким документооборотом: рассылка информационных сообщений о конференции; прием материалов на конференцию; рассылка приглашений на конференцию; рецензирование материалов; формирование программы конференции и другие.

В рамках данной работы предложен результат реализации информационной системы (ИС) для организации и проведения конференций, реализованной с использованием Интернет-технологий.

Структура ИС выполнена в виде клиент-серверной архитектуры. При создании приложения был использован паттерн проектирования MVC (ModelViewControl), так как он делает уровни приложения менее

зависимыми друг от друга. Например, изменение базы данных (Model) не сильно повлияет на контроллер (Control).

Серверная часть – это реляционная база данных (БД), сервлеты и иные приложения, которые обеспечивают сбор, хранение и воспроизведение информации. Для реализации серверной части использованы язык Java, библиотеки (Spring MVC, Hibernate), фреймворк для автоматизации сборки проектов Maven, контейнер сервлетов Tomcat и другие.

Клиентская часть (сайт) ИС обеспечивает диалоговый интерфейс для пользователей ИС – интернет страницы. Для реализации клиентской части использованы следующие средства: язык программирования JavaScript, язык HTML, CSS, JQuery, технология Ajax.

Возможности созданной системы для Интернет-пользователей обеспечивают следующие сервисы: регистрация участников конференции; пересылка материалов на конференцию и другие. Для организаторов конференций ИС позволяет автоматизировать: переписку с участниками конференции, формирование макета программы конференции; формирование макета сборника материалов конференции и другие.

В данный момент для ИС выполняются испытания. Для эксплуатации ИС разработан комплект необходимой эксплуатационной документации.

Г.С. Жуковец (УО «ГГУ им. Ф. Скорины», Гомель)

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

СРЕДСТВО ПРЕДСТАВЛЕНИЯ УЧЕБНО-МЕТОДИЧЕСКОГО МАТЕРИАЛА В ФОРМАТЕ CHM (ФОРМАТ СПРАВКИ WINDOWS)

Была поставлена задача автоматизация предоставления учащимся ГГУ им. Ф.Скорины мультимедийного учебно-методического материала по дисциплине «Микропроцессоры и микрокомпьютеры», цель проекта – создание chm файла, обеспечивающего эффективное взаимодействие компонентов методического пособия.

Наличие доступного методического пособия является важным аспектом в усвоении материала студентами. Возможность в любое время обратиться к необходимым для выполнения лабораторных занятий знаниям, повышает качество получаемых студентами навыков в конкретной предметной области.

Для удобства использования учебно-методического пособия был выбран формат представления документа в виде справки Microsoft Windows – chm.

HTMLHelp (Microsoft Compressed HTML Help, .chm) – формат файлов контекстной справки, который содержит в себе набор

HTML-страниц, может также включать в себя содержание со ссылками на страницы, предметный указатель, а также базу для полнотекстового поиска по содержимому страниц.

Для просмотра chm-файлов используется стандартное средство просмотра, встроенное во все версии Windows, начиная с Windows 98. Кроме того, существует ряд сторонних программ.

Целесообразность использования данного формата обусловлена тем, что он позволяет представить документ в виде древовидной структуры, которая позволяет представить документ в виде глав и подглав с названиями, что позволяет быстро сориентироваться и выбрать именно ту часть документа, которая необходима в данный момент. Так же это позволяет ознакомиться с содержанием учебно-методического пособия.

Формат справки Windows обеспечивает возможность вставлять в любую часть документа ссылки, при помощи которых можно моментально попасть на интересующую часть текста или перейти на внешний источник информации, например, веб-ресурс.

Также данный формат представления материала поддерживает использование иллюстраций, flash, видео, аудио файлов, таблиц, что позволяет в полном объеме представить учебную информацию.

Преимущества данного формата:

- chm – это стандартный формат операционных систем семейства Microsoft Windows, поэтому он не обязывает использование специальных программ для просмотра;

- chm – это одиночный файл, который содержит в себе все элементы (элементы навигации, текста, графики). Это очень удобно, когда Вы передаете по электронной почте один файл, в котором находится весь интересующий Вас материал;

- размер файла меньше, чем у обычного HTML;

- используются все возможности форматирования, имеющиеся в HTML и CSS;

- возможность полнотекстового поиска;

- возможность просмотра множества chm-файлов как один, с общим содержанием и предметным указателем.

Самыми распространенными инструментами создания chm-файлов являются такие бесплатные средства, как: Microsoft HTML Help Workshop, Htm2Chm, плагины для Total Commander'a и многие другие.

В связи с развитием информационных технологий, немаловажным фактором является поддержка данного формата современными мобильными устройствами (смартфоны, планшеты и т.д.), позволяя в любое время и в удобной форме обратиться к необходимому материалу.

Создание файла в формате chm предусматривает, как правило,

предварительную подготовку исходного материала (в зависимости от программы они могут быть в форматах HTML, DOC и т.д.). Затем создают файл проекта, формируют в нем содержание и индекс и наполняют проект информацией. Завершает всё процесс компиляции, при котором отдельные файлы собираются в единый chm-документ.

Так же можно воспользоваться иным путем и предварительно создать chm файл в одной из специально предназначенных программ и уже после этого наполнять его необходимой информацией.

При возникновении необходимости в редактировании уже созданного файла всегда можно воспользоваться специализированными программными продуктами без повреждения старых данных. Данная функция позволяет обновлять учебно-методическое пособие при первой же необходимости.

Разработка данного учебно-методического пособия в формате chm имеет практическую значимость в подготовке к выполнению лабораторных работ по учебным дисциплинам.

Созданный проект позволяет удовлетворять следующие цели автоматизации:

- повышение эффективности учебного процесса;
- уменьшение размера, занимаемого учебно-методическом пособием на сервере университета;
- увеличение скорости доступа к данным.

Литература

1. HTMLHelp [Электронный ресурс]. – В режиме доступа: <http://ru.wikipedia.org/wiki/HTMLHelp>. – Дата доступа: 18.12.2013
2. Программы для работы с CHM-документами [Электронный ресурс]. – В режиме доступа: <http://compress.ru/article.aspx?id=20503>. – Дата доступа: 18.12.2013
3. Как создать chm-файл? [Электронный ресурс]. – В режиме доступа: <http://www.siteedit.ru/chm/>.

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

Науч. рук. **В.С. Давыдов**, канд. тех. наук, доцент

НАНОПРОВОДНИКОВЫЙ АККУМУЛЯТОР

Проблема хранения электроэнергии для мобильных устройств в наше время состоит в том, что ее нам никогда не хватает, чем и объясняются работы, направленные на создание более энергоемких аккумуляторов. Также следует учесть тот факт, что сегодня создаются все

более и более энергопотребляемые устройства. Разрешение же этой проблемы находится в нанопроводниковом аккумуляторе.

Нанопроводниковый аккумулятор – вид литий-ионного аккумулятора, изобретённый группой под руководством доктора Йи Кю в Стэнфордском университете в 2007 г.. Тогда было заявлено о совершенном прорыве в технологиях, связанных с кремниевыми нанопроводниками. Они позволяют создавать аккумуляторы с емкостью, в 10 раз превышающей таковую в существующих на сегодняшний день литий-ионных батареях. Первоначально проведенные исследования традиционных кремниевых анодов были отвергнуты в связи с тенденцией кремния растрескиваться и увеличиваться в объёме. Такой анод становится неработоспособным, потому что в процессе работы трещины заполняются литием. Нанопроводник не обладает этим недостатком. По словам доктора Кю, аккумуляторы достигали десятикратной плотности заряда при первой зарядке и затем стабилизировались на уровне восьмикратной плотности при последующих зарядках. Поскольку это достигается только за счёт усовершенствования анода, необходимо будет провести эквивалентное изменение катода, чтобы получить максимальное повышение плотности энергии хранения.

Т.е. суть самого изобретения состоит в замене традиционного графитового анода аккумулятора на анод из нержавеющей стали, покрытый кремниевым нанопроводником. Кремний, способный удерживать в 10 раз больше лития чем графит, позволяет создавать значительно большую плотность энергии на аноде, снижая таким образом массу аккумулятора.

Преимущества нанопроводникового аккумулятора состоят в следующем:

1. Более высокая удельная плотность запасённой энергии. Это означает, что при одинаковых геометрических размерах других видов аккумуляторов нанопроводниковые аккумуляторы способны отдавать больше энергии во внешнюю нагрузку.

2. Низкий саморазряд. В отсутствии нагрузки эти аккумуляторы дольше сохраняют накопленную энергию без заметного её снижения.

3. Простота обслуживания. Очень существенный фактор при длительной эксплуатации.

Также следует отметить и его недостатки:

1. Данный аккумулятор опасен для человека при нарушении герметизации его корпуса.

2. Как следствие – малая долговечность аккумуляторов.

3. Недопустимость глубокого разряда, который, как правило, полностью выводит аккумулятор из строя.

4. Нанопроводниковый аккумулятор обладает неприятным свойством – стареть, даже если он не эксплуатируется. Уже через 1,5–2 года он теряет свою ёмкость более чем на 50%.

Итак, проведя небольшое знакомство с литиево-ионными аккумуляторами, мною были рассмотрены изменения в устройстве нанопроводниковых аккумуляторов по сравнению с их прародителями:

1. Анод нанопроводникового аккумулятора изготовлен из нержавеющей стали, лишь сверху покрытый изобретённым недавно нанопроводником. Запасы кремния в Земле огромны: на порядок больше, чем лития.

2. Новейшие технологии изготовления сверхпористого кремния, благодаря чему увеличивается активная поверхность анода, привели к значительному увеличению энергетической плотности.

3. Как следствие, при одинаковых энергетических характеристиках классических литиево-ионных и нанопроводниковых аккумуляторов последние будут иметь меньшую массу, что немаловажно.

4. По сравнению с обычными, литиево-ионными, нанопроводниковые аккумуляторы даже на данной стадии производства обладают в 8–10 раз большей плотностью заряда.

5. Стоимость хранения ватт-часа энергии меньше, чем у классических литиево-ионных аккумуляторов.

6. Количество циклов заряд-разряд у нанопроводниковых аккумуляторов уже приближается к тысяче, что для других типов аккумуляторов недостижимо.

Из всего вышеизложенного можно сделать вывод, что нанопроводниковые аккумуляторы – это оптимальный вариант для хранения электроэнергии в ближайшем будущем.

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

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

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ КАК МЕТОД ИССЛЕДОВАНИЯ СЛОЖНЫХ СИСТЕМ

Основным методом, который используется для исследования сложных систем, является метод моделирования – способ изучения объекта через рассмотрение подобного ему и более простого объекта, т.е. его модели.

Имитационное моделирование состоит в исследовании системы с помощью компьютерных (вычислительных) экспериментов на имитационной модели – программе, позволяющей имитировать на

компьютере поведение отдельных элементов системы и связи между ними в течение заданного времени моделирования.

Имитационный метод наиболее эффективен для исследования сложных стохастических систем, на функционирование которых существенное влияние оказывают случайные факторы. Результат одного эксперимента на имитационной модели может рассматриваться лишь как точечная оценка истинных характеристик системы. Требуется проведение большого числа экспериментов и статистическая обработка их результатов. Также имитационное моделирование позволяет имитировать поведение системы во времени. Причём плюсом является то, что временем в модели можно управлять: замедлять в случае с быстропротекающими процессами и ускорять для моделирования систем с медленной изменчивостью.

Имитационное моделирование позволяет рассматривать процессы, происходящие в системе, практически на любом уровне детализации. В имитационной модели можно реализовать практически любой алгоритм управленческой деятельности или поведения системы. Кроме того, модели, которые допускают исследование аналитическими методами, также могут анализироваться имитационными методами. Все это служит причиной того, что имитационные методы моделирования в настоящее время становятся основными методами исследования сложных систем.

Одно из наиболее важных решений, которые приходится принимать разработчику имитационных моделей, касается выбора программного обеспечения. Если программное обеспечение недостаточно гибко или с ним сложно работать, то имитация может дать неправильные результаты или будет вообще невыполнима.

Программное обеспечение, используемое для создания имитационных моделей, можно условно разделить на *универсальные языки моделирования* и *системы имитационного моделирования*, которые имеют несколько преимуществ перед первыми:

1 Они автоматически предоставляют функциональные возможности, которые требуются для создания имитационных моделей:

- генераторы случайных чисел;
- продвижение модельного времени;
- добавление и удаление записей из списка событий;
- сбор выходных статистических данных и создание отчета с результатами;
- и другое.

Это позволяет сократить время, требуемое для программирования и общую стоимость проекта.

2 Основные конструкции систем имитационного моделирования больше подходят для создания имитационных моделей, чем конструкции универсальных языков программирования (естественная среда моделирования).

3 Системы имитационного моделирования обеспечивают более совершенный механизм обнаружения ошибок имитации.

Литература

1. Лоу, А. Имитационное моделирование / А. Лоу, В. Кельтон. – СПб. : Питер, 2004.
2. Рыжиков, Ю.И. Имитационное моделирование. Теория и технологии / Ю.И. Рыжиков. – СПб : Корона принт, 2004.
3. Советов, Б.Я. Моделирование систем : практикум / Б.Я. Советов, С.А. Яковлев. – М. : Высш. шк., 2005.
4. Шрайбер, Т. Дж. Моделирование на GPSS/ Т.Дж. Шрайбер. – М.: Машиностроение, 1980.
5. Харин Ю.С. Основы имитационного и статистического моделирования. Учебное пособие/ Ю.С. Харин, В.И. Малюгин, В.П. Кирлица и др. – Мн.:Дизайн ПРО, 1997.
6. Кудрявцев Е.М. GPSS World. Основы имитационного моделирования различных систем/ Е.М. Кудрявцев. – М.: ДМК, 2004.
7. Максимей И.В. Имитационное моделирование на ЭВМ/ И.В. Максимей. – М.: Радио и связь, 1988.

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

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

ЗАДАЧИ ВВОДА, ХРАНЕНИЯ И ОБРАБОТКИ ИНФОРМАЦИИ НА ПРЕДПРИЯТИИ

В последние десятилетия огромное применение приобрели компьютеры в повседневной жизни каждого учебного заведения, предприятия, а собственно и каждого из нас. То есть, использование персонального компьютера стало неотъемлемой частью рабочего процесса.

С помощью компьютеров можно в значительной степени упростить и облегчить множество рабочих процессов. В свою очередь, автоматизация многих процессов с помощью компьютеров позволяет экономить значительное количество времени и средств. В не меньшей степени многие процессы ещё более облегчаются при использовании систем управления базами данных (СУБД), которые предназначены специально для работы с большими объёмами информации.

В представленной работе рассматривается разработка информационного и программного обеспечения для ввода, автоматизации хранения и обработки информации при организации работы предприятия нефтепромышленного 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 обработкой данных. Интерфейс должен быть ориентирован на пользователя, не имеющего специальной подготовки.

Разработанное приложение «Подсистема по ведению и учёту расходов денежных средств предприятия, проходящего через финансовый отдел» производственного объединения РУП «Белоруснефть» будет использоваться как независимо от других программных комплексов приложение отдела.

В приложении решаются задачи автоматизации всей приходной финансовой деятельности гомельского офиса производственного объединения. Подсистема используется параллельно с банковскими клиентскими программами, так как соответствующая информация ежедневно передаётся в отделения банков, с которыми работает объединение. В комплексе используются данные обо всех расходах предприятия, которые ведутся или поступают из других подразделений и обрабатываются финансовым отделом.

Предприятие РУП ПО «Белоруснефть» имеет весьма сложную структуру, большое количество отделов и управлений, поэтому приложение рассчитано на интенсивную многопользовательскую работу, и помимо собственно быстрого действия, 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 по финансовым расходам, отражающим специфику деятельности предприятия, расчёт и работа с документами. Разрешается обработка произвольного количества документов за указанный период; данные сохраняются

с пометкой о том, кто и когда создал или изменил их. Допускается обработка данных за прошедшие месяцы при условии, если документы ещё не заблокированы.

Основные операции: возможность подведения итогов по расходам за любой указанный период, 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 внутренних служб, отделов и управлений.

Е.И. Зинчук (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **А.В. Воруев**, канд. тех. наук, доцент

РЕШЕНИЕ ЗАДАЧИ ПРЕОБРАЗОВАНИЯ АДРЕСОВ NAT44 НА ИНТЕЛЛЕКТУАЛЬНЫХ СЕТЕВЫХ УСТРОЙСТВАХ

NAT (Network Address Translation) – трансляция сетевых адресов. Процедура по изменению адресов в заголовках IP-пакетов при их прохождении через маршрутизатор или другое устройство.

В современных условиях преобразование адресов ведется между схемами адресации IPv6 и IPv4 (NAT64 и NAT46 – в зависимости от направления), но наиболее актуальной формой преобразования является private/public в схеме адресации IPv4 (NAT44).

Для организации процедуры NAT необходимо настроить интеллектуальное сетевое устройство, выполняющее функции третьего уровня модели OSI. Чаще всего для этого используются маршрутизаторы.

Для примера можно использовать незавершенную сетевую структуру (рисунок 1).

Так как топология сети готова и преднастройки уже введены, нам необходимо только настроить NAT на маршрутизаторах S1R1 и S2R1.

Шаг 1: Указываем, что транслировать. Для этого создаем access-list:

```
Router(config)#access-list 100 permit ip 192.168.10.0 0.0.0.255 any
```

Шаг 2: Создаем пул из адресов, указывая стартовый и конечный адрес:

```
Router(config)#ip nat pool NAME_OF_POOL 192.168.1.3 192.168.1.20  
netmask 255.255.255.0
```

Шаг 3: Маркируем интерфейсы:

```
Router(config)#int fa 0/0
```

```
Router(config-if)#ip nat inside
```

```
Router(config-if)#ex
```

```
Router(config)#int fa 0/1
```

```
Router(config-if)#ip nat outside
```

```
Router(config-if)#ex
```

Шаг 4: И создаем трансляцию:

```
Router(config)#ip nat inside source list 1 pool NAME_OF_POOL
```

Теперь посмотрим статистику NAT:

```
Router#show ip nat statistics
```

```
Total translations: 0 (0 static, 0 dynamic, 0 extended)
```

```
Outside Interfaces: FastEthernet0/1
```

```
Inside Interfaces: FastEthernet0/0
```

```
Hits: 0 Misses: 0
```

```
Expired translations: 0
```

```
Dynamic mappings:
```

```
-- Inside Source
```

```
access-list 1 pool NAME_OF_POOL refCount 0
```

```
pool NAME_OF_POOL: netmask 255.255.255.0
```

```
start 192.168.1.3 end 192.168.1.20
```

```
type generic, total addresses 18 , allocated 0 (0%), misses 0
```

Аналогичным образом настраиваем NAT на втором маршрутизаторе.

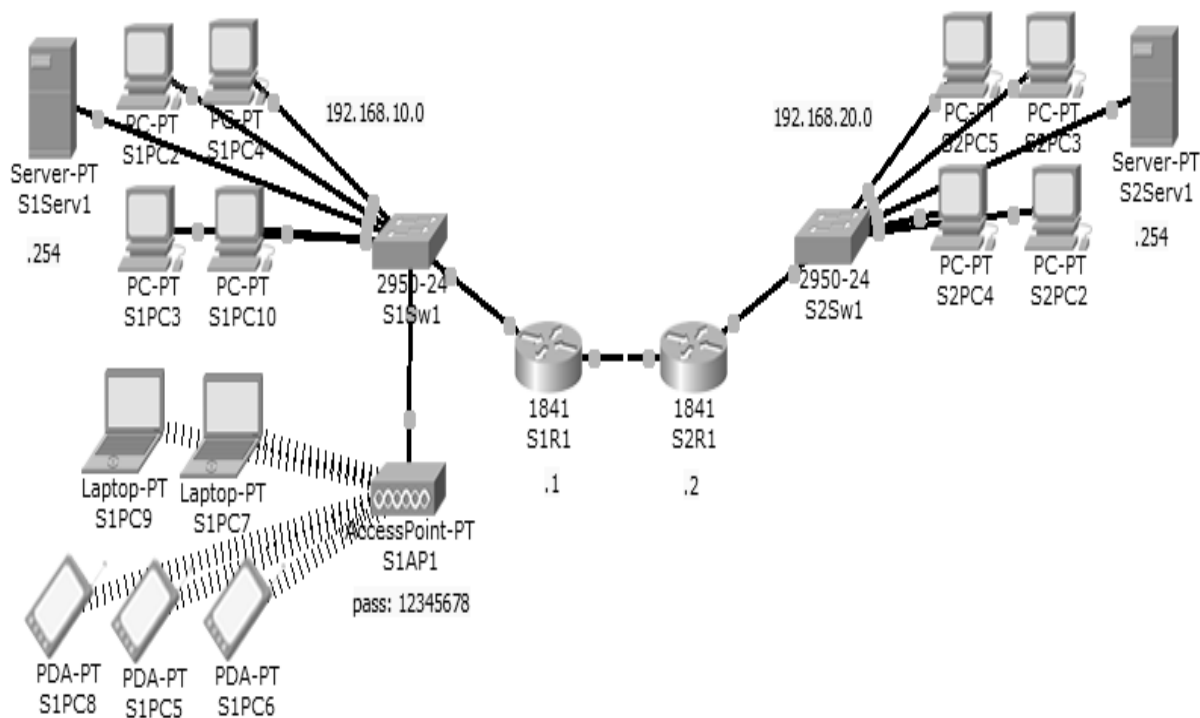


Рисунок 1 – Исходная сеть

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

Науч. рук. **П.Л. Чечет**, канд. тех. наук, доцент

АВТОМАТИЗАЦИЯ ДЕЯТЕЛЬНОСТИ ВЕТЕРИНАРНОЙ ЛАБОРАТОРИИ

Удельный вес биохимических исследований в диагностике различных заболеваний увеличивается с каждым годом. В основном, это касается диагностики заболеваний человека. Однако в практике ветеринарных врачей биохимические исследования пока еще не получили достаточного распространения. Между тем, в зарубежных руководствах по клинической биохимии домашних животных этим вопросам уделено значительное внимание.

Для автоматизации деятельности ветеринарной лаборатории необходимо обобщить имеющиеся литературные данные по относительным нормам биохимических показателей биологических жидкостей кошек и собак, используемых при лабораторной диагностике заболеваний этих животных. Требуется все данные свести в таблицы и сопроводить кратким описанием клинического значения отклонения того или иного показателя от нормы в плане обменных нарушений при различных патологических состояниях, не претендующее на биохимическую оценку каждого конкретного заболевания.

Также дополнительно к данным по биохимическим показателям, нужно рассмотреть данные по влиянию различных факторов, которые могут исказить результаты биохимических исследований и тем самым приводить к ошибочным выводам.

Сегодня в мире насчитывается свыше 700 пород домашних собак. Процесс урбанизации вызывает возрастание потребности человека в общении с собаками и кошками «братьями меньшими». На основе генетической изменчивости созданы разнообразные породы собак, которые доставляют эстетическое наслаждение человеку.

В случае болезни домашних собак и кошек, у их владельцев всегда возникают эмоциональные (часто и физические) расстройства. Поэтому, конечно, крайне важно знать и с точки зрения здоровья человека физиологобиохимические основы жизнедеятельности его питомцев.

Биохимические показатели плазмы крови сведены в таблицу 1. С развитием средств доступа к глобальной сети, распространением мобильной связи, актуальным становится разработка сайта, как средства автоматизации деятельности организации. Хранение данных в таком случае можно возложить на серверную базу данных MySQL, а обработку данных – на динамический язык программирования PHP.

Таблица 1 – Биохимические показатели плазмы крови

Показатель	ЕД. измерения	Значение
Глюкоза	ммоль/л	4,49
Мочевина	ммоль/л	48
Креатинин	мкмоль/л	70180
Билирубин общ.	мкмоль/л	0,78
Мочевая кислота	мкмоль/л	955
Общие липиды	г/л	715
Общий холестерин	ммоль/л	3,57,5
Общий белок	г/л	50100
Альбумины	г/л	2040
Глобулины	г/л	3,59,5
Кальций	ммоль/л	2,33
Фосфор	ммоль/л	0,81,9
Натрий	ммоль/л	139153
Калий	ммоль/л	3,65,2
Бикарбонаты	ммоль/л	1430
Магний	ммоль/л	0,61,2
АСТ	МЕ/л	1075
АЛТ	МЕ/л	460
Щелочная фосфатаза	МЕ/л	826
Кислая фосфатаза	МЕ/л	18

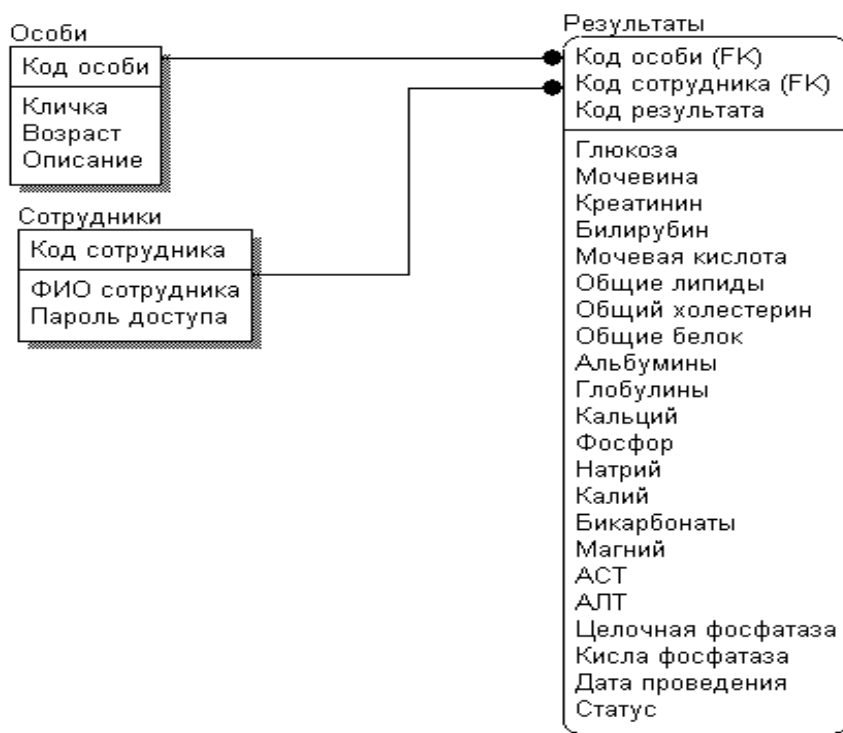


Рисунок 1 – Структура базы данных

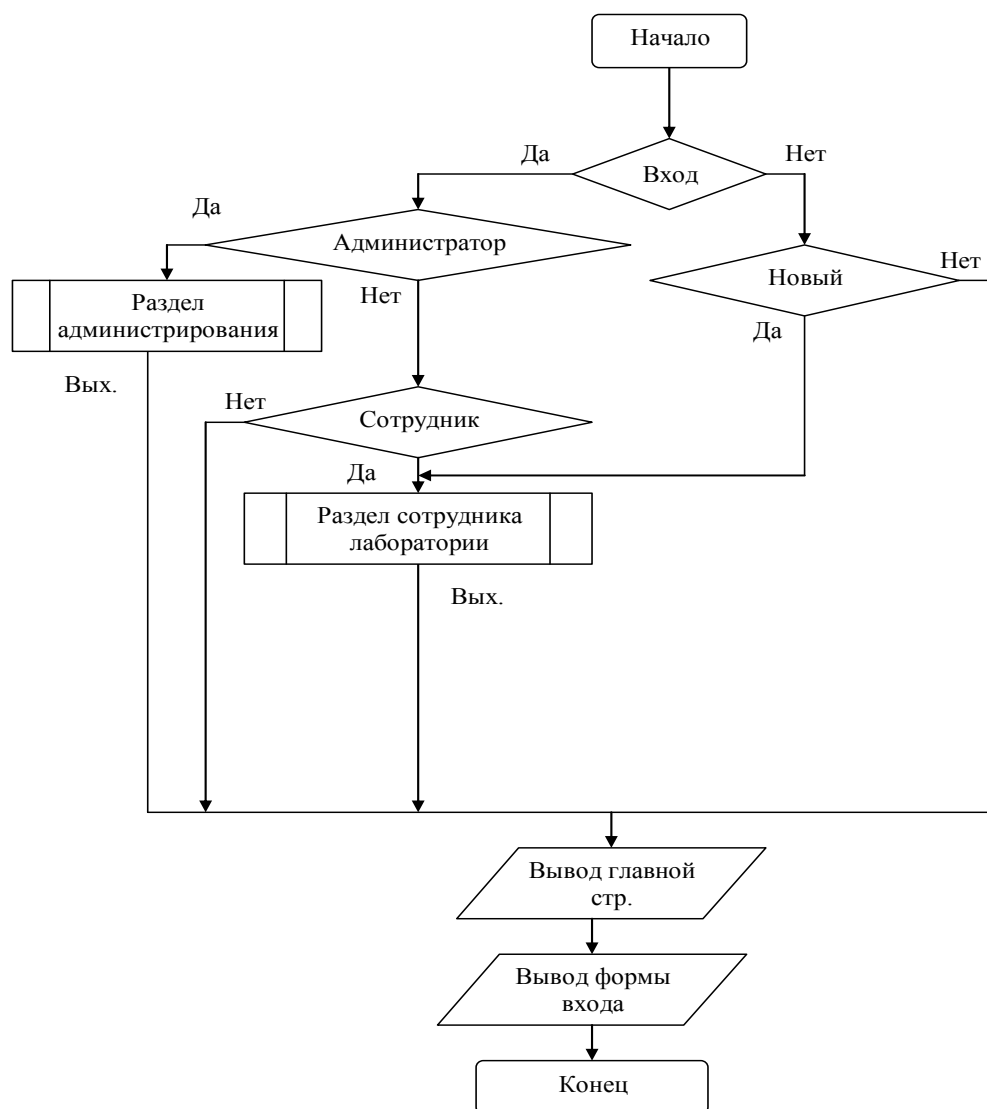


Рисунок 2 – Логическая структура сайта

Каждый результат исследований привязывается к конкретной особи, которая регистрируется администратором. Для особи необходимо учитывать следующие атрибуты: кличка, возраст и описание.

При автоматизации деятельности лаборатории необходимо также вести учет сведений о сотрудниках (личные данные, пароль доступа к защищенному режиму). Структура базы данных показана на рисунке 1.

На рисунке 2 представлена упрощённая схема функционирования разрабатываемого сайта ветеринарной лаборатории.

Использование современных сетевых Интернет технологий позволяет повысить эффективность работы ветеринарной лаборатории, а так же упрощает обработку данных, которые теперь хранятся в электронной форме в базе данных.

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

Науч. рук. П.Л. Чечет, канд. тех. наук, доцент

АВТОМАТИЗАЦИЯ УЧЕТА ПРОМЫШЛЕННОГО ОБОРУДОВАНИЯ, ПОДДЕРЖИВАЮЩЕГО РАБОТУ ПО ПРОТОКОЛУ TCP/IP V4

Системы дистанционного мониторинга процессов (телеметрия, теле-сигнализация) и дистанционного управления (телеуправление) – явление давнее, по крайней мере, в военно-промышленной сфере. На протяжении всего существования этих систем наблюдались сложности с каналами передачи информации к центру обработки. В системах, находящихся на одном предприятии, использовались проводные каналы, в масштабных военных комплексах, например ракетных, – радиоканалы. И те и другие решения никогда не были дешевыми.

В последнее время ситуация во многом изменилась благодаря тому, что широкое распространение получили проводные высокоскоростные сети и беспроводные технологии, прежде всего радиосвязь стандарта GSM, поддерживающие работу по протоколу IP.

К настоящему времени в энергосистеме страны происходит постоянное обновление промышленного оборудования (промышленных устройств) участвующего в производстве, преобразовании, трансформации, передачи, распределения, потребления электрической энергии и преобразовании её в другой вид энергии, – общее количество которого на одном объекте (подстанции) может достигать 100 и более (шт).

Промышленное оборудование (устройство) поддерживающие работу по протоколу IP (например, IPv4) выступает как клиент внешней сети, вследствие чего присваивается IP-адрес (постоянный или динамический). IP-адрес (англ. Internet Protocol Address) – уникальный сетевой адрес узла компьютерной сети, построенной по протоколу IP.

Но эта информация на предприятии хранится в электронном виде без использования баз данных. Как результат, на поиск необходимого IP-адреса или «сетевой» информации на конкретное промышленное оборудование (устройство) приходится потратить немало времени. Поэтому возникла идея создания автоматизированной системы учета промышленного оборудования, поддерживающего работу по протоколу IPv4.

Базы данных – это часть информационных систем – программно-аппаратных комплексов, осуществляющих хранение и обработку огромных информационных массивов.

Для удобства использования разработанной базой данных, было принято решение реализовать GUI-интерфейс (англ. Graphical User Interface, русск. Графический интерфейс пользователя), позволяющий

пользователю уменьшить время на создание запросов выборки и получение результата их выполнения. Для выполнения выше поставленной цели были использованы: Microsoft Access 2003 для создания БД, и среда EmbarcaderoDelphiXE5 для создания GUI-интерфейса.

Применение вышеуказанных технологий позволило получить эффективное и удобное GUI-приложение для работы с базой данных учета промышленного оборудования, поддерживающего работу по протоколу IPv4.

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

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

РОЛЬ ЛАЗЕРНОЙ ТРЁХМЕРНОЙ ПЕЧАТИ В СОЗДАНИИ НОВОГО ЭКСПЕРИМЕНТАЛЬНОГО МАТЕРИАЛА

Используемые материалы всегда определяли уровень прогресса человечества, подтверждением этому служат такие известные термины, как «Каменный век» и «Бронзовый век». А сейчас человечество находится в периоде, который можно охарактеризовать, как «Молекулярный век», ведь ученые уже имеют возможность контролировать вещества на атомарном, молекулярном уровне и создавать новые материалы, обладающие поразительными свойствами. В этом направлении работает группа из технологического института Карлсруэ (Karlsruhe Institute of Technology, KIT), возглавляемая Йенсом Бауэром (Jens Bauer), которая разработала и изготовила при помощи технологии лазерной трехмерной печати новый материал, плотность которого меньше плотности воды, но его прочность превышает прочность некоторых сортов стали. «Наш материал является первым экспериментальным доказательством того, что такие материалы могут существовать и их можно производить» – рассказывает Бауэр.

Все известные на сегодняшний день материалы могут быть представлены в виде одной диаграммы, осями которой являются плотность и прочность материала (рисунок 1).

Средняя точка оси X, имеющая значение 1000 кг/м³, соответствует плотности воды, все материалы, находящиеся слева от этой точки, легче чем вода, и как можно увидеть, что существует достаточно мало плотных однородных материалов, плотность которых меньше плотности воды. Большинство таких материалов являются пористыми материалами, такими как дерево или костные ткани. Внешне такие материалы кажутся однородными, но при рассмотрении через микроскоп можно увидеть то, что они состоят из крошечных структур, между которыми существуют пустые промежутки.

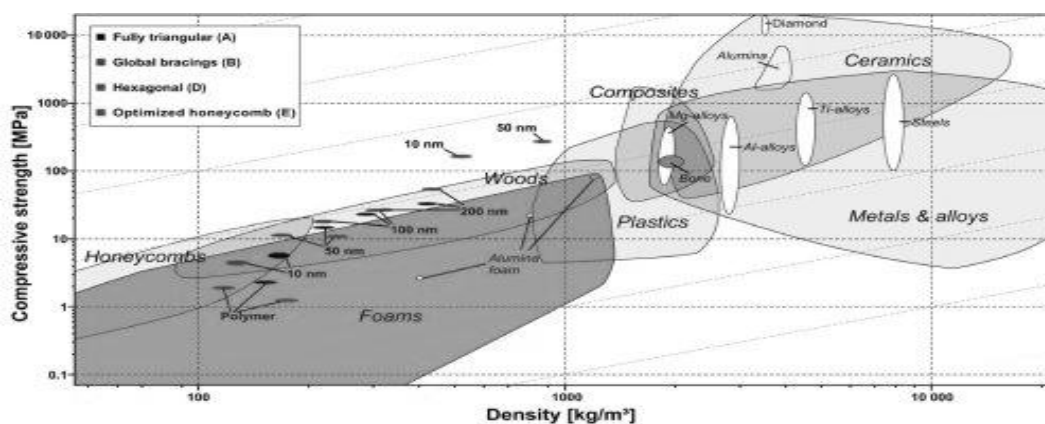


Рисунок 1 – Диаграмма плотности и прочности материалов

Уже достаточно долгое время ученым, при помощи расчетов и компьютерных моделей, было известно о возможности создания прочных и легких материалов, структура которых состоит из крошечных элементов, размеры которых сопоставимы с диаметром человеческого волоса. Но до последнего времени не было технической возможности создать такие материалы в реальности. И лишь не так давно немецкая компания под названием Nanoscribe создала лазерные системы, которые можно использовать для производства микроструктурированных материалов. Система Nanoscribe использует гелеобразный полимер, который полимеризуется в свете лазера, луч которого может быть сфокусирован в любой точке пространства.

Однако, чисто полимерные материалы, изготовленные при помощи технологии Nanoscribe, не столь прочны, как этого хотелось бы. Поэтому Бауэр и его группа при помощи специальной технологии нанесли на поверхности структуры материала 50-нанометровый слой окиси алюминия и корунда. Этот слой, конечно, увеличил значение плотности материала, но все же значение плотности осталось ниже значения плотности воды. При этом материал может выдерживать внешнюю нагрузку, оказывающую давление 280 МПа, давление, которое не выдерживают некоторые сорта стали.

Однако у данной технологии еще существует ряд ограничений, система Nanoscribe способна производить объекты, размерами всего в несколько десятков микрон, а будущие варианты установок будут способны производить объекты, размеры которых будут исчисляться единицами миллиметров. Естественно, возможностей и первого и второго вариантов недостаточно для любого реального применения вышеописанной технологии.

Однако, такое положение вещей с учетом нынешних темпов развития технологий, таких как лазерная трехмерная печать и новые полимерные материалы, будет сохраняться еще не очень долго. И в недалеком будущем новые легковесные, но невероятно прочные материалы можно будет найти в самых разных местах, начиная от лыж и заканчивая плоскостями крыльев летательных аппаратов.

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

Науч. рук. **В.Д. Левчук**, канд. тех. наук, доцент

ТЕСТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ ВОЗМОЖНОСТЕЙ БИЗНЕС-ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ ВЕБ-ХОСТИНГОМ

Все виды тестирования программного обеспечения, в зависимости от преследуемых целей, можно условно разделить на следующие группы:

1. Функциональные;
2. Нефункциональные;
3. Связанные с изменениями.

В бизнес-приложении Parallels Business Automation используется функциональное тестирование, а именно компонентное или модульное тестирование. Задача заключается в проверке функциональности и поиске дефектов в частях приложения. Например, в таких, как создание регистратора, отвечающего за доменные зоны, которые будут предоставлены тем или иным доменным плагином для последующего использования их в сервисных планах. Тестирование отдельных частей приложения жизненно важно, т.к. если выйдет из строя одна часть функционала, то она может повлечь за собой отключение какой-либо другой связанной с ней частью. В случае с регистратором это невозможность создать сервисный темплит для создания на его основе сервисного плана, чтобы в дальнейшем его продавать.

Тестирование безопасности используется для проверки безопасности системы, а также для анализа рисков, связанных с обеспечением целостного подхода к защите приложения, атак хакеров, вирусов, несанкционированного доступа к конфиденциальным данным. Общая стратегия безопасности основывается на трех основных принципах: конфиденциальность, целостность, доступность. При создании пользователя в провайдерской панели необходимо следить за безопасностью. Вводимый или генерируемый кнопкой «Generate» пароль не должен попасть в лог сессии или же каким-либо другим способом отобразиться в доступном для злоумышленников месте. Поэтому при тестировании безопасности необходимо это учитывать и проверять после ввода пароля и создания пользователя самые уязвимые места в системе.

Тестирование взаимодействия заключается в проверке способности приложения взаимодействовать с одним и более компонентами или системами. Оно включает в себя тестирование совместимости и интеграционное тестирование. При добавлении каких-либо новых свойств необходимо убедиться, что добавление нового функционала не оказало никакого эффекта в работе остальной части приложений. Также при

взаимодействии некоторого функционала между собой нужно проверять как стандартные сценарии для нового и уже имеющего функционала, так и добавлять комбинированные тесты для полной оценки его работы со всех возможных сторон.

М.Н. Коваленко (УО «ГГУ им. Ф. Скорины», Гомель)

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

МНОГОКАМЕРНАЯ ВИДЕОАНАЛИТИКА

Видеоаналитическое ПО стало почти обязательным компонентом современной системы охранного видеонаблюдения. Интеллектуальные видеодетекторы позволяют многократно повысить продуктивность службы охраны и снизить психологическую нагрузку операторов ситуационных центров. Многокамерное сопровождение – следующий виток развития видеоаналитической технологии и концепции охранной системы в целом.

Ключевая функция системы многокамерного сопровождения состоит в автоматической регистрации траекторий движения физических объектов на обширном пространстве, контролируемом множеством телекамер. Если при однокамерном сопровождении возможна независимая обработка данных по каждому каналу, то многокамерная система должна анализировать все каналы интегрально. Многокамерная система учитывает трехмерную геометрию пространства и взаимное расположение телекамер, а также делает рациональные предположения о наиболее вероятной траектории движения объекта, даже если он временно выходит из объединенной зоны контроля всех камер. Задача существенно усложняется в случае, когда сопровождаемые объекты многочисленны и плохо различимы или, наоборот, плохо сопоставимы из-за своей изменчивости, различных ракурсов наблюдений и условий регистрации изображения.

Технологии многокамерного сопровождения в первую очередь востребованы в таких отраслях, как охрана, безопасность, транспорт, маркетинг розничных продаж и интерактивная реклама.

Многокамерная система имеет два режима: интерактивный и автоматический. При интерактивном режиме сотрудник службы безопасности может выделить субъекта, действия которого являются подозрительными, на дисплее, отображающем видео с одной из камер, и мгновенно получить траекторию движения этой личности на трехмерной модели здания. При автоматическом режиме сигнал о подозрительном поведении может быть автоматически сформулирован на основе правил о штатных моделях движения человека или группы по контролируемому объекту с учётом текущего времени и режима работы. Примерами

простых правил являются: отклонение от стандартного маршрута пассажира, пребывание свыше указанного времени, перемещение с повышенной скоростью, повторное появление.

Правила могут быть подобраны оптимальным образом на основе статистических данных о перемещении людей, собранных на любом промежутке времени. Возможно полностью автоматическое обучение системы «паттернам» поведения для последующего обнаружения нестандартных перемещений. Для транспортной отрасли многокамерные системы позволяют отслеживать поведение пассажиров и точно измерять загрузку каждого направления на различных участках. В области маркетинга и розничных продаж многокамерные системы позволяют исследовать поведение покупателей в торговых центрах и супермаркетах. В случае интеграции системы сопровождения и системы транслирования рекламы на цифровых панелях появляется возможность оценивать эффект рекламы на индивидуальных покупателей, а также показывать посетителю взаимосвязанные видеоматериалы в процессе его перемещения по торговому центру.

На уровне однокамерного сопровождения возникает проблема сегментирования людей в плотном потоке на различном масштабе. Так, алгоритмы машинного зрения часто «ошибаются» на этапе отделения объекта от фона (например, когда люди находятся близко друг к другу, частично или полностью перекрываются, перестают двигаться, выглядят нестандартно). В то время как разнообразие внешнего вида и поведения человека безгранично, простые отклонения от модели (скажем, перемещение на инвалидном кресле или раскрытие газеты) могут «ввести» компьютерную систему в «заблуждение».

Люди появляются на различном расстоянии от камеры, следовательно, фрагменты изображения имеют различную детализацию и информативность. Зачастую трудно обеспечить однородное освещение на большом пространстве. Эти факторы существенно повышают вычислительную сложность алгоритма обработки видео. Дополнительную неопределенность создают физические преграды сцены, ограничивающие обзор камеры, такие как колонны и ларьки. Если программа «теряет» объект в какой-то момент времени, то происходит разрыв траектории и утрачивается возможность проследить движение объекта от начальной до конечной точки.

Сопровождаемый объект может наблюдаться под различными ракурсами, на различном расстоянии и в любом состоянии (сидит, стоит, идет, бежит). Разные типы освещения (естественное, искусственное) затрудняют использование цветовых признаков, поскольку они существенно зависят от спектра излучения осветителя. Подобные факторы приводят к нестабильности численных значений признаков и к ошибкам

сопоставления объектов при переходе из зоны действия одной камеры к другой. Точная калибровка камеры в трехмерном пространстве контролируемого объекта позволяет повысить эффективность системы многокамерного сопровождения, особенно если области наблюдения камер перекрываются. На практике численные данные о местоположении и ориентации телекамер недоступны, и калибровка системы становится сложной и трудоемкой процедурой. Чаще всего она производится ручным сопоставлением трехмерной модели и двухмерных изображений, поступающих с камеры. Возможна частичная или полная автоматизация, когда алгоритмы регистрации сопоставляют ракурсы по узловым точкам и производят объектную привязку.

Разработка многокамерной системы сопровождения является актуальной научно-инженерной задачей, решение которой сегодня востребовано пользователями различной категории. Появляется возможность анализировать поведение множества людей в масштабе всего охраняемого объекта, а не только в поле зрения одной камеры. Индивидуальные или обобщенные данные о перемещениях людей на обширной территории открывают принципиально новые возможности для построения систем поддержки принятия решения.

По мере совершенствования математической теории машинного зрения и вычислительной мощности аппаратного обеспечения приложения систем многокамерного сопровождения будут расширяться в новые отрасли.

Д.П. Ковалев (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **Е.И. Сукач**, канд. тех. наук, доцент

ПРОСТРАНСТВЕННОЕ МОДЕЛИРОВАНИЕ ХРАНИЛИЩ ДАНЫХ С ИСПОЛЬЗОВАНИЕМ IBM INFOSPHERE DATA ARCHITECT

Особенности пространственного моделирования данных. Пространственное моделирование данных (Dimensional Data Modeling) это имя набора методов и понятий, используемых в хранилище данных с измерениями (dimension data warehouse design). Это моделирование отличается от моделирования “сущность-связи” (entity-relationship modeling), и вообще любого нормализованного моделирования. Оно реализует подход к моделированию данных на логическом уровне, который может использоваться для любых физических форм, например, таких как многомерные-пространственные (multidimensional) базы данных. По словам главного консультанта по технологиям хранилищ данных

Ральфа Кимбалла [1], пространственное моделирование является методом проектирования баз данных, предназначенным для поддержки запросов конечных пользователей в хранилище данных, основными преимуществами которого является понятность и высокая производительность.

В модели «сущность-связь», описывающей нормализованные хранилища, данные находятся в предметно ориентированных таблицах третьей нормальной формы. Нормализованные хранилища характеризуются как простые в создании и управлении, но они используют большое количество таблиц, из-за чего для получения какой-либо информации формируется выборка из многих таблиц одновременно, что приводит к ухудшению производительности системы [1].

Пространственное моделирование направлено на создание хранилища с измерениями, которое использует схему «звезда» или схему «снежинка». При этом в центре «звезды» находятся данные (таблица фактов), а измерения образуют лучи звезды (таблица измерений). Как следствие, различные таблицы фактов совместно используют таблицы измерений, что значительно облегчает операции объединения данных из нескольких предметных таблиц фактов (например, факты продаж и поставок товара). Таблицы данных и соответствующие измерения образуют архитектуру «шина». Измерения часто создаются в третьей нормальной форме, в том числе, для протоколирования изменения в измерениях. Основным недостатком хранилища с измерениями является более сложные процедуры подготовки и загрузки данных, а также управление измерениями данных [2].

Работа с IBM InfoSphere Data Architect. Моделирование хранилища с измерениями начинается с создания логической модели данных, которая содержит в себе объекты (эквивалент таблиц), атрибуты (эквивалент столбцов и полей), отношения между объектами (relationships) и ключи (рисунок 1).

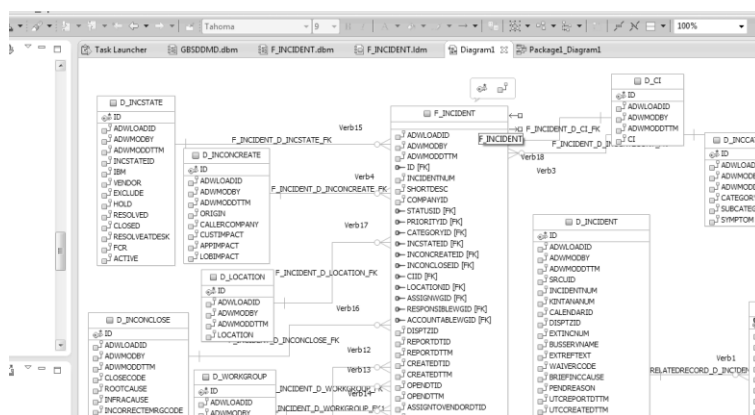


Рисунок 1 – Пример логической модели данных

Структура модели имеет четкую иерархию и очень удобно отображается в дополнительном окне (рисунок 2).

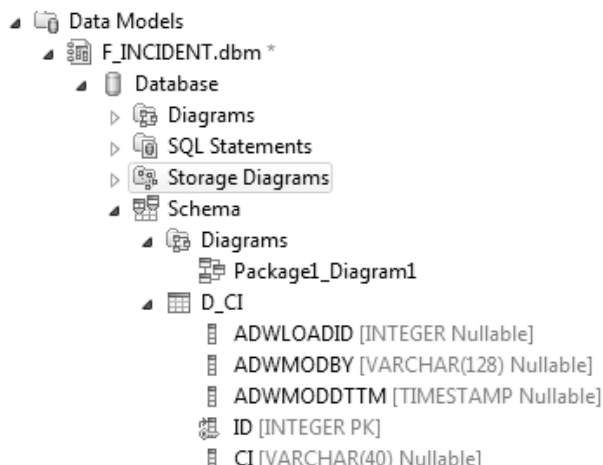


Рисунок 2 – Иерархия физической модели данных

Далее, посредством специальной функции IDA (transform to Physical Data Model) логическая модель данных трансформируется в физическую. Объекты становятся таблицами, атрибуты – столбцами, и появляются типы данных колонок (column data type), а отношения трансформируются во внешние ключи. В физической модели данных, в отличие от логической, более высокий уровень детализации. У полей появляется больше параметров, появляется возможность настраивания всех дополнительных параметров базы данных, которые недоступны в логической модели (индексы, автоповышение значений, вид, псевдонимы, процедуры и т.д. (рисунок 3).

Columns	Name	Primary ...	Domain	Data Type	Length	Scale	Not Null	Generated	Default Value/Generate ...
Privileges	ADWLOADID	<input type="checkbox"/>		INTEGER			<input type="checkbox"/>	<input type="checkbox"/>	
Partition Key	ADWMOBY	<input type="checkbox"/>		VARCHAR	40		<input type="checkbox"/>	<input type="checkbox"/>	
Table Spaces	ADWMODTTM	<input type="checkbox"/>		TIMESTAMP			<input type="checkbox"/>	<input type="checkbox"/>	
MDC	ID	<input checked="" type="checkbox"/>		INTEGER			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Relationships	SRCUID	<input type="checkbox"/>		INTEGER			<input type="checkbox"/>	<input type="checkbox"/>	
Volumentrics	INCIDENTNUM	<input type="checkbox"/>		VARCHAR	40		<input type="checkbox"/>	<input type="checkbox"/>	
Documentation	DESCRIPTION	<input type="checkbox"/>		VARCHAR	2056		<input type="checkbox"/>	<input type="checkbox"/>	
Annotation	COMMENTS	<input type="checkbox"/>		VARCHAR	512		<input type="checkbox"/>	<input type="checkbox"/>	
	WORKNOTES	<input type="checkbox"/>		VARCHAR	512		<input type="checkbox"/>	<input type="checkbox"/>	
	CLOSENOTES	<input type="checkbox"/>		VARCHAR	512		<input type="checkbox"/>	<input type="checkbox"/>	
	WATCHLIST	<input type="checkbox"/>		VARCHAR	512		<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 3 – Табличные настройки физической модели данных

В табличных настройках настраиваются не только поля, но и общее описание таблицы (закладка General); привилегии для доступа к таблице и полям таблицы (закладка Privileges); метод распараллеливания данных по определенным колонкам (закладка Partition Key); табличное

пространство, где будет создана таблица (закладка Table spaces); расширение таблицы до таблицы с многомерной кластеризацией (закладка MDC); отношения таблицы (закладка Relationships); объемность информации (закладка Volumetrics). Имеется возможность документирования текущего состояния таблицы и изменений (закладка Documentation), а так же просмотра общего описания (закладка Annotation).

После того, как все поля, их внешние ключи, индексы, типы данных и остальные параметры, необходимые для указания особенностей хранилища данных, настроены, генерируется ddl-скрипт, с полным указанием всех имен пространств (instance), схем (schema) и табличных пространств (table space). При запуске ddl-скрипт производит развертывание физической модели на выбранном сервере со всеми необходимыми деталями. При чем генерация скрипта полностью контролируется пользователем, а именно: какие части физической модели генерируются; какие синтаксические особенности имеет генерация. По завершении работы настраивается соединение с базой данных, и производятся необходимые преобразования хранилища прямо из IDA.

Литература

1. Кимбал, Р. The Data Warehouse Lifecycle Toolkit / Р. Кимбал: Wiley, 2004. – 800 с.
2. Спирли, Э. Корпоративные хранилища данных. Планирование, разработка и реализация. Том первый / Э. Спирли: Вильямс, 2001. – 400 с.

А.П. Кожемякин (УО «ГГУ им. Ф. Скорины», Гомель)

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

ПРОЕКТ МОДЕРНИЗАЦИИ ЛВС ЧУП «ГОМЕЛЬСКАЯ УНИВЕРСАЛЬНАЯ БАЗА»

Локальная вычислительная сеть предприятия – это нечто большее, чем просто сумма объединяемых ею компонентов. На подключенных к сети компьютерах можно совместно использовать общее подключение к Интернету, общий принтер и другое оборудование, а также общие файлы.

В работе рассматривается проблема модернизации локальной вычислительной сети ЧУП «Гомельская универсальная база». Основными задачами локальной вычислительной сети (ЛВС) являются организация информационного обмена данными между рабочими станциями, а также организация доступа пользователей к ресурсам ЛВС: базам данных, файловому серверу, сетевым принтерам. Кроме этого, проектируемая локальная сеть обеспечит взаимодействие системных приложений в

различных узлах, доступ к ним сотрудников, а также должный уровень защиты информации, не допускающий её искажения или утечки.

В ходе разработки проекта будут решены следующие задачи:

– во-первых, определение целей и задач разрабатываемой корпоративной сети, анализ существующей системы и требований, предъявляемых к ней, формирование технического задания;

– во-вторых, анализ возможных вариантов проекта сети, расчёт до проектной ситуации;

– в-третьих, выбор сетевого оборудования и создание проекта сети в имитационном пакете.

Проектирование локальной вычислительной сети – процесс сложный, длительный, требующий особого внимания и хороших знаний в области сетевых технологий.

Для построения локальных связей между компьютерами используются различные виды кабельных систем, сетевые адаптеры, концентраторы, повторители, мосты, коммутаторы и маршрутизаторы.

Для реализации проекта модернизации локальной вычислительной сети ЧУП «Гомельская универсальная база» необходимо будет осуществить плановое изучение всего предприятия с целью поиска узких мест в работе сети, а также устаревшего программного обеспечения. Модернизация ЛВС позволит увеличить ее пропускную способность, что в свою очередь, повысит эффективность работы в ней пользователей.

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

Науч. рук. **А.И. Кучеров**, ст. преподаватель

ПРИМЕНЕНИЕ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ В ПОВСЕДНЕВНОЙ ЖИЗНИ

2014 год в сфере технологий обещает стать не менее интересным, чем прошлый. Мобильные устройства продолжают наращивать «интеллектуальность» и научатся ещё точнее предугадывать желания владельца. 3D-печать, робототехника, интеллектуальные устройства обещают качественно изменить промышленный и потребительский рынок. Города и населённые пункты будут больше, чем сегодня связаны со smart-технологиями.

Носимые гаджеты и smart-одежда. Практически вся вторая половина 2013 года прошла под знаком «умных» носимых устройств: очков, часов, браслетов, колец и даже обуви с одеждой. Самые яркие представители этого класса устройств: «умные очки» GoogleGlass, «умные

часы» GalaxyGear, умеющие делать фотографии и позволяющие отвечать на звонки.

Уже существуют браслеты для контроля сна и питания JawboneUp и FitbixFlex. В 2014 году мобильные фитнес-устройства будут отслеживать все показатели здоровья и станут частью системы здравоохранения. К таким фитнес-устройствам можно отнести фитнес-браслет, который на самом деле знает, что вы делаете в тренажёрном зале. В 2014 году можно ожидать расцвета этой сферы для всех остальных производителей: Sony, Samsung, LG и других, которые также продолжат движение в этом направлении.

Общемировой объём поставок носимых устройств в 2013 году превысил 50 млн. штук. К концу 2014 года на рынке будет присутствовать порядка 100 млн. носимых гаджетов, а к 2018 году их число превысит 540 млн.

Что же касается смарт-одежды, то этот сегмент рынка должен окончательно оформиться в 2014 году. Уже известны примеры первых разработок в этой области. Например, OMsignal собирается стать первым в мире производителем специальной биочувствительной одежды, которая способна отслеживать изменения самочувствия человека. Её встроенные датчики позволят выявлять частоту биения сердца, ритмичность дыхания, уровень физической активности. Собранные данные можно будет контролировать в реальном времени через смартфон.

Мобильные деньги. В 2014 году продолжит набирать популярность мобильный кошелек – специальное приложение для смартфона. Однако, этот способ оплаты по-настоящему завоюет рынок лишь через несколько лет. Пока разные стартапы разрабатывают собственные версии мобильного кошелька, но окончательный выбор пользователя, скорее всего, выделит PayPal или Google. Но не исключено, что могут появиться и другие лидеры.

Смарт-кухня. В 2013 году произошла «эволюция» домашней утвари. К ней можно отнести: холодильники, микроволновки, чайники, которые научились общаться с пользователями голосом, знаками, выходить в интернет, получать задания удалённо. Наше с вами завтра – графины для молока, которые посредством SMS будут извещать о том, что молоко закончилось.

Необычные формы. Изогнутая форма корпуса станет постепенно вытеснять плоский дизайн, характерный для нынешних моделей смартфонов и телевизоров. В 2014 году эти устройства будут выходить на рынок, их станет больше, разнообразие форм и форматов будет расти. В компании LG, например, считают, что уже к 2015 году около 12% новых смартфонов будут иметь изогнутый дизайн, а к 2018 году их доля составит 40%. «Законодателями» этой моды стали LG и Samsung. LG

одна из первых выпустила модель с изогнутым дисплеем (LGGFlex). Более того, она объявила о своих планах по разработке смартфона, который будет иметь гибкий корпус. Выпуск смартфонов с изогнутой геометрией дисплея попыталась освоить в 2013 году компания Samsung в модели GalaxyRound. В отличие от LG корпус был изогнут не вдоль, а поперёк. Samsung готова продолжить эксперименты с дизайном. В планах компании на 2014 год – выпуск смартфона, дисплей которого изогнут с одного края по торцу, где расположатся кнопки управления.

В 2012 году был утверждён термин «UltraHD». В 2013 году ведущие производители телевизоров ориентировались на этот стандарт – разрешение экрана не менее 3840 x 2160 пикселей, наличие цифрового входа и соотношение сторон экрана 16 x 9. Появилась новая серия OLED-телевизоров с изогнутой формой экрана. Главными конкурентами на рынке телевизоров были Sony, LG, Samsung. В январе 2014 года состоялась выставка CES 2014, где компании LG и Samsung представили свои первые модели телевизоров с эластичным экраном. Это тот же самый изогнутый экран, но теперь его форму можно адаптировать с учётом местоположения зрителя. Samsung намерен в ближайшее время продемонстрировать технологию управления телевизором с помощью жестов.

Облачные сервисы. В 2014 году с дальнейшим развитием мобильного интернета будет расти интерес потребителей к облачным технологиям. Пользователи начнут активно перемещать данные с персональных систем хранения в «облако». От этого выиграют и производители, и потребители, которые получают доступ к набору новых облачных услуг. Всё это может привести к взрывному росту популярности персональных облаков.

Умные автомобили. Лидеры IT-рынка и крупнейшие игроки автопрома всюду развивают направление смарт-автомобилей. На январской выставке CES в Лас-Вегасе такие бренды как Hyundai, Audi, Honda, GeneralMotors, Google и производитель чипов Nvidia объявили о создании OPENAutomotiveAlliance (ОАА), который будет заниматься развитием рынка смарт-автомобильных систем на Android, содействовать созданию новых полезных или развлекательных приложений и совершать прочие действия для развития мультимедийных технологий в автопроме. Речь идёт о плотном взаимодействии смартфонов и планшетов на Android с бортовыми системами автомобилей. В том же направлении работу с автопромом ведут Apple и Microsoft.

3D-печать. В 2014 году 3D-печать продолжит бить рекорды популярности, мировые поставки 3D-принтеров вырастут на 75%, а к 2015 году – на все 100%. Ожидается массовое распространение относительно недорогих 3D-принтеров (от \$500 за один принтер, при том, что первоначальная цена на 3D-принтер – около \$3000).

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

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

СРАВНИТЕЛЬНЫЙ АНАЛИЗ НАИБОЛЕЕ ПОПУЛЯРНЫХ СУБД

Система Управления Базами Данных (СУБД) – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Приведем девятку наиболее популярных СУБД во всем мире:

Oracle.

MySQL.

Microsoft SQL Server.

PostgreSQL.

MongoDB.

Microsoft Access.

SQLite.

SybaseASE.

Cassandra.

Теперь немного о каждой из СУБД.

Oracle – объектно-реляционная система управления базами данных компании Oracle, которая является одной из ведущих коммерческих систем. Эта система предлагает большой функционал для решения поставленных задач, представлена в различных версиях.

MySQL – это свободная реляционная система управления базами данных, которая популярна для создания WEB-приложений и операций, связанных с элементами БД на платформах LINUX/BSD/Mac/Windows. MySQL является решением для малых и средних приложений. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц.

Microsoft SQL Server – это СУБД, которая используется для создания рабочих БД, БД для предприятий и WEB-приложений. Эта СУБД позволяет работать с большим количеством данных, таким как видео, фотографиями и т.д. MS SQL Server разработан таким образом, что может управлять 1Тб данных (для сравнения, MS Access – 1 Гб).

PostgreSQL – это реляционная СУБД, которую используют разработчики WEB-приложений. В основном, находит своё применение во многих организациях, используется в особо важных и широкомасштабных проектах. Эта система имеет следующие преимущества:

- расширенный набор функций;
- простота администрирования.

Эти преимущества делают его отличным выбором для тех, кто использует БД для разработки WEB-сайтов.

MongoDB – документо-ориентированная система управления базами данных с открытым исходным кодом, не требующая описания схемы таблиц. Написана на языке C++. Основные возможности:

- документно-ориентированное хранение (JSON-подобная схема данных);
- достаточно гибкий язык для формирования запросов;
- динамические запросы;
- поддержка индексов;
- профилирование запросов;
- быстрые обновления «на месте»;
- эффективное хранение двоичных данных больших объёмов, например, фото и видео;
- журналирование операций, модифицирующих данные в базе данных;
- поддержка отказоустойчивости и масштабируемости: асинхронная репликация, набор реплик и распределения базы данных на узлы;
- может работать в соответствии с парадигмой MapReduce;
- полнотекстовый поиск, в том числе на русском языке, с поддержкой морфологии.

Microsoft Access – реляционная СУБД корпорации Microsoft. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Может использоваться в небольшом бизнесе. MS Access, безусловно, стал очень популярным из-за его ключевого преимущества – возможности создавать ПО в кратчайшие сроки.

SQLite – компактная встраиваемая реляционная база данных. Обеспечивает простой и быстрый способ для работы с файловыми базами. Эта СУБД является идеальным решением в тех случаях, когда реализация, обслуживание и простота управления являются более значимыми, чем бесчисленное количество сложных функций. SQLite – это правильный выбор для небольших и средних WEB-сайтов.

Adaptive Server Enterprise (ASE или Sybase ASE) – реляционная система управления базами данных компании SAP, одна из СУБД, использующая в качестве основного процедурного SQL-расширения язык Transact-SQL.

Apache Cassandra – распределённая система управления базами данных, относящаяся к классу noSQL-систем и рассчитанная на создание высокомасштабируемых и надёжных хранилищ огромных массивов данных, представленных в виде хэша. Для упрощения взаимодействия с БД поддерживается язык формирования структурированных запросов CQL (Cassandra Query Language), который на первый взгляд напоминает SQL, но существенно урезан в функциональности. Промышленные решения на базе Cassandra развёрнуты для обеспечения сервисов таких компаний, как Cisco, IBM, Cloudkick, Reddit, Digg, Rackspace и Twitter.

Я.А. Коледа (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **В.Д. Левчук**, канд. тех. наук, доцент

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ ОБЛАЧНОГО СЕРВИСА ДЛЯ УПРАВЛЕНИЯ УЧЕБНЫМ СЕМИНАРОМ

Семинар является одним из основных видов практических занятий по техническим наукам. Он предназначен для углубленного изучения дисциплины, овладение определенной научной методикой. Главная цель семинарских занятий – обеспечить студентам возможность овладеть навыками и умениями использования теоретического знания применительно к особенностям изучаемой отрасли. Тема семинара определяется преподавателем заранее и повторяет тему ранее проведенной лекции. На таких занятиях учащиеся выступают перед своими однокурсниками с сообщениями, докладами и рефератами, которые могут сопровождаться показом иллюстраций, слайдов, фильмов.

При организации семинарского занятия важно обеспечить доступ студентов к освящаемой информации, представленной в различных формах: текстовые доклады, таблицы, изображения и других. Доступ можно организовать посредством нескольких способов, одним из которых является использование облачного сервиса. В настоящее время наиболее популярны такие сервисы, как: Dropbox, iCloud, Amazon Cloud Drive, Google Drive, SkyDrive, Evernote.

В сравнении с другими способами организации совместного доступа к ресурсам, облачные сервисы имеют ряд преимуществ, среди которых:

возможность просмотра информации различного типа непосредственно в окне браузера, не прибегая к специализированным программным продуктам;

интеграция в некоторые облачные сервисы онлайн-офисных пакетов, позволяющих совместно редактировать и просматривать документы;

широкая доступность через сеть; доступ к ним осуществляется на основе стандартных механизмов, что обеспечивает использование разнородных тонких и толстых клиентских платформ (например, мобильных телефонов, ноутбуков, КПК);

способность к быстрой адаптации: вычислительные возможности могут быстро и гибко резервироваться (часто автоматически) для оперативного масштабирования под задачи, и также быстро освобождаться;

фиксированный размер предоставляемой памяти для хранения данных (на платной или бесплатной основе).

Перечисленные выше возможности облачных сервисов могут быть эффективно применены при организации семинарского занятия, обеспечив при этом совместный доступ к данным, возможность редактирования документов, безопасность при работе с файлами, минимальные затраты системных ресурсов и требования к оборудованию пользователя.

Литература

1. Облачные вычисления / Википедия [Электронный ресурс]. – 2013. – Режим доступа: http://en.wikipedia.org/wiki/Cloud_computing. – Дата доступа: 15.01.2014.

2. Вендров, А.М. Проектирование программного обеспечения экономических информационных систем: учебник / А.М. Вендров. – М.: Финансы и статистика, 2002. – 352 с., ил.

3. Ресурс программ, групп, инструментов и продуктов Google для разработчиков [Электронный ресурс]. – 2013. – Режим доступа: <https://developers.google.com/>. – Дата доступа: 20.01.2014.

Я.А. Коледа (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **В.Д. Левчук**, канд. тех. наук, доцент

АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ ВНЕШНИМ ТРЕНИНГОМ ПРЕДПРИЯТИЯ «ЭПАМ СИСТЕМЗ»

Задача данного проекта заключается в автоматизации управления внешним тренингом предприятия «Эпам Системз». Практическим результатом проекта являются программные модули, обеспечивающие эффективное взаимодействие компонентов, предоставляемых сервисами Google. Автоматизируемым является процесс работы с практическими заданиями курса тренинга и их решениями. Необходимо перенести функции, выполняемые администратором или преподавателем, на сторону сервера. Среди этих функций следующие:

- проверка правильности названия решения;
 - поиск решения в папке студента и перемещение в другую папку;
 - открытие доступа к правильному решению;
 - открытие доступа к следующему этапу задачи;
 - написание рецензии на решение;
 - удаление решения из папки пользователя.
- Средствами автоматизации выступают сервисы GoogleDocs и GoogleDisk и встроенный язык GoogleScript [1].

Созданный проект позволяет удовлетворять следующие цели автоматизации:

- повышение эффективности учебного процесса;
- повышение безопасности;
- увеличение скорости обработки входящей информации;
- увеличение частоты производимых операций по обработке.

На рисунке 1 приведена диаграмма прецедентов, показывающая функциональность приложения с указанием выполняющих функциональность ролей и отношениями между прецедентами.

Такое действие как просмотр всех папок пользователя и поиск нужного решения могут занимать длинный промежуток времени, поэтому данное действие передано для выполнения серверу и реализовано с помощью переноса полученных решений из папок студентов в папку решений, доступной преподавателю. При этом для удобства можно файл с решением переименовывается (название задания и фамилия студента). Также автоматизирован процесс выставления отметки в журнал, открытие доступа к новому заданию (если задача состоит из нескольких этапов) и написание рецензии (занесение в документ названия проверенного задания и даты проверки). В таком случае, для преподавателя определены только действия создания документа, предоставления доступа к нему и проверка поступивших решений. Для студента доступен просмотр задания и отправление выполнено работы. Все остальные действия выполняются на стороне сервера.

Выполняемые сервером действия реализованы с помощью сервиса `GoogleAppsScripts`, который позволяет на языке `JavaScript` писать код, работающий в облаке `Google`. Сервис предоставляет широкий `API`. Он работает с сервисами `Google`: почта, документы, календарь, контакты, финансы, переводчик, карты; со сторонними сервисами: `OAuth`, `JDBC`, `SOAP`. Реализация класса с помощью языка `JavaScript` подразумевает создание функции-конструктора, которая содержит свойство `Название задания`. В параметрах конструктора передается значение этого свойства. Методы, описанные для объекта, должны быть реализованы как свойства прототипа объекта [2].

Интерфейс работы приложения реализован с помощью сервиса `Google Sites`. Он представляет собой упрощённый бесплатный хостинг, который позволяет сделать информацию доступной для людей, которым предоставлен доступ к сайту. Преимущества `GoogleSites`: большой выбор шаблонов и настроек; широкие возможности по организации контента; большой выбор форматов: от простых веб-страниц до досок объявлений и файловых хранилищ; возможность создать единый ресурс для хранения веб-контента и файлов; гибкая настройка доступа к контенту; поисковые технологии `Google`. Страница управления

обработкой задания представляет собой форму, в которой преподаватель выбирает задание и действия, которые необходимо произвести над выбранным заданием. После этого создается объект типа Задание и запускается функция, описанная как свойство объекта.

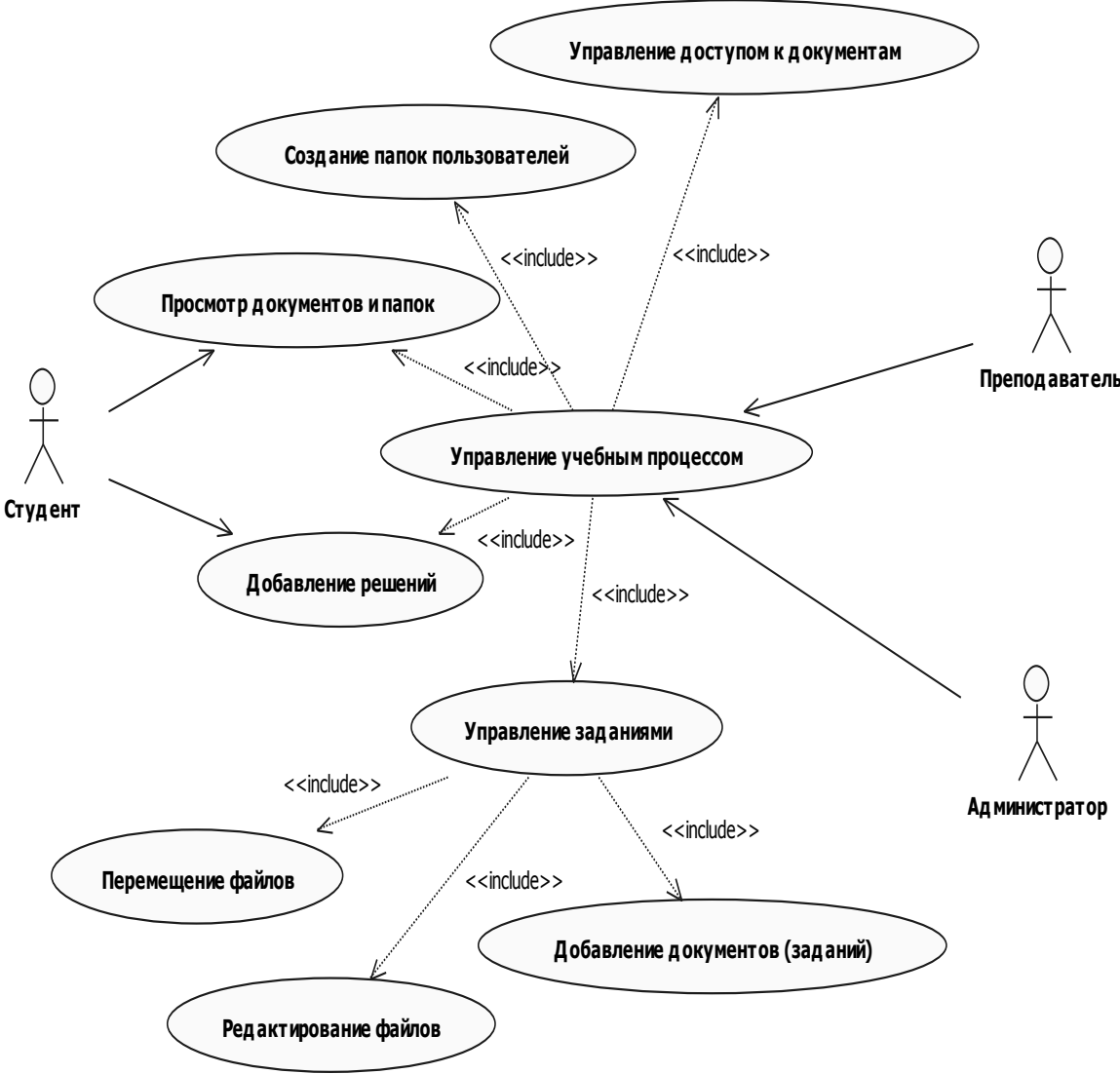


Рисунок 1 – Диаграмма прецедентов

Описанные выше действия в данном проекте представлены в виде объектной модели, показанной на рисунке 2.

В результате реализации проекта на основании функциональных возможностей приложения была разработана и создана объектная модель, задачами которой являются выполнение функций по обработке поступающих решений от студентов. В ходе работы над проектом были применены возможности сервисов GoogleDocs и GoogleDisk,

методы взаимодействия компонентов сервисов, и основные принципы написания кода на языке Javascript.

Task
+Название задания
+Переименование решения() +Перемещение в другую папку() +Удаление решения() +Выставление отметки о выполнении() +Открытие доступа к рефакторингу() +Проверка формата названия решения() +Занесение данных в рецензию()

Рисунок 2 – Диаграмма класса Задание

В результате реализации проекта на основании функциональных возможностей приложения была разработана и создана объектная модель, задачами которой являются выполнение функций по обработке поступающих решений от студентов. В ходе работы над проектом были применены возможности сервисов GoogleDocs и GoogleDisk, методы взаимодействия компонентов сервисов, и основные принципы написания кода на языке Javascript.

Литература

1. GoogleAppsScripts [Электронный ресурс]. – 2013. – Режим доступа: <https://developers.google.com/apps-script/>. – Дата доступа: 12.12.2013.

2. Спецификация языка JavaScript [Электронный ресурс]. – 2013. – Режим доступа: <http://www.w3schools.com/js/default.asp>. – Дата доступа: 12.12.2013.

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

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

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИИ СОЗДАНИЯ ГЕЛИЕВЫХ ЖЁСТКИХ ДИСКОВ

Если говорить о прогрессивных технологиях в области хранения данных в ЦОДах, то речь идёт об SSD накопителях. Выделим

некоторые плюсы SSD накопителей: постоянное увеличение скорости записи, дешевизна твердотельных дисков и так далее. Жёсткие диски уходили на «второй план», а их место стали занимать SSD накопители. В начале ноября 2013 года началась продажа жёстких дисков, заполненных гелием, от компании Western Digital и её дочерней компании HGST.

Гелий является вторым по распространённости химическим элементом во Вселенной и вторым по лёгкости после водорода. Для того чтобы осознать преимущества гелия, надо понимать, что происходит внутри жёсткого диска. Диски вращаются со скоростью в несколько тысяч оборотов в минуту, головка совершает бессчётное количество операций записи и считывания, все это осложняется трением и сопротивлением воздуха.

Наиболее очевидное свойство гелия применительно к жёстким дискам – это то, что он легче воздуха. Именно это делает его идеальным для использования в системах хранения данных. Если пространство внутри HDD накопителя заполнено гелием, диски могут вращаться быстрее, вибраций между головкой и диском становится меньше, сопротивление воздуха значительно сокращается. Выделим преимущества использования гелия в жёстких дисках.

Во-первых, благодаря сниженному сопротивлению внутри стандартного корпуса можно уместить больше дисков. Обычно их в дюймовый корпус HDD накопителя помещается пять, а при снижении сопротивления можно упаковать диски более плотно и поместить целых семь штук. Первый «гелиевый» жёсткий диск рассчитан на шесть терабайт, в то время как стандартные вмещают лишь четыре терабайта. В HDD накопителях, корпус которых заполнен одним из самых легких газов, вращение дисков более быстрое, что оборачивается увеличением скорости считывания и записи данных. Но это вовсе не означает, что диски будут быстрее выходить из строя: в силу того что сопротивление гелия меньше воздушного, расчётный цикл жизни HDD накопителя увеличивается.

Такая конструкция позволяет значительно снизить энергопотребление. На небольших системах хранения данных это традиционно незаметно. Однако корпоративные клиенты и компании, разворачивающие коммерческие ЦОДы, определённо оценят нововведение. По расчётам производителя, развёртывание системы хранения данных на 11 петабайт требует 12 стоек, заполненных суммарно 2 880 дисками. Это хранилище требует около 33 киловатт электроэнергии. В то время как увеличение объема каждого диска на 2 терабайта означает, что для хранения аналогичного количества данных необходимо лишь 1 920 дисков и 14 киловатт для того,

чтобы «завести» их. Не говоря уже о том, что меньшее количество HDD означает меньше пространства и меньше соединительных кабелей.

Еще одна особенность жёстких дисков, заполненных гелием – их герметичность. А это означает, что их можно использовать в стойках с жидкостным охлаждением. Например, Intel уже давно экспериментирует с масляным охлаждением, разрабатывая компоненты, которые можно погружать в специальное масло, не опасаясь коррозии или замыкания. Основной ключевой проблемой для Intel были как раз жёсткие диски. Масляное охлаждение сегодня – это вообще одна из наиболее прогрессивных технологий в строительстве дата-центров. До массового внедрения ей далеко, но её преимущества заставляют многих задуматься. Наиболее очевидное из этих преимуществ заключается в том, что при масляном охлаждении снимается необходимость во всех остальных охлаждающих элементах, основанных на снижении температуры за счёт отвода горячего воздуха. Более того, масло как охлаждающий элемент можно использовать повторно – в качестве источника электроэнергии. А главный секрет этой технологии – возможность поддерживать постоянную температуру внутри стоек, что значительно увеличит эффективность работы компонентов системы, а также их срок эксплуатации. Intel могла бы заниматься исследованием в области «гелиевых» систем хранения данных, но не выпускали бы свои собственные HDD накопители.

Несколько крупных компаний уже объявили о том, что будут применять HDD с гелием в своих дата-центрах. HP планирует ставить такие диски в серверы, Netflix хочет построить часть динамической системы хранения данных на базе прогрессивных HDD накопителей, Huawei также работает над внедрением «гелиевых» дисков. Однако один из самых перспективных и крупных клиентов WD – CERN. Европейская организация по ядерным исследованиям генерирует действительно большие объёмы данных и для записи использует жёсткие диски и кассеты. CERN уже тестирует диски, заполненные гелием, в своей системе хранения данных, рассчитанной на хранение 60 петабайт. Сегодня большая часть данных организации хранится на кассетах, так как гигантские их объёмы делают хранение на жёстких дисках нецелесообразным с финансовой точки зрения. По мнению европейской организации по ядерным исследованиям, жёстким дискам необходимо стать более финансово эффективными, чтобы конкурировать с кассетами, когда дело касается хранения действительно больших объёмов данных. HDD накопитель с гелиевым заполнением – это как раз шаг в необходимом направлении.

Много важнейших преимуществ новых жёстких дисков, особенно для компаний, которые разворачивают собственные ЦОДы, и для строителей коммерческих дата-центров. Эти преимущества сводят на нет

практически все основные проблемы жёстких дисков. HDD накопители с гелием внутри энергоэффективны, позволяют хранить больше данных в одном корпусе, могут использоваться в системах с жидкостным охлаждением, работают в среднем дольше, чем обычные жёсткие диски.

Несложно выявить единственный, но веский аргумент производителя жёстких дисков с гелием против SSD. Это цена. Несмотря на то, что стоимость SSD накопителей падает с каждым днём, твердотельным дискам всё ещё далеко до HDD дисков. Жёсткие диски значительно дешевле. Учитывая, что HGST долго работала над тем, чтобы поставить производство «гелиевых» HDD накопителей на поток, цена их также будет падать и со временем станет эквивалентна стоимости обычных HDD накопителей.

И.И. Коляскин (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

ТЕХНИЧЕСКИЕ АСПЕКТЫ ИГРОВОЙ СИТУАЦИИ

Разработку и управление компьютерными играми позволяет реализовать Microsoft XNA – набор инструментов с управляемой средой времени выполнения (.NET). Пакет включает в себя обширный набор библиотек классов, специфичных для разработки игр. Игры XNA пишутся для среды времени выполнения, поэтому они могут запускаться на любой платформе, поддерживающей XNA Framework.

Наибольший интерес при создании Арканоида, представляет файл Game1.cs, в котором определен класс Game1, наследованный от Microsoft.Xna.Framework.Game, где и разрабатывается приложение. В классе Game1 переопределены следующие методы Game: Initialize – инициализация ресурсов до начала игры; LoadContent – загрузки контента (спрайты и т.д.); UnloadContent – выгрузки контента; Update – реализации логики игры, обработка событий клавиатуры или джойстика, проигрывание аудио и т.д.; Draw – прорисовки игрового поля.

В папку Content добавляются игровые ресурсы: картинки фона, кирпича, ракетки и мячика. В методе LoadContent добавленные ресурсы используются для создания игровых объектов. В методе Draw рисуется фоновое изображение, ракетка, шарик и кирпичи. Прорисовываться должны кирпичи, находящиеся в состоянии «жив», то есть не разбитые мячом. Обработка движения и столкновения объектов происходит в методе Update. При каждом вызове Update проверяется состояние клавиш перемещения влево и вправо, и в случае, если какая-нибудь из них нажата, ракетка перемещается в соответствующую сторону. Для описания движения

мячика создан отдельный метод – UpdateBall, который вызывается из общего метода Update. В UpdateBall к текущей позиции мячика добавляется определенное значение. Также при каждом вызове UpdateBall осуществляется проверка положения мячика. Он должен отбиваться от кирпичей и от всех краев поля, кроме нижнего. В зависимости от поверхности столкновения движение мяча отражается либо по оси X, либо по оси Y.

Во время выполнения программы методы Update и Draw вызываются с достаточно высокой частотой. Таким образом, получается постоянное изменение положения мячика и других объектов и мгновенную прорисовку произошедших в игре изменений.

А.И. Кононович (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **А.Л. Самофалов**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА WEB-САЙТА НИЛ «ФИЗИКА ВОЛНОВЫХ ПРОЦЕССОВ»

В современном мире наличие web-сайта является неотъемлемой частью функционирования научного сообщества, поэтому разработка web-сайта для НИЛ «Физика волновых процессов» весьма актуальна и своевременна. При разработке сайта были учтены все современные тенденции в разработке web-сайтов. Функциональная часть сайта выполнена для удобства пользователей.

Основными задачами при разработке сайта были: представление научной лаборатории в сети интернет, возможность ознакомления зарубежных коллег с деятельностью лаборатории, информационная функция сайта.

Сайт представлен на русской версии, для русского сегмента пользователей, а также на английской версии для англоговорящего сегмента пользователей, которые будут находиться на сайте. Сайт наполнен всей актуальной информацией, имеет информативные графические элементы, удобную навигацию, которая помогает перемещаться по страницам сайта. Обладает системой управления, которая позволяет удобно работать с сайтом, редактировать информацию на сайте, добавлять материалы, новости, графические материалы и всю необходимую информацию.

При разработке графического интерфейса сайта использовался графический редактор AdobePhotoshop CS6, в котором разработаны все компоненты графической оболочки. При разработке функциональной часть сайта использовались такие технологии как HTML, CSS, а также бесплатная система управления контентом Joomla!и сопутствующие ей компоненты.

Разработка сайта включает в себя следующие этапы:

1. Разработка графического интерфейса web-сайта. Разработка графического psd-шаблона со всеми компонентами и блоками сайта.
2. Программная верстка psd-шаблона средствами HTML и CSS.
3. Интеграция программной верстки в систему управления контентом.
4. Наполнение сайта контентом.
5. Тестирование работоспособности сайта.
6. Размещение web-сайта в сети Интернет.

Все данные этапы являются неотъемлемой частью разработки современного web-сайта.

Литература

1. Основы работы в Adobe Photoshop CS5 [Электронный ресурс]. – URL: <http://www.e-reading.co.uk> – Дата доступа: 25.11.2013.
2. Реактивные веб-сайты. Клиентская оптимизация в алгоритмах и примерах: Учебное пособие /Н.С. Мациевский, Е.В. Степанищев, Г.И. Кондратенко – М.: Интернет Университет Информационных технологий: БИНОМ; 2010. – 336 с.
3. Создаем шаблон Joomla по стандартам [Электронный ресурс]. – URL: <http://habrahabr.ru/post/41213/> – Дата доступа: 25.11.2013.

Е.М. Кончиц (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. Е.А. Дей, канд. физ.-мат. наук, доцент

РАЗРАБОТКА КРАТКОГО СПРАВОЧНИКА ПО МЕХАНИКЕ В СИСТЕМЕ ANDROID

Система Android – одна из самых молодых и перспективных мобильных операционных систем (ОС), основанная на базе операционной системы Linux и разрабатываемая Open Handset Alliance (ОНА) при поддержке Google. К достоинствам этой ОС можно отнести гибкость, открытые исходные коды, большое количество готовых программ, высокое быстродействие, удобное взаимодействие с сервисами от Google, многозадачность. К недостаткам можно отнести множество актуальных версий и невысокую защищённость от хакерских атак из-за открытости кода.

Являясь студентом физического факультета и начинающим разработчиком, я решил освоить эту операционную систему в ходе создания краткого справочника по механике. В качестве среды разработки была выбрана программа Eclipse. Eclipse является основной средой программирования для ОС Android, так как именно для нее Google выпустила официальную библиотеку с самым широким функционалом. Для работы в среде Eclipse необходимы: программа Eclipse Classic, плагины Android Development Tools, Java Development Kit, Android

Software Development Kit. Все это является бесплатным и общедоступным.

Разработанное приложение содержит главное и дочерние окна, графический интерфейс которых формируется с использованием объектов View и ViewGroup. Главное окно представлено в виде списка элементов управления, называемых виджетами, в данном случае кнопками. Название каждой кнопки соответствует определённому разделу механики. Дочерние окна либо отображают информацию о разделе, выбранного пользователем, либо состоят из кнопок, названия которых отображают пункты раздела «Справочные таблицы». За расположение кнопок на форме отвечает класс ViewGroup, представляющий из себя контейнер, который служит ядром для подклассов, называемых компоновками (layouts).

Справочник содержит следующие разделы (рисунок 1а): «Кинематика материальной точки», «Кинематика вращательного движения», «Динамика материальной точки», «Закон сохранения импульса», «Закон сохранения энергии», «Всемирное тяготение», «Справочные таблицы». Раздел «Справочные таблицы» делится на подразделы: «Фундаментальные константы», «Обозначения физических величин», «Приставки для кратных и дольных единиц», «Поиск в Википедии» (рисунок 1б). Пример отображения информации подраздела «Фундаментальные константы» приведён на рисунке 1в.

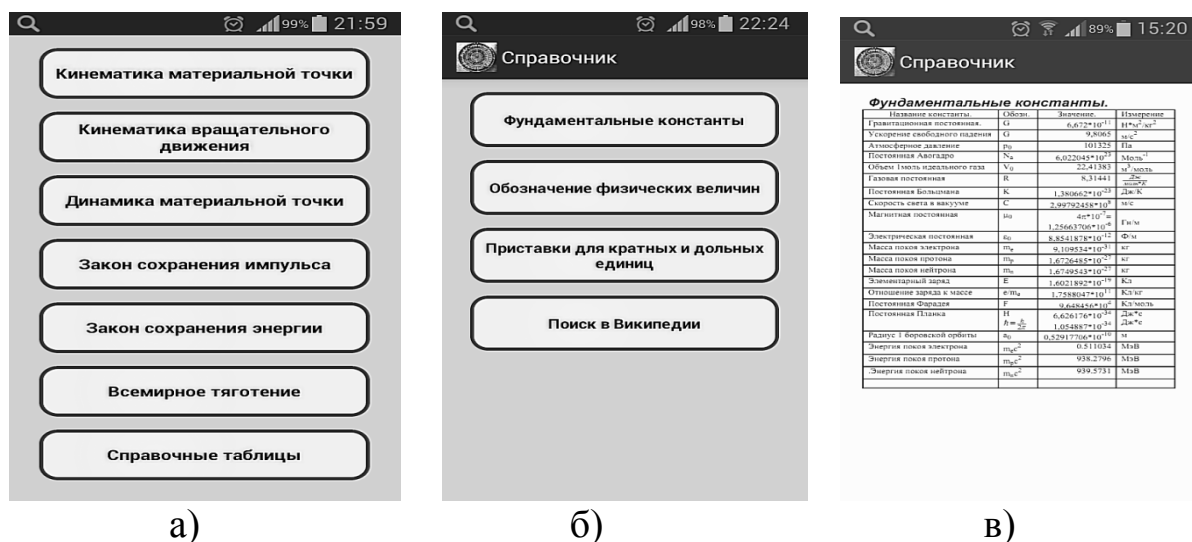


Рисунок 1 – вид экрана: а) при запуске приложения, б) в режиме «Справочные таблицы», в) в режиме «Фундаментальные константы»

При запуске приложения система получает ссылку на корневой элемент представления (layout-файл), который используется для графической прорисовки изображения на экране мобильного телефона. Для этого в методе onCreate() вызывается метод setContentView(), который

передаёт ему в качестве параметра ссылку на ресурс компоновки в виде `setContentView(R.layout.layout_file_name)`

Пример загрузки разметки в коде программы:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.zakon_impulsa);
}
```

Для того, чтобы по нажатию кнопки отображалась информация о конкретном разделе, в layout-файл добавлен компонент `WebView`. В него загружается информация в виде html-файла. Чтобы осуществить загрузку html-файла в `WebView`, в программном коде `Activity` прописана следующая команда:

```
WebView myWebView = (WebView) findViewById(R.id.webView);
myWebView.loadUrl("file:///android_asset/zakon_impulsa.html");
```

В справочнике реализована возможность увеличения отображаемой информации, что очень удобно при просмотре. В качестве примера на рисунке 2 а изображена увеличенная информация раздела «Кинематика материальной точки». Кроме этого, для удобства использования справочника в него встроен свой браузер, который работает при помощи компонента `WebView`. Сам компонент использует движок `WebKit`. По нажатию кнопки «Поиск в Википедии», которая находится в разделе «Справочные таблицы», браузер открывает начальную страницу информационного сайта «Википедия». Пример результатов поиска изображён на рисунке 2 б. Диалоговое окно, содержащее информацию о самом приложении, показано на рисунке 2 в.

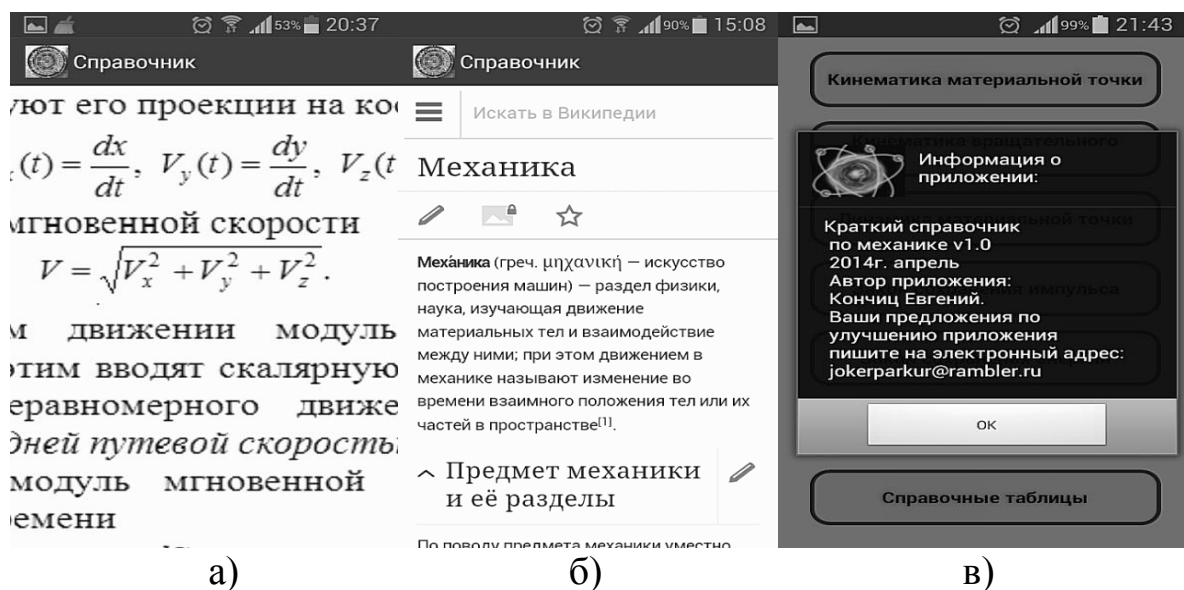


Рисунок 2 – Вид экрана в режиме:

- а) увеличения отображаемой информации, б) «Поиска в Википедии»,
- в) диалогового окна «Информация о приложении»

В заключение хочется отметить, что ОС Android – интуитивно понятная платформа для разработки и, благодаря этому, программисты, которые не имели дела с разработкой ПО для мобильных устройств, могут легко и быстро начать создавать собственные полноценные приложения под Android.

Литература

1. Голощапов А.Л. Google Android. Создание приложений для смартфонов и планшетных ПК/ А.Л. Голощапов СПб.: БХВ-Петербург, 2013. – 832 с.: ил. – (В подлиннике).

В.Н. Копачев (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА СЕТЕВОГО ПРИЛОЖЕНИЯ НА ЯЗЫКЕ JAVA С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ SWING

С развитием различных технологий в Интернете появляются все больше услуг, предлагаемых обычным пользователям, соответственно, растет и число пользователей. Зачастую возникает необходимость общения пользователей с общими интересами в реальном режиме времени. Примером такой задачи является реализация чата. Историческим предшественником чатов, несомненно, был телефон. Ни почта, ни телеграф не позволяли общаться в реальном времени. Возможность поговорить с собеседником на другой стороне Земли казалась настоящим чудом.

В ходе выполнения задачи был разработан дизайн интерфейса, структура серверной и клиентской части приложения, разработана база данных. Для разработки графического интерфейса используется библиотека Swing, для реализации передачи данных между серверной частью приложения и базой данных используется технология Hibernate.

Данный проект ориентирован на различных пользователей с общими интересами. С его помощью пользователи смогут общаться между собой, делиться различной информацией и задавать интересующие их вопросы.

Эту задачу способно решить разработанное приложение, которое позволяет объединять пользователей в открытую дискуссионную группу, в которой можно общаться с другими людьми в режиме реального времени, используя псевдоним. В обсуждении может участвовать множество пользователей, а также возможно личное общение между двумя пользователями.

Главной целью создания приложения являлось, обеспечение простой и максимально удобной в эксплуатации среды общения. Реализованная программа чрезвычайно легка в настройке для любого пользователя. Так же одним из плюсов реализованной программы является её переносимость на различные операционные системы.

Функционал программы и её интерфейс может быть расширен и усовершенствован в зависимости от потребностей разработчика с целью дальнейшего повышения информативности, привлекательности и удобства приложения.

К.В. Корж (УО «ГГУ им. Ф. Скорины», Гомель)

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

РАБОТА С ПОЛЬЗОВАТЕЛЯМИ В СИСТЕМЕ 1С: ПРЕДПРИЯТИЕ 8

Администрирование и контроль пользователей 1С – это неотъемлемая часть внедрения и поддержки любого программного продукта фирмы 1С.

Управление пользователями в 1С процесс достаточно простой и интуитивно понятный, однако всё равно нуждается в описании.

Условно, администрирование и контроль включает в себя:

- создание пользователя;
- установка прав пользователю;
- просмотр активных пользователей;
- анализ действий пользователей.

В зависимости от конфигурации пользователи вводятся либо в конфигураторе, либо в режиме пользователя. Практически все современные конфигурации поддерживают ввод пользователей в режиме 1С-Предприятие. Также, в режиме 1С-Предприятие, как правило, вводятся дополнительные параметры пользователей.

Однако, независимо от конфигурации, первый пользователь с административными правами всегда вводится в режиме конфигуратор.

Доступ к базе осуществляется по 3-м сценариям:

- ввод имени пользователя и пароля;
- аутентификация по пользователю компьютера;
- ввод только имени, без пароля;
- вход без ввода данных, как правило, такой вариант присутствует только в только что созданной конфигурации.

Как такового «суперпользователя» или «группы администраторов» не существует. Администратором считается тот пользователь,

у которого включены все права в конфигурации и права на администрирование. В пустой базе (когда список пользователей еще пуст) – первым должен быть добавлен именно такой пользователь.

В 1С: Предприятие существует два списка пользователей. Один из них (список пользователей 1С) – с точки зрения программиста «настоящий». Он находится в конфигураторе. Именно по нему система определяет пользователя.

Это подход старых типовых конфигураций – пользователи редактируются в этом списке, а в справочник пользователей попадают автоматически при первом входе.

Второй (пользователи версии 1С 8.2, «не настоящий») – это справочник пользователи (и справочник внешние пользователи). Справочник был и раньше, однако подход новых типовых конфигураций состоит в том, что пользователи заводятся именно в нем, а в «настоящий» список попадают автоматически.

Основная проблема этого подхода состоит в том, что тем кому не нравится так работать и хочется по старому – не могут это сделать, так как при заведении заполняются определенные поля, а если завести пользователя вручную в список, то они уже не подхватятся в справочнике автоматически.

У пользователя может быть несколько ролей. При назначении нескольких ролей, полномочия по доступу к объектам суммируются. В управляемом приложении роли пользователя оказывают серьезное влияние на формирование пользовательского интерфейса и его форм. Желательно не вносить изменений в существующие роли, а создавать новые и на это есть следующие причины:

- в случае обнаружения ошибок можно легко вернуться к первоначальному рабочему варианту, посмотреть, как это было сделано там;
- если конфигурация находится на поддержке нескольких поставщиков, то создание своих ролей тоже может стать определенным плюсом. Например, поставщик внес изменения в свою роль, а заказчика все устраивает в старом варианте.

Обычный интерфейс пользователям и разработчикам 1С хорошо знаком, он существует со времени выхода платформы «1С: Предприятие 8.0».

Основные особенности обычного интерфейса это:

- наличие главного меню;
- неизменность главного меню для всех пользователей независимо от их прав доступа и каких-либо настроек;
- для разных пользователей можно создавать разные интерфейсы.

Управляемый интерфейс (синоним «командный интерфейс») – состоит из команд и окон, является динамическим, т.е. доступность тех или иных команд зависит от прав пользователей, настроек, сделанных в

конфигурации и других параметров.

С помощью команд в управляемом интерфейсе можно переключаться между разделами конфигурации, открывать различные списки, формировать отчеты.

Запуск конфигурации в управляемом режиме начинается с рабочего стола. На рабочем столе должна быть представлена информация, актуальная для пользователя – задачи пользователя, текущие сделки и т.д. Рабочий стол играет роль своеобразного помощника пользователя. Начало рабочего дня всегда происходит с ознакомлением с информацией, которую предоставляет рабочий стол:

- Что появилось нового?
- Что нужно сделать сегодня?
- Каково состояние важных для меня сведений?
- На что следует обратить внимание?

Рабочий стол обычно содержит несколько форм: список курсов валют, список документов поступления/продажи товаров, текущие взаиморасчеты и тому подобную информацию. Состав рабочего стола настраивается под конкретную должность. К примеру, стол продавца и стол менеджера по продажам, скорее всего, будут различаться. Рабочий стол имеет возможности настройки, не прибегая к средствам конфигурирования.

Главное преимущество управляемого интерфейса – возможность работы в веб-клиенте (веб-браузере). Нет необходимости ставить платформу 1С на компьютер. Для пользователей на операционной системе Linux, для доступа к информационной базе 1С используется веб-браузер Mozilla Firefox.

В платформе 8.2 основной интерфейс управляемый работает на управляемых формах, но платформа 8.2 поддерживает и обычный интерфейс с обычными формами.

Литература

1. Хомоненко, А.Д. Базы данных. Учебник для высших учебных заведений / А.Д. Хомоненко. – СПб.: Корона, 2002. – 266 с.
2. Бояркин, В.Э. 1С: Предприятие 8. Простые примеры разработки/ В.Э. Бояркин, А.И. Филатов – М.: ООО «1С-Публишинг», 2010 – 383 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. Рыбалка, В.В. Пример быстрой разработки приложений на платформе 1С: Предприятие 8.2 / В.В. Рыбалко. – М.: ООО «1С-Публишинг», 2009. – 182 с.
5. Ролланд, Ф.Д. Основные концепции баз данных/ Ф.Д. Ролланд – М.: Вильямс, 2002.– 256 с.

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

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

РАЗРАБОТКА WEB-САЙТА ДЛЯ ЧТПУП «МЕДИА-МИР»

Целью разработки было создание web-сайта для фотосалона ЧТПУП «Медиа-мир».

В определении задач сайта необходимо отталкиваться, прежде всего, от сферы деятельности заказчика. «Медиа-мир» имеет достаточную базу клиентов, партнеров, покупателей и поставщиков. Есть задачи поиска новых клиентов или покупателей, задачи обмена информацией с партнерами или филиалами компании. Каким-то образом эти задачи сейчас решаются. Сайт может позволить выполнять эту работу более эффективно, но для этого нужно решить, какого рода информация должна быть на сайте, в каком виде будет осуществляться взаимодействие между участниками этого информационного обмена.

Трудоемкость работы над конкретным сайтом зависит от множества факторов: креативности дизайнерской идеи, детализации проработки дизайна, структуры и объема информации, наличия уникальных программных блоков, функциональности административного интерфейса, наличия средств интерактивного взаимодействия с посетителями сайта, связи с локальными базами данных, и многого другого.

Для того чтобы понять, какие именно из этих элементов, в каком виде и объеме будут использоваться на сайте, нам необходимо узнать от заказчика о задачах, которые должен решать сайт, его концепции, и структуре информации на нем.

Для более точной оценки мы подробно обсуждаем как общие задачи сайта, так и функциональности каждого отдельного блока. От руководства компании, как говорится «из первых рук», мы получаем общую информацию о концепции и задачах проекта.

Получив необходимую информацию о задачах и общей структуре сайта, мы можем предложить конкретные варианты представления информации на сайте, технические решения по реализации разделов и программных блоков.

Конечно же, сайт выполняет функции формирования имиджа компании. Помимо создания общего образа и настроения, дизайн сайта влияет на лояльность потенциального клиента – детальная проработка каждого элемента внушает посетителю уверенность в качестве предлагаемых услуг или товаров, в надежности фирмы.

Но не меньшую роль в завоевании доверия посетителя имеет и информация, представленная на сайте, а также то, насколько удобно этой информацией пользоваться и насколько она свежа и актуальна.

Основными задачами для сайтов коммерческих организаций обычно являются: привлечение новых клиентов, продажа товаров или услуг, информационная поддержка существующих клиентов или партнеров, организация процесса получения и обработки заявок. Но каждая из этих задач может быть решена различными техническими средствами и воплощена в различной форме.

У предприятия широкий спектр услуг, таких как печать фотографий в различных форматах; проявка, печать и сканирование фотографий с пленки; распечатка, копирование и сканирование документов; печать на футболках, нанесение рисунков на посуду; изготовление брелоков; изготовление визиток; изготовление календарей; ламинирование; создание медальонов и прочее.

Пользователь, заходя на данный web-ресурс, желает найти нужную для него информацию. Данный web-сайт является далеко не развлекательным порталом, а несет в себе больше рекламную суть, информируя о новых ценах, услугах и товарах фотосалона.

Для реализации поставленных требований выбран язык PHP с использованием баз данных MySQL. Также в проекте будут задействованы Macromedia Dreamweaver 8 и программная оболочка Denwer 3. Для разработки дизайна сайта использовался пакет Adobe Photoshop CS. Основной упор разработки делался на дизайн сайта, и на его контент.

В данном web-сайте предусматриваются следующие роли:

1 Администратор: единственное лицо, которое имеет доступ к администраторской части сайта, которое занимается его обновлением;

2 Зарегистрированный пользователь: лицо, которое имеет право просматривать информацию сайта, а также пользоваться форумом и оставлять сообщения;

3 Гость: лицо, которое имеет право исключительно на просмотр открытой информации сайта;

Данное приложение реализует следующие интерфейсы:

1 Главная: данный интерфейс из себя представляет страницу приветствия.

2 Галерея багетов: в данном интерфейсе пользователь может просмотреть имеющиеся багеты.

3 Галерея паспарту: в данном интерфейсе пользователь может просмотреть имеющиеся виды паспарту.

4 Компьютерные услуги: здесь пользователь имеет возможность увидеть перечень предоставляемых услуг, просмотреть образцы компьютерной обработки, увидеть ценники на предоставляемые услуги.

5 Печать на плоттере: в данном интерфейсе пользователь может

просмотреть максимальную ширину печати на плоттере, типы бумаги, холстов, а также сроки изготовления и цены.

6 Печать фото: предлагаются различные форматы печати фотографий, цены и скидки.

7 Статьи: раздел, содержащий статьи фотоцентра.

8 Новости: в данном интерфейсе пользователи могут видеть новости сайта по разделам.

9 Поиск по сайту: в данном разделе сайта можно найти нужную пользователю информацию.

Связь с нами: в данном разделе любой пользователь может связаться с администратором сайта по указанной там электронной почте.

Пользовательский интерфейс представляет собой php-страницы. Эти данные передаются при помощи объектов запроса и сессии, причём в сессии хранятся данные, важные для пользователя, а в запросе – для конкретной страницы.

Дизайн – вещь достаточно субъективная. У каждого человека есть свои критерии оценки качества дизайна. Дизайн разрабатывался исходя из нововведенного мной общего стиля фотосалона. В данном стиле оформлено множество вывесок, преysкурантов, афиш, справочников и прочего. После создания средствами Photoshop CS дизайна, руководство ЧТПУП «Медиа-мир» было с ним ознакомлено, а необходимые изменения были произведены.

При разработке были использованы следующие технологии:

- 1 Язык PHP;
- 2 База данных MySQL;
- 3 Языки разметки HTML;
- 4 Adobe Photoshop CS.

Были решены основные задачи сайта:

- 1 Поиска новых клиентов или покупателей;
- 2 Задачи обмена информацией с партнерами или филиалами компании;
- 3 Задача формирования имиджа компании;
- 4 Реклама предоставляемых услуг предприятия;
- 5 Реклама продаваемых товаров;

Таким образом, разработанный веб-сайт уже имеет достаточную для заказчика функциональность, имеет достаточно простой интерфейс, приемлемый уровень защищенности, но данный веб-сайт в ближайшем времени будет модернизироваться, совершенствоваться, до тех пор, пока не доведется до совершенства.

Ю.А. Котлярчук (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. П.Л. Чечет, канд. тех. наук, доцент

СОВЕРШЕНСТВОВАНИЕ ДЕЯТЕЛЬНОСТИ КОММУНАЛЬНЫХ ЖИЛИЩНЫХ ПРЕДПРИЯТИЙ

Внедрение информационных технологий на коммунальных жилищных предприятиях происходит очень медленно. В Республике Беларусь существует всего лишь несколько официальных сайтов коммунальных предприятий в силу слабого уровня модернизации экономической деятельности в этой области. В связи с этим, разработка сайта, как совершенствование деятельности предприятия, является актуальным в настоящее время.

В качестве функциональности в первую очередь необходимо решить проблемы предоставления услуг для населения. Разработка сайта позволит предоставить населению полную оперативную информацию о предприятии в любое время суток:

- услуг;
- контактной информации;
- выполнить прием заявок от населения к службе;
- выполнить хранение заявок от населения;
- выполнить контроль над заявками от населения и контроль над их выполнением.

С использованием технологии HTML в процессе разработки сайта решаются задачи отображения статического контента. При помощи технологии PHP для сайта разрабатывается его динамическая составляющая. При помощи сервера MySQL для сайта создается база данных, а также разрабатываются функции управления данными. При помощи JavaScript на страницах сайте создано интерактивное меню сайта с главными пунктами и подпунктами.

Сайт состоит из двух частей:

– Информационная часть, которая будет доступна любому пользователю. Эта часть содержит общие сведения о предприятии, а также контактную информацию.

– Управленческая часть, которая доступна только для администратора, содержит элементы управления базой данных заказов.

Web-сайт содержит: информационную часть и управленческую часть под паролем – для работы администратора со списком заказов. Общая схема управления данными и информационной части сайта представлена на рисунке 1.



Рисунок 1 – Общая схема управления данными WEB-сайта

По схеме сайт состоит из двух частей:

- Информационная часть, которая будет доступна любому пользователю. Эта часть будет содержать общие сведения о предприятии, а также контактной информацией.

- Управленческая часть, которая будет доступна только для администратора. Вход и просмотр этой части будет разрешена после введения пароля. Управленческая часть будет содержать элементы управления базой данных заказов.

Для реализации поставленных целей использовались язык разработки PHP пятой версии и сервер баз данных MySQL.

При помощи HTML средств можно создать лишь только статический сайт, который будет представлять набор страниц. Связи же в таком сайте будут ссылками на страницы.

Технология PHP предоставляет возможность написания динамического сайта, страницы которого будут изменяться в зависимости от управления пользователем.

Информационная часть, описывающая заказы населения, следующая:

- фамилия, имя, отчество заказчика;
- номер услуги из списка;
- адрес заказчика;
- контактная информация.

При каждом открытии сайта, проверяется существование базы данных, таблицы с информацией о заказчике. При первом запуске сайта, такой таблицы нет.

Создание и обработка данных в базе данных в курсовом проекте осуществляется при помощи запросов SQL. На рисунке 2 представлена общая структура макета сайта.

1 Шапка сайта с названием предприятия	
2 Главное меню сайта	
3 Левый столбец сайта с дополнительной информацией и элементами управления для администратора	4 Центральная основная информационная часть сайта
5 О проекте	

Рисунок 2 – Общая структура макета сайта

Из рисунка 2 видно, что сайт состоит из 5 структурных частей:

1 Шапка сайта с названием предприятия. В этой части должен быть размещен логотип, название предприятия, а также дополнительные элементы: выход на главную страницу, добавление страницы в «Избранное», установка страницы в качестве домашней и другие.

2 Главное меню сайта должно быть интерактивным со следующими пунктами: «Официальный сайт», «О предприятии», «Структура», «История», «Услуги», «Список», «Заказать», «Контакты», «Телефоны», «Адрес».

3 Левый столбец сайта с дополнительной информацией и элементами управления для администратора: «Показать все заказы» и «Выход».

4 Центральная основная информационная часть сайта. В центральную часть сайта должна выводиться вся информация по пунктам меню.

5 О проекте.

Таким образом, предложенная структура сайта позволяет обеспечить поставленную функциональность как для пользователей-клиентов, так и для администратора, что будет способствовать повышению качества предоставления услуг коммунального жилищного предприятия.

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

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

АНАЛИЗ ТИПОВЫХ ТОПОЛОГИЙ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ

Под топологией (компоновкой, конфигурацией, структурой) компьютерной сети обычно понимается физическое расположение компьютеров сети друг относительно друга и способ соединения их линиями связи.

Сетевая топология может быть:

- физической – описывает реальное расположение и связи между узлами сети;
- логической – описывает хождение сигнала в рамках физической топологии.

Физические топологии различаются по стоимости и функциональности. Логическая топология определяет реальные пути движения сигналов при передаче данных по используемой физической топологии. Базовые топологии сетей: шина, звезда и кольцо

Шина. Для организации сети минимально необходимо одна линия передачи данных и по одному сетевому интерфейсу для каждого участника сети. Такая топология называется шинной (другое название – мнoкoкaнaл). К единственной незамкнутой линии передачи данных в произвольных точках подключаются все участники

Шина позволяет легко добавлять новых участников к сети, для прокладки линии требуется минимальное количество кабеля. Основным недостатком – любой разрыв линии делает сеть неработоспособной. К тому же такой отказ довольно трудно локализовать, поскольку все абоненты включены параллельно, и понять, какой из них вышел из строя, невозможно.

Звезда. В звездообразной топологии принято различать два типа топологий:

- звезда с пассивным центром;
- звезда с интеллектуальным центром.

Звездообразная топология требует наличия специального многопортового устройства – концентратора.

Концентратор соединяется с каждым участником сети отдельной линией передачи данных. При выходе из строя одной из линий доступ к сети теряет только один участник. Однако если откажет концентратор, работа сети станет полностью невозможной. Среди концентраторов выделяются активные (active) и пассивные (passive). Активные концентраторы регенерируют и передают сигналы так же, как это делают репитеры. Иногда их называют многопортовыми репитерами – они имеют от 8 до 12 портов для подключения компьютеров.

Некоторые типы концентраторов являются пассивные, например монтажные панели или коммутирующие блоки. Они просто пропускают через себя сигнал как узлы коммутации, не усиливая и не восстанавливая его. Пассивные концентраторы не надо подключать к источнику питания.

Гибридными (hybrid) называются концентраторы, к которым можно подключать кабели различных типов. Сети, построенные на концентраторах, легко расширить, если подключить дополнительные концентраторы.

Активное оборудование обладает информацией о структуре сети и может выбирать путь передачи данных, передавая данные только одному участнику, для которого они предназначены и не загружая остальные линии.

Большое достоинство звезды (как активной, так и пассивной) состоит в том, что все точки подключения собраны в одном месте. Это позволяет легко контролировать работу сети, локализовать неисправности путем простого отключения от центра тех или иных абонентов (что невозможно, например, в случае шинной топологии), а также ограничивать доступ посторонних лиц к жизненно важным для сети точкам подключения. Общим недостатком для всех топологий типа звезда (как активной, так и пассивной) является значительно больший, чем при других топологиях, расход кабеля.

Кольцо. При кольцеобразной топологии каждый участник соединен отдельной линией передачи данных с двумя соседями. Данные по каждой линии передаются обычно только в одном направлении. Блоки данных ретранслируются каждым участником до тех пор, пока не попадут к получателю. Как правило, после этого блок продолжают передавать по кольцу дальше. Отправитель, получив свой блок, прошедший полный круг, изымает его из кольца и удаляет. Основным недостатком кольцевой топологии – при обрыве хотя бы одной линии, или выходе из строя хотя бы одного участника, сеть перестает функционировать.

Из трех рассмотренных топологий кольцо наиболее уязвимо к повреждениям кабеля, поэтому в случае топологии кольца обычно предусматривают прокладку двух (или более) параллельных линий связи, одна из которых находится в резерве. Иногда сеть с топологией кольцо выполняется на основе двух параллельных кольцевых линий связи, передающих информацию в противоположных направлениях.

Сравним основные характеристики физических топологий вычислительных сетей (таблица 1).

Таким образом, в конечном счете, основным критерием выбора удачной топологии являются требования пользователей к производительности. Такие факторы, как стоимость, предполагаемая модернизация и ограничения существующих технологий, играют второстепенную роль.

Таблица 1 – Характеристики физических топологий сетей

Характеристики	Топология		
	<i>Звезда</i>	Кольцо	Шина
Стоимость расширения	Незначительная	Средняя	Средняя
Присоединение абонентов	Пассивное	Активное	Пассивное
Защита от отказов	Незначительная	Незначительная	Высокая
Размеры системы	Любые	Любые	Ограниченны
Защищенность от прослушивания	Хорошая	Хорошая	Незначительная
Стоимость подключения	Незначительная	Незначительная	Высокая
Поведение системы при высоких нагрузках	Хорошее	Удовлетворительное	Плохое
Возможность работы в реальном режиме времени	Очень хорошая	Хорошая	Плохая
Разводка кабеля	Хорошая	Удовлетворительная	Хорошая
Обслуживание	Очень хорошее	Среднее	Среднее

Топологии различаются требуемой длиной соединительного кабеля, удобством соединения, возможностями подключения дополнительных абонентов, отказоустойчивостью, возможностями управления обменом. Топологическая структура влияет на пропускную способность и стоимость локальной сети. Каждая топология сети налагает ряд условий: например, она может диктовать не только тип кабеля, но и способ его прокладки. На основе проведенного исследования можно сделать выбор наиболее подходящей для определенных целей топологии вычислительной сети.

Ю.В. Кулаков (УО «ГГУ им. Ф. Скорины», Гомель)

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

РАЗРАБОТКА ПРОЕКТА МОДЕРНИЗАЦИИ ЛВС ДЛЯ ЧТУП «МЕДИАМИР»

На сегодняшний день разработка и внедрение локальных информационных систем является одной из самых интересных и важных задач в

области информационных технологий. Появляется потребность в использовании новейших технологий передачи информации. Интенсивное использование информационных технологий уже сейчас является сильнейшим аргументом в конкурентной борьбе.

Но современные условия организации труда требуют постоянного улучшения технологического процесса. Практика показывает, что модернизацию локальных вычислительных сетей необходимо проводить приблизительно каждые 3–6 лет. Это обусловлено не только развитием новых технологий и увеличением объёмов передачи данных, но и необходимостью решать задачи стоящие перед предприятием в данный период развития. Однако часто причиной для модернизации является неправильный подход при изначальном построении сети. Это связано с тем, что большинство компаний при построении первой сети не имели чётко сформулированных целей, с позиции развития ИТ-инфраструктуры. От модернизации ЛВС повышается эффективность руководителя, позволяя видеть более целостную картину работы предприятия, появляется возможность собирать телеметрическую информацию, автоматизировать регулировку процессов производства, осуществлять непрерывный контроль за продажами и закупками товаров и т.п. Важной причиной для модернизации являются вопросы, связанные с информационной безопасностью. Локальная вычислительная сеть должна быть спроектирована таким образом, чтобы обеспечить надлежащую степень защищённости данных. Надо помнить, что от этого не должно страдать удобство пользователей и администраторов сети.

Для модернизации ЛВС ЧТУП «Медиамир» необходимо произвести анализ помещений предприятия, анализ локальной вычислительной сети, уже существующей в организации. На основе проведенного анализа сделать выводы и разработать новую структуру предприятия, обеспечить доступ к сетевым ресурсам, подобрать необходимое оборудование, удовлетворяющее требованиям данного предприятия, обеспечить защиту информации в сети.

А.С. Кулешов (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **Г.Л. Карасёва**, канд. физ.-мат. наук, доцент

СОЗДАНИЕ КОМПЛЕКСНОГО УЧЕБНОГО ПРИЛОЖЕНИЯ

В математике не последнее место занимают дифференциальное уравнения. Дифференциальное уравнение – уравнение, связывающее значение некоторой неизвестной функции в некоторой точке и значение

её производных различных порядков в той же точке. Дифференциальное уравнение содержит в своей записи неизвестную функцию, её производные и независимые переменные; однако не любое уравнение, содержащее производные неизвестной функции, является дифференциальным уравнением.

Разработано и реализовано комплексное учебное приложение для представления заданных уравнений в графической среде, рассмотрены фракталы, фазовые портреты и численные методы интегрирования функций.

Фрактал – геометрическая фигура, обладающая свойством самоподобия, то есть составленная из нескольких частей, каждая из которых подобна всей фигуре целиком. В математике под фракталами понимают множества точек в евклидовом пространстве, имеющие дробную метрическую размерность (в смысле Минковского или Хаусдорфа), либо метрическую размерность, отличную от топологической. Программа, представленная в работе, обрисовывает множество Мандельброта и множество Жюлиа.

Численное интегрирование представлено методами Эйлера, Эйлера-Коши и Рунге-Кутты. Отображение происходит путём выбора хранящейся в памяти программы функции, а также выбора метода.

Работа с фазовым пространством происходит путём выбора нужной нам системы уравнений (либо мы можем задать её вручную), выбора параметров отображения фазового портрета, а также выбора вида графика (точечный или линейный).

Разработана программа, которая, используя компьютерную графику, отображает решение дифференциального уравнения в графической среде. При написании работы были изучены соответствующие разделы современной математики, изучена среда разработки и написана программа.

Данное приложение разработано для людей, изучающих дифференциальные уравнения, в частности фракталы (на примере множества Мандельброта и множества Жюлиа). Также может использоваться преподавателем для наглядного объяснения. Удобный и простой интерфейс позволяет быстро разобраться в возможностях программы.

Основными возможностями приложения являются:

- выбор вида фрактала;
- выбор величины итерации;
- настройки цветовой гаммы;
- выбор степеней;
- выбор шага интегрирования;
- возможность сортировки полученных данных;
- выбор размера изображения;
- сохранение изображения на компьютере.

– возможность работать с численными методами интегрирования функций.

– построение фазовых портретов систем уравнений.

В разработанном приложении также присутствует функция автоматического увеличения определённой области выбранного нами фрактала. Для удобства пользователя она названа коротко «Автопилот». После нажатия кнопки происходит заданное программистом событие. Программа автоматически выбирает заданную позицию (заранее выбранные координаты) и начинает от неё пошагово увеличивать изображение, пока не дойдёт до заданного разработчиком окончания.

Вся отрисовка в приложении осуществляется с помощью средств только лишь центрального процессора, без подключения графических возможностей видеокарты. Таким образом, для быстрой обработки, которая позволяет видеть плавное увеличение изображения, нужна достаточно мощная машина. Логично сделать вывод, что чем больше будет выбран размер изображения и чем выше будет задана чёткость – тем дольше будет обрабатываться данные, дольше будет происходить построение и это негативно скажется на плавности увеличения изображения.

Все элементы программы написаны на языке программирования Delphi. Среда разработки – Borland Delphi 7.0.

Данное приложение разработано для людей, изучающих дифференциальные уравнения, в частности фракталы. Также может использоваться преподавателем для наглядного объяснения. Приложение позволяет наглядно объяснить, что такое фрактал на конкретном примере.

Н.В. Курник (БТЭУ ПК, Гомель)

Науч. рук. **С.В. Карпенко**, канд. эконом. наук, доцент

СОВЕРШЕНСТВОВАНИЕ ПРОЦЕССА ОПЛАТЫ В 1С

Предприятие ОАО «Гомельский Радиозавод» занимается выпуском и продажей продукции. Программным обеспечением информационной системы является «1С:Предприятие». Но т.к. стандартная конфигурация «1С:Предприятие» не отвечает требованиям работников бухгалтерии, были внесены некоторые изменения в типовые документы. Выполнена адаптация документа Счет на оплату к требованиям работников бухгалтерии.

Для реализации бизнес-процессов приобретения и продажи продукции на предприятии необходимо наличие документа Счет для оплаты материалов и услуг. Счет – это своего рода соглашение, по которому продавец (поставщик) указывает цену за свой товар или услуги, а покупатель принимает условия, перечисляя сумму. Система 1С

позволяет выполнять адаптацию бизнес-процессов и предоставляет соответствующие средства [1–5].

Помимо обязательных реквизитов (номер, дата, расчетный счет), счет может содержать дополнительную информацию, такую как:

- ✓ Сроки оплаты и поставки;
- ✓ Самовывоз товара;
- ✓ Уведомление о предоплате;
- ✓ Контактный телефон для уточнения деловых вопросов;
- ✓ Небольшую рекламу;
- ✓ Информацию о предстоящих скидках;

Перечислим основные особенности создаваемого документа.

1) Документ является необходимым для выставления счета на оплату материалов и услуг;

2) Помимо стандартных реквизитов имеется возможность ввода следующих данных:

- ✓ Установление нового типа цен (например, в договоре был установлен розничный тип цен, но для данной закупки он является оптовым или наоборот) (рисунок 1).

- ✓ При изменении типа цены пересчитывается надбавка (или скидка).

- ✓ Установка курса валюты для всех полей документа (в предыдущих версиях документа не всегда можно было произвести расчет в валюте (только при изменениях в структуру документа) (рисунок 2).

- ✓ Добавлено поле Адрес покупателя. Оно предназначено для того, чтобы можно было в дальнейшем печатать конверты перед рассылкой счетов.

- ✓ На вкладке услуги добавлено поле Содержание услуги. Оно предназначено для расшифровки услуги (в справочнике Номенклатура указано название услуги в рамках организации и иногда требуется расшифровка, чтобы было понятно, за что именно нужно платить).

- ✓ Реквизит сроки оплаты позволит выявлять контрагентов, которые вовремя оплачивают все услуги и материалы (рисунок 3).

- ✓ Срок поставки определит для покупателя дату поставки собственным транспортом поставщика.

- ✓ Галочка Самовывоз товара указывает на то, что покупатель сам забирает свой товар.

- ✓ Другая информация на счете (реклама, скидки) позволит покупателям ориентироваться и продолжать сотрудничество на взаимовыгодных условиях.

Адрес доставки: _____

Группа договоров: _____

Наименование: **Договор №4**

Вид договора: **С покупателя**

Ведение взаиморасчетов

Взаиморасчеты ведутся: **По д**

Расчеты в условных единица

Вид взаиморасчетов: _____

Установлен срок оплаты по

Срок действия договора: _____

Цены

Тип цен: **Оптовая**

Проводки: НДС включать

Цены и валюта

Новые значения

Тип цен: **Розничная**

Перезаполнить цены по типу цен

Валюта: **BYR** Курс: **1,0000**

1 BYR = 1 BYR

Пересчитать цены по валюте

Валюта взаиморасчетов: **BYR** Курс взаим.: **1,0000**

1 BYR = 1 BYR

Налого: Учитывать НДС Сумма вкл. НДС

Проводки: НДС включать в стоимость

Рисунок 1 – Установление типа цен

Перезаполнить цены по типу цен

Валюта: **BYR** Курс: **1,0000**

1 BYR = 1 BYR

Пересчитать цены по валюте

Валюта взаиморасчетов: **BYR** Курс взаим.: **1,0000**

1 BYR = 1 BYR

Рисунок 2 – Изменение курса валют

Цены и валюта... Действия [иконки] Советы ?

Номер: _____ от: 17.12.2013 0:00:00 Договор: **Договор №431**

Организация: **ОАО "Гомельский радиозавод"** Банковский счет: _____

Контрагент: **ЧУТПП "Джоркер"** **Срок оплаты:** _____

Склад: **Основной склад** Срок поставки: _____

Цель приобретения: _____ Контактный телефон: _____

Адрес доставки: _____

Товары (1 поз.) Услуги (0 поз.)

[иконки] Подбор Изменить

№	Номенклатура	Количество	Цена	Сумма	% НДС	Сумма
1	Лак	10,000			20% / 120%	

Рисунок 3 – Срок оплаты и поставки контрагентов

```

□ Процедура РассчитатьТорговуюНадбавку(СтрокаТабличнойЧасти)
    СуммаСНДС = (СтрокаТабличнойЧасти.Сумма + ?(СуммаВключаетНДС, 0, СтрокаТабличнойЧа
    Количество = СтрокаТабличнойЧасти.Количество;
    СуммаНДС = СтрокаТабличнойЧасти.СуммаНДС;
    СуммаБезНДС = СуммаСНДС - СуммаНДС;
    Цена = СуммаБезНДС / ?(Количество = 0, 1, Количество);
    Если СтрокаТабличнойЧасти.ЦенаИзготовителя > 0 Тогда
        ОптоваяНадбавка = Окр((Цена - СтрокаТабличнойЧасти.ЦенаИзготовителя) /
        СтрокаТабличнойЧасти.ЦенаИзготовителя*100, 1, 1);
        СтрокаТабличнойЧасти.ОптоваяНадбавка = ОптоваяНадбавка;
    Иначе
        СтрокаТабличнойЧасти.ОптоваяНадбавка = 0;
    КонецЕсли;
КонецПроцедуры
□ Процедура ТоварыЦенаПриИзменении(Элемент)
    СтрокаТабличнойЧасти = ЭлементыФормы.Товары.ТекущиеДанные;
    // Рассчитать реквизиты табличной части.
    ОбработкаТабличныхЧастей.РассчитатьСуммуТабЧасти(СтрокаТабличнойЧасти, ЭтотОбъект)
    ОбработкаТабличныхЧастей.РассчитатьСуммуНДСТабЧасти(СтрокаТабличнойЧасти, ЭтотОбъект)
    РассчитатьТорговуюНадбавку(СтрокаТабличнойЧасти);
КонецПроцедуры // ТоварыЦенаПриИзменении()
□ Процедура КонтрагентПриИзменении(Элемент)
    // Выполняем общие действия для всех документов при изменении Контрагент.
    УправлениеВзаиморасчетами.ПриИзмененииЗначенияКонтрагента(ЭтотОбъект, мСписокВидовДокументов)
    ПриИзмененииДоговора();
    ИзменитьАдресДоставки();
КонецПроцедуры // КонтрагентПриИзменении()

```

Рисунок 4 – Процедуры расчета торговой надбавки, формирование отчета и адреса доставки

Таким образом, можно сделать вывод, что созданный документ предоставляет работникам бухгалтерии расширенные возможности, будет предоставлять возможность для составления более подробной отчетности и ее анализа. Реализуется главный принцип процессного управления – ориентация на клиента. Обеспечивается удобство работы пользователя.

Литература

- 1С: Предприятие 8.0 Конфигурация «Управление торговлей». Редакция 10.2.: Методика учета торговых операций. [Текст]– М.: Фирма 1С, 2014.
2. Радченко М. Г. 1С: Предприятие 8.1. Практическое пособие для разработчика. Примеры и типовые приемы / М.Г. Радченко. – М.: ООО «1С-Паблишинг», СПб: Питер. 2014. – 512с.: ил.
3. Самоучитель по 1С. [Электронный ресурс] / – Режим доступа: <http://www.1c-h.ru/?p=2902/> – Дата доступа: 08.03.2014.
4. 1С:Предприятие 8. [Электронный ресурс] / – Режим доступа: <http://v8.1c.ru/metod/books/> – Дата доступа: 05.03.2014.
5. 1С: предприятие 8.[Электронный ресурс] / – Режим доступа: http://v8.1c.ru/beta_ma/ – Дата доступа: 05.03.2014.

Р.В Кучеренко (УО «ГГУ им. Ф. Скорины», Гомель)

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

АНАЛИЗ СОВРЕМЕННЫХ СИСТЕМ УДАЛЕННОГО МОНИТОРИНГА ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ

В любой сети, где есть больше, чем один сервер, очень полезно иметь перед глазами полную картину происходящего. В крупных сетях, где количество хостов переваливает за несколько десятков, следить за каждым в отдельности – непосильная задача для администраторов. Для облегчения задачи наблюдения применяются системы удаленного мониторинга вычислительных ресурсов.

Системы мониторинга сети позволяют автоматизировать, упростить и ускорить процесс получения информации о состоянии сетевого оборудования, что необходимо каждому интернет-провайдеру. Если есть возможность в режиме реального времени наблюдать за состоянием оборудования, то появляется возможность оперативного реагирования и устранения проблем, которые могут возникнуть в сети.

Систем удаленного мониторинга множество. Рассмотрим наиболее известные из них:

Ganglia – масштабируемая распределенная система мониторинга кластеров параллельных и распределенных вычислений и облачных систем с иерархической структурой. Позволяет наблюдать статистику и историю вычислений в реальном времени для каждой из наблюдаемых машин;

Sastі – система позволяет строить графики при помощи RRDtool. Sastі собирает статистические данные за определённые временные интервалы и позволяет отобразить их в графическом виде. Преимущественно используются стандартные шаблоны для отображения статистики по загрузке процессора, выделению оперативной памяти, количеству запущенных процессов, использованию входящего/исходящего трафика;

Nagios – программа мониторинга компьютерных систем и сетей с открытым кодом. Предназначена для наблюдения, контроля состояния вычислительных узлов и служб, оповещает администратора в том случае, если какие-то из служб прекращают или возобновляют свою работу;

Zabbix – свободная система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования.

Критерии отбора, предъявляемые системам удаленного мониторинга вычислительных ресурсов:

- прозрачная масштабируемость – способность системы, сети или процесса справляться с увеличением рабочей нагрузки;
- удаленный мониторинг нескольких тысяч устройств;
- обработка нескольких тысяч уведомлений в день.

Системы Ganglia и Cacti слишком просты и не способны выполнить предъявленные требования, поэтому рассмотрим системы Nagios и Zabbix.

Система Nagios:

- система не отказоустойчива и масштабируется с помощью переноса части проверок на отдельные сервера;
- все изменения конфигурации выполняются правкой файлов конфигурации с последующим перезапуском Nagios;
- слишком большой интервал между проверками и замерами параметров;
- база данных, в которой хранится информация по мониторингу, усредняет параметры, поэтому их точное значение за прошлый месяц посмотреть не получится.

Система Zabbix:

- вся конфигурация хранится в базе, управляется через web-интерфейс;
- единая точка доступа для пользователей;
- разграничение доступа к данным и конфигурации;
- возможность настройки интервала между замерами с минимальным значением – 1 секунда;
- с серверов собираются не результаты проверок, а количественные характеристики работы, которые анализируются на стороне сервера;
- время хранения данных ограничено лишь дисковым пространством.

Структура Zabbix:

1 Zabbix сервер – это ядро программного обеспечения Zabbix. Сервер может удаленно проверять сетевые сервисы, является хранилищем, в котором хранятся все конфигурационные, статистические и оперативные данные, и он является тем субъектом в программном обеспечении Zabbix, который оповестит администраторов в случае возникновения проблем с любым контролируемым оборудованием.

2 Zabbix прокси – собирает данные о производительности и доступности от имени Zabbix сервера. Все собранные данные заносятся в буфер на локальном уровне и передаются Zabbix серверу, к которому принадлежит прокси-сервер. Zabbix прокси является идеальным решением для централизованного удаленного мониторинга мест, филиалов, сетей, не имеющих локальных администраторов. Он может быть также

использован для распределения нагрузки одного Zabbix сервера. В этом случае, прокси только собирает данные, тем самым на сервер ложится меньшая нагрузка на ЦПУ и на ввод/вывод диска.

3 Zabbix агент – осуществляет контроль локальных ресурсов и приложений на сетевых системах, эти системы должны работать с запущенным Zabbix агентом. Zabbix агенты являются чрезвычайно эффективными из-за использования родных системных вызовов для сбора информации о статистике.

4 Веб-интерфейс – является частью Zabbix сервера, и, как правило, запущен на том же физическом сервере, что и Zabbix сервер.

Исходя из вышеописанного, можно сделать вывод, что Zabbix – наиболее подходящая система мониторинга. Она включает в себя весь функционал, необходимый для работы системы удаленного мониторинга, бесперебойно следящей, максимально точно показывающей состояние сетевого оборудования в режиме реального времени и хранящей историю, к которой максимально легко и быстро получить доступ.

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

Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ РЕГИСТРАЦИИ

При проектировании Web-сайта были проанализированы современные Web-технологии, позволяющие создавать интерактивные Web-страницы. Разработка системы регистрации производится на языке PHP с использованием баз данных MySQL. Визуальная часть разрабатывается непосредственно с помощью HTML и CSS.

Форма для логина и регистрации объединены в одну, где пользователи будут указывать свой email. Если указанного почтового адреса нет в базе, создается новая запись. Генерируется случайный ключ и отправляется на почту пользователя в виде ссылки, которая активна в течении 10 минут.

Наже приведен пример методов используемых при создании:

- loginOrRegister(\$email) – проверка на существующего пользователя;
- create(\$email) – создание нового пользователя;
- generateToken() – генерация случайного ключа;
- login() – авторизация пользователя;
- logout() – выход пользователя.

При нажатии на ссылку пользователь переходит на наш сайт. Система сама авторизует пользователя по переданному в ссылке ключу.

Преимуществами такого подхода является то, что, нет необходимости в хранении и валидации паролей, нет необходимости в системе восстановления паролей, процесс регистрации очень прост и привлекателен.

Также, после регистрации, для каждого пользователя будет создан его личный кабинет. В личном кабинете будет доступна информация о пользователе, его аватар, день рождения и различная другая информация.

Как дополнение, для каждого пользователя, будет реализована возможность общения друг с другом по средствам личных сообщений. Отправить письмо можно будет со страницы просмотра данных о пользователе, нажав на соответствующую кнопку. Форма отправки письма будет содержать поля для темы и самого текста сообщения. Информацию о отправленных и входящих письмах можно будет посмотреть из личного кабинета.

А.П. Лапушкина (УО «ГГУ им. Ф. Скорины», Гомель)

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

СВЯЗЬ СИСТЕМЫ 1С:ПРЕДПРИЯТИЕ 7.7 С ДРУГИМИ ПРОГРАММАМИ

Успешное развитие современного бизнеса во многом зависит от широкого применения новейших информационных технологий, позволяющих обрабатывать информацию любого вида с наибольшей эффективностью. Состав программ системы «1С: Предприятие» ориентирован на актуальные потребности отечественных предприятий. Очень важным преимуществом «1С:Предприятия» является открытость системы.

1С:Предприятие 7.7 содержит разнообразные средства для связи с другими программами. Для руководителя, принимающего решение о выборе средства автоматизации, достаточно важно быть уверенным, что система не будет для предприятия «черным ящиком», что существует реальная возможность понять работу системы и, при необходимости, изменить.

Программные продукты системы 1С:Предприятие 7.7 содержат разнообразные средства для связи с другими программами и аппаратными средствами.

Обмен файлами. Возможность обмена файлами различных форматов, реализованная в системе 1С:Предприятие, позволяет легко интегрироваться практически с любыми системами. Также позволяет сохранение печатных форм в форматах Microsoft Excel и HTML.

Текстовые файлы. Работа с текстовыми файлами – наиболее простой механизм обмена данными. Он может быть использован для решения

самых разнообразных задач. Его основное преимущество – простота освоения и удобное текстовое представление информации.

Система 1С:Предприятие позволяет, создавать и записывать строковые значения в файл, считывать информацию из файла с последующим ее разбором по строкам устанавливать режим кодировки (Windows или DOS).

Файлы DBF. Механизм XBase, встроенный в 1С:Предприятие, позволяет обрабатывать информацию в файлах DBF непосредственно из встроенного языка. Возможно практически любое манипулирование данными. Применение формата DBF позволяет, например, легко подключать к системе 1С:Предприятие торговое оборудование, так как многие драйверы и программы используют именно формат DBF для хранения своей информации.

Формат XML. В системе 1С:Предприятие XML-формат используется как для обмена коммерческой информацией по стандарту CommerceML и обмена данными между различными информационными базами (различных конфигураций), так и для реализации обмена XML-документами других форматов.

Специально для работы с XML-документами во встроенном языке 1С:Предприятия поддерживается объект компоненты, реализованный как оболочка MSXMLParsers. С его помощью можно расширить или переопределить состав свойств и методов объектов, входящих в состав DOM (Document Object Model) модели, реализованных в анализаторе XML-документов от фирмы Microsoft, а также реализовать модель последовательного доступа к XML-данным (Simple API for XML).

OLE, OLE Automation. Система 1С:Предприятие может быть запущена внешним приложением в качестве OLE-Automation сервера и предоставляет доступ ко всем своим объектам, атрибутам и методам своего глобального контекста. Кроме того OLE-сервер 1С:Предприятие имеет дополнительные методы, с помощью которых можно выполнить последовательность операторов или вычислить выражение, заданное на встроенном языке 1С:Предприятие.

Являясь OLE-Automation клиентом, система 1С:Предприятие предоставляет возможность управлять другими программами непосредственно при помощи операторов встроенного языка. Например, можно запустить программу MS Excel с одновременным выводом в нее результатов расчета, сделанных в системе 1С:Предприятие.

Поскольку 1С:Предприятие является одновременно OLE Automation сервером и OLE Automation клиентом, то предоставляется возможность полностью интегрироваться с теми программными продуктами, которые поддерживают механизм OLE Automation.

Технология внешних компонент. «Технология создания внешних компонент» разработана для решения специальных задач интеграции, в

которых требуется тесное взаимодействие между 1С:Предприятием и другими программами.

Данная технология позволяет создавать программы, которые будут динамически подключаться и тесно взаимодействовать с системой 1С:Предприятие, расширяя ее возможности.

Технология внешних компонент позволяет решать широкий спектр специальных задач, в частности, задачи, связанные с использованием различного торгового оборудования совместно с 1С:Предприятием.

Внешние компоненты позволяют расширять встроенный язык 1С:Предприятия 7.7 новыми агрегатными объектами. Структуры механизмов внешних компонент максимально приближены к внутренним структурам 1С:Предприятия 7.7, что повышает эффективность работы.

Внешняя компонента может порождать события, обрабатываемые в предопределенной процедуре языка ОбработкаВнешнегоСобытия. Это позволяет подключать к 1С:Предприятию 7.7 сканеры и другие устройства, требующие асинхронного обмена данными.

Внешние компоненты могут добавлять свои страницы свойств в диалог параметров 1С:Предприятия 7.7. Таким образом, торговое оборудование может включаться в систему и управляться стандартным для 1С:Предприятия 7.7 способом.

При сохранении своих параметров внешняя компонента может использовать механизмы 1С:Предприятия 7.7. При работе состояние внешней компоненты может отражаться в строке состояния 1С:Предприятия 7.7.

Внешние компоненты могут разрабатываться пользователями программ системы 1С Предприятие и независимыми фирмами на языках MS Visual C++, MS Visual Basic, Borland Delphi.

Работа с торговым оборудованием. Для пользователей системы программ 1С Предприятие 7.7 существует набор решений по подключению различного оборудования, применяемого в торговле и складском деле: контрольно-кассовых машин, POS-терминалов, сканеров и принтеров штрих-кодов, дисплеев покупателя, электронных весов, терминалов сбора данных. Эти решения основаны, как было указано выше, на технологии внешних компонент, а также на технологии режимов экспорта-импорта информации.

Использование этих средств позволяет:

- используя встроенный язык 1С:Предприятие 7.7, формировать отчеты и графики в Microsoft Excel, производить выгрузку информации в Microsoft Word и др;
- получать доступ к данным 1С:Предприятие 7.7 из других программ;
- размещать в документах и отчетах рисунки и графики.

Р.А. Лебедев (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

СОЗДАНИЕ ИГР С ИСПОЛЬЗОВАНИЕМ MICROSOFT XNA

Microsoft XNA – набор инструментов суправляемой средой времени выполнения, созданный Microsoft, облегчающий разработку и управление компьютерными играми. Пакет включает в себя обширный набор библиотек классов, специфичных для разработки игр и поддерживающий максимальное повторное использование кода. Игры XNA пишутся для среды времени выполнения, поэтому они могут запускаться на любой платформе, поддерживающей XNA Framework, что делает подобный набор универсальным.

Наибольший интерес при создании платформера, как и любого другого приложения, представляет файл `Game1.cs`, в котором определен класс `Game1`, где и разрабатывается приложение. В классе `Game1` переопределены следующие методы `Game`:

- `Initialize` – вызывается единожды, для инициализации ресурсов до начала игры;

- `LoadContent` – вызывается единожды, используется для загрузки контента (спрайты и т.д.);

- `UnloadContent` – вызывается единожды, используется для выгрузки контента;

- `Update` – в этом методе реализуется логика игры, обработка событий клавиатуры или джойстика, проигрывание аудио и т.д.;

- `Draw` – вызывается для прорисовки игрового поля.

В папку `Content` добавляются игровые ресурсы, в данном случае картинки фона, платформ, и самого главного персонажа. В методе `LoadContent` добавленные ресурсы используются для создания игровых объектов. Отрисовка созданных объектов осуществляется в методе `Draw`. В этом методе рисуется фоновое изображение, персонаж, препятствия и платформы, имеющие одинаковое изображение. Обработка движения и столкновения объектов, а также их реакция на действия пользователя происходит в методе `Update`. При каждом вызове `Update` проверяется состояние клавиш перемещения влево и вправо, и в случае, если какая-нибудь из них нажата, персонаж перемещается в соответствующую сторону. Во время выполнения программы методы `Update` и `Draw` вызываются с достаточно высокой частотой. В результате получаем вариативное изменение положения персонажа и других объектов и мгновенную прорисовку произошедших изменений.

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

Науч. рук. **А.В. Воруев**, канд. тех. наук, доцент

ОРГАНИЗАЦИЯ РЕЗЕРВИРОВАНИЯ ФУНКЦИИ DEFAULT GATEWAY С ПОМОЩЬЮ ТЕХНОЛОГИИ HSRP

HSRP (Hot Standby Router Protocol) был разработан компанией Cisco Systems. В данный момент в качестве основного стандарта, описывающего данный протокол, принят документ RFC 2281, написанный представителями Cisco Systems и Juniper Networks. Основная задача и предназначение данного протокола состоит в том, чтобы добиться практически 100 % доступности и отказоустойчивости первого хопа от отправителя (также иногда называемый маршрут по умолчанию или шлюз последней надежды). HSRP позволяет нескольким маршрутизаторам в одной локальной сети делить между собой один виртуальный IP и MAC-адрес.

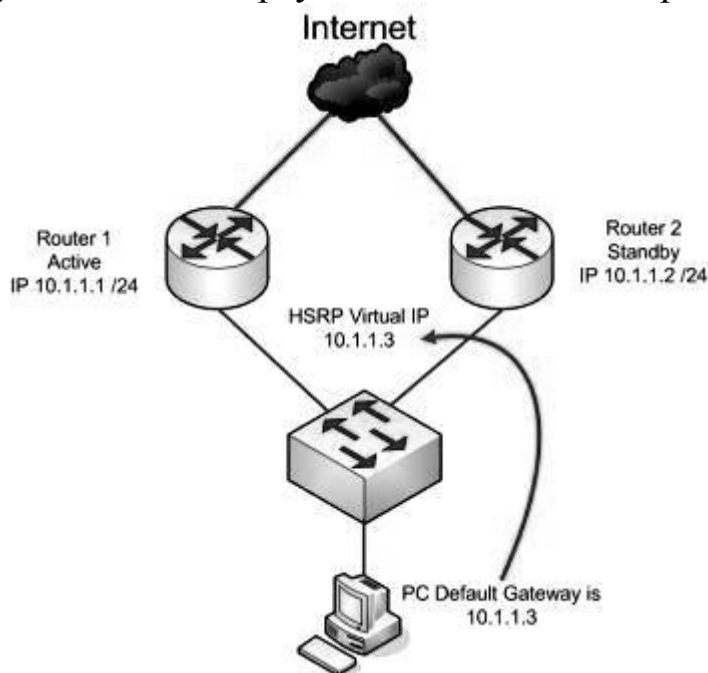


Рисунок 1 – Алгоритм предоставления IP и MAC-адрес

Из группы маршрутизаторов, сконфигурированных в группе HSRP, один маршрутизатор выбирается в качестве активного маршрутизатора, а другой в качестве резервного маршрутизатора. Активный маршрутизатор берет на себя роль пересылки пакетов, посланных на виртуальный IP-адрес. Если активный маршрутизатор выходит из строя, резервный маршрутизатор берет на себя в качестве нового активного маршрутизатора. Все команды HSRP настраиваются в режиме конфигурации интерфейса маршрутизатора.

Для настройки HSRP выполняется команда

standby [group-number] ip [ip-address] и указывается виртуальный IP-адрес для группы.

Далее используется тот же номер группы и виртуальный IP-адрес на других маршрутизаторах, которые обеспечивают резервирование для этой группы. Это единственная команда, что необходимо для включения HSRP.

Другие команды HSRP являются необязательными и могут быть настроены в случае необходимости.

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

Науч. рук. **А.В. Воруев**, канд. тех. наук, доцент

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ПОСТРОЕНИЯ ВЫСОКОСКОРОСТНЫХ СЕТЕЙ

Современные высокоскоростные технологии связи предоставляют широкие возможности для построения мультисервисных сетей передачи данных и сетей нового поколения NGN.

Высокоскоростные технологии разрабатываются для того, чтобы создавать локальные компьютерные сети с высокой пропускной способностью. К таким технологиям относятся широко распространенные технологии Fast Ethernet, Gigabit Ethernet, Myrinet и ATM.

Проведем краткий обзор технологий.

Fast Ethernet (100BaseT). Представляет собой дальнейшую модификацию стандарта Fast Ethernet (10BaseT) с пропускной способностью от 10 Мбит/с до 100 Мбит/с. Стандарт 100BaseT обладает большими полномочиями и поддерживает протокол CSMA/CD – обработки множественного доступа с опознаванием несущей и обнаружением конфликтов. Используется в локальных сетях с решениями Cisco Systems.

Преимущества Fast Ethernet (100BaseT):

возможность использования в сетевой инфраструктуре кабелей различных типов;

хорошая совместимость кабельной инфраструктуры и ПО с аналогичными элементами сетей, построенными на протоколе Fast Ethernet (10BaseT);

быстрота миграции сетей с Fast Ethernet (10BaseT) на Fast Ethernet (100BaseT);

стандарт поддерживает три модификации для обеспечения работы с разными видами кабелей Fast Ethernet: 100BaseTX, 100BaseT4 и 100BaseFX..

Gigabit Ethernet. Технология Gigabit Ethernet – дальнейшее развитие стандартов 802.3 для сетей Ethernet с пропускной способностью 10 и 100 Мбит/с. Технология специально разработана для создания

высокоскоростных оптоволоконных сетей передачи данных. Преимуществами Gigabit Ethernet являются:

- совместимость с существующими сетями Ethernet;
- поддержка дуплексного и полудуплексного доступа к среде передачи.

Asynchronous Transfer Mode (ATM). Технология асинхронной передачи данных предназначена для устранения проблем в сети, связанных с возникновением больших задержек с сети. Продукты АТМ обеспечивают поддержку передачи данных на скоростях от 25 Мбит/с по неэкранированной и экранированной витым парам, 155 Мбит/с по среде UTP и ВОЛС, до 4,8 Гбит/с на линиях ВОЛС.

В отличие от традиционных сетевых технологий Ethernet, технология АТМ не имеет физических или архитектурных ограничений. Современные продукты АТМ обеспечивают поддержку передачи данных на скоростях от 25 Мбит/с по неэкранированной витой паре (UTP) и экранированной витой паре (STP), 155 Мбит/с по среде UTP и оптоволокну, и от 622 Мбит/с до 4,8 Гбит/с по оптоволокну.

Быстрая аппаратная коммутация, позволяет оперативно маршрутизировать потоки данных в сети еще на уровне проектирование СКС. Кроме того, технология АТМ предполагает использование единого способа передачи данных по сети, что позволяет применять данную технологию при построении сетей различного типа.

Единовременная передача по сети данных различного типа: голоса, видео и традиционных данных.

Гибкая полоса пропускания, позволяющая использовать технологию АТМ для подключения наиболее требовательных к ширине полосы пропускания приложений: мультимедиа и средств видеокommunikаций.

Е.С. Левинская (УО «ГГУ им. Ф. Скорины», Гомель)

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

МУЛЬТИПЛАТФОРМЕННЫЙ ИНСТРУМЕНТ ДЛЯ РАЗРАБОТКИ ИГР UNITY

Unity – это инструмент для разработки различных приложений и игр. Unity является мультиплатформенным инструментом, то есть созданные с его помощью приложения работают под операционными системами Windows, OS X, Windows Phone, Android, Apple iOS, Linux, а также на игровых приставках Wii, PlayStation 3 и Xbox 360. Стоит отметить, что также существует возможность создавать Интернет-приложения с по-

мощью специального подключаемого модуля к браузеру Unity. Приложения, созданные с помощью Unity, поддерживают уже готовые классы, процедуры, функций, структур и констант, то есть программист может воспользоваться уже готовыми решениями при работе над своим проектом.

Unity чаще всего используется для разработки двух- и трёхмерных игр. Немаловажным является то, что в Unity встроен набор приемов для устранения проблем совместимости. Unity стабильно поддерживает большинство возможных вариантов сочетания аппаратной и программной части. Это значит, что тот, кто работает с данным инструментом, может не отвлекаться на технические аспекты своей работы, а больше внимания уделять непосредственно творчеству.

Unity содержит полный набор качеств профессионального приложения:

Интегрированный редактор. Все действия производятся через простой интерфейс пользователя. Unity3D полностью интегрирован в Среду Разработки Unity. Тесная интеграция позволяет прямо в редакторе получить всё то, на что будет способна выпущенная игра. Простая, визуальная и интуитивная природа Редактора делает конструирование игр очень простым.

Импорт ресурсов. Unity поддерживаются все основные форматы файлов и практически все приложения для создания графики. Импортируются все основные файловые форматы, и любое графическое приложение может быть использовано с Unity. Unity может импортировать 3D модели, кости и анимацию практически из всех 3D приложений. Это значит, что как только произошло какое либо изменение модели в Maya, Cinema 4D, Cheetah3D или Blender и Unity проверит произведенные изменения и отразит во всем вашем проекте в режиме реального времени.

Графика. Основным плюсом Unity при создании графического контента является скорость. При достаточных ресурсах аппаратной составляющей Unity рендерит миллионы полигонов в секунду. Рендеринг оптимизирован для минимизации количества изменений состояния, принимая во внимание источники света и тени. Визуальное создание и манипуляция системой частиц достаточно просты даже для обычного пользователя. С их помощью можно создавать дождь, искры, пылевые столбы.

Тени и свет. Unity предоставляет высоко-оптимизированную систему освещения с тенями реального времени. При работе со светом и тенью пользователь может не заботиться о технических деталях. Разнообразные тени реального времени могут отбрасываться от любого источника света в созданной сцене.

Ландшафты. Unity предоставляет встроенные в редактор инструменты для создания и редактирования участков местности всех видов. Немаловажным является то, что даже огромные, густо заполненные

растительностью ландшафты, хорошо работают на низкопроизводительных компьютерах. Генерируйте карту освещенности для ландшафта в любое время. Пользователь может создать достаточно живую и впечатляющую картину.

Шейдеры. Шейдер – алгоритм, который модифицирует текстуру объекта в соответствии с теми условиями, в которых объект находится. Шейдеры могут быть использованы для рисования поверхности из трёхмерной керамической плитки на абсолютно плоской поверхности. Шейдерная система Unity сочетает простоту использования, гибкость и производительность. Все встроенные шейдеры прекрасно работают с любым типом света. Если этого не достаточно, пользователь может написать свои собственные шейдеры на мощном языке Unity под названием ShaderLab с использованием Cg и GLSL. Unity поставляется с 40 шейдерами, начиная от простых: Diffuse (диффузный), Glossy (блестящий) и т.д. И заканчивая весьма сложными: Self Illuminated Bumped Specular (самосветящийся зеркальный с неровностями) и т.д.

Физика. В Unity встроен физический движок Ageia PhysX. Unity полностью поддерживает физику Твёрдого тела (rigid body). Твёрдые тела действуют под воздействием сил, сталкиваются как в реальной жизни. Также Unity поддерживает Шарниры (Hinges), пружины, шаровые шарниры (ball-sockets).

Анимация. Каждая часть каждого игрового объекта (расположение, вращение, масштаб, цвет и т.д.) может быть анимирована с помощью интегрированного интерфейса для покадровой анимации. И это в дополнение к полноценной системе анимации персонажей.

Сценарный язык. Unity поддерживает три языка создания сценариев (кратких описаний действий, выполняемых системой): JavaScript, C# и диалект Python'a под названием Boo. Все три языка могут использовать лежащие в основе .NET библиотеки, которые поддерживают базы данных, регулярные выражения, XML, доступ к файлам и работу с сетью. Игровая логика работает на платформе .NET с открытым исходным кодом.

Работа с сетью. Переход от однопользовательских к мультиплеерным режимам для игры можно осуществлять в режиме реального времени при создании игры. Unity делает трудоемкую и запутанную работу с сетью в реальном времени легкой и простой в использовании. Превращение однопользовательской игры в сетевую многопользовательскую требует всего пары изменений в имеющихся сценариях. Unity делает всю сложную работу за пользователя, обрабатывая технические детали, связанные с сетью.

Аудио и Видео. С помощью Unity пользователь может осуществить смешивание 3д-графики реального времени с потоковым аудио и видео.

Любые звуки, записанные пользователем можно интегрировать в игровую сцену. Размещать позиционные аудио-источники можно где угодно. Для максимально возможной полноты эффекта присутствия существует настройка дистанционного затухания звука.

Документация. Для тех, кто хочет работать с Unity существуют пошаговые уроки, документация и проекты-примеры. Подробное руководство пользователя по Unity и справка доступны как в онлайн, так и в оффлайн варианте. В справке по скриптам работает поиск, в ней содержится множество наглядных примеров. У Unity активное, дружелюбное и бурно растущее сообщество.

Сервер ресурсов Unity. Сущность разработки игр – это работа в команде. Сервер ресурсов Unity это дополнение, которое добавляет контроль версий в функционал Unity. Наблюдать за изменением ресурсов\версий можно прямо внутри Редактора Unity. Если файлы подвергаются изменениям, их статус немедленно обновляется. Переименование и перемещение ресурсов не создает каких-либо препятствий для непрерывного рабочего процесса.

Размещение одним кликом. Unity поддерживает широкий диапазон платформ для запуска законченного проекта. Созданные с помощью Unity игры могут проигрываться внутри браузера благодаря Unity Web Player Plug-in. Плагин работает на всех современных браузерах, включая Internet Explorer, Firefox, Safari, и большинство браузеров, основанных на ядре Mozilla.

Стандартные версии Unity являются абсолютно бесплатными и доступными для любого пользователя. Помимо бесплатной, существуют четыре сборки – стандартная Unity, Unity iOS Pro (для разработки игр под iOS), Android Pro и командная лицензия. Они отличаются стоимостью. В бесплатной версии многое ограничено, однако возможность создавать игры сохраняется, но их можно распространять, если доход с игры не превышает \$5000.

По состоянию на 1 февраля 2014 г. число зарегистрированных пользователей выросло Unity до 2,5 млн. Причем 500 тыс. из них – активные разработчики.

Unity прошла длительное и разностороннее тестирование. С помощью этого инструмента любой желающий может начать создавать свои игры.

Е.В. Левшунов (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **Е.Е. Пугачева**, ассистент

РАЗРАБОТКА ФУНКЦИОНАЛА СОЗДАНИЯ ТАБЕЛЕЙ УЧЕТА РАБОЧЕГО ВРЕМЕНИ В СРЕДЕ LOTUS NOTES 8.5

Начисление заработной платы в организациях производится на основании различных первичных документов в зависимости от отраслей экономики и видов производств. Ими могут быть наряды на сдельную работу, маршрутные листы, путевые листы, табели учета рабочего времени и другие. Наличие табелей учета рабочего времени важно для учета трудовых затрат и в целях бухгалтерского, и в целях налогового учета. Что делает последние крайне важными документами в сфере бухгалтерского учета, перекадывая на плечи работников большую ответственность при заполнении табелей учета рабочего времени.

Каждый табель включает в себя таблицу отработанных и неотработанных часов сотрудника с отображением рабочих и не рабочих кодов явок, отпусков, больничных, командировок и так далее. Стремительный прогресс ЭВМ позволяет легко создать информационную систему, которая позволила бы хранить все эти коды (справки о больничном, заявки на отпуска и командировки) для каждого сотрудника, позволив автоматизировать создание табелей учета рабочего времени.

В настоящее время существует множество программных продуктов, которые позволяют создавать мощные информационные системы. Все эти программы делятся на множество видов и категорий, но обратим внимание на документа-ориентированные СУБД, и как вариант, на систему LotusNotes.

Lotus Notes – хорошо известный программный продукт, выполненный в архитектуре клиент-сервер и относящийся к классу Groupware, то есть предназначенный для поддержки групповой работы на рабочих станциях, объединенных в локальную сеть либо подключающихся к серверу по IP любым доступным способом. Ядром системы служит СУБД, хранящая данные в формате Notes Object store format (типичное расширение для файлов базы .nsf), и обеспечивающая функционирование многочисленных приложений, необходимых для ежедневной сетевой работы любого офиса. Среди них поддержка электронной почты для клиентов Lotus, SMTP/POP3, IMAP, X.400, а также документооборота предприятия, основанного на совместном использовании баз данных с управляемыми правами доступа. Начиная с версии 4.5, серверное ПО Lotus, называемое Lotus Domino, включило в себя технологию, позволяющую превратить Lotus Notes в сервер приложений Интернет. Сочетание поддержки открытых сетевых стандартов и протоколов Интернет с мощными средствами разработки и управления базами данных в одном серверном продукте открыло дорогу для создания новых бизнес-приложений Интернет. Таким образом, технология Domino может рассматриваться как:

- многоплатформенная реализация Web-сервера;

- среда для разработки Web-приложений;
- возможность работы с базой данных Lotus Notes с помощью Web-браузера в качестве альтернативного клиента.

Web-сервер Domino, работая как одна из задач на сервере базы данных Lotus Notes, позволяет ему функционировать также и в качестве HTTP-сервера. На практике это означает, что при получении соответствующего запроса на доступ к базе через Интернет/интранет задача Web достаточно оперативно конвертирует все элементы дизайна базы – формы, представления, навигаторы, связи, документы и другие – в HTML формат и отображает результаты запроса через окно Web-браузера клиента. В большинстве случаев этот результат будет эквивалентен тому, что увидит клиент Lotus Notes через свой стандартный интерфейс. При этом Web-сервер Domino сохраняет возможность обслуживать расположенные на нем «чистые» HTML, Java, CGI и другие стандартные типы Internet-ресурсов.

Работа с документами баз данных через Web позволяет использовать всю их спроектированную в Lotus Notes функциональность, включая редактирование и создание новых документов на сервере, прием и отправку почты, участие в дискуссиях, полнотекстовый поиск и сортировку, репликации, защиту данных от несанкционированного доступа. При этом клиенты Lotus Notes с соответствующими правами (которые регулируются менеджером базы в интервале от прав менеджера до полного закрытия доступа для конкретного или анонимного клиента) имеют возможность изменять или создавать новые документы баз данных. При правильно организованном Web-интерфейсе, сделанные изменения немедленно отражаются на соответствующем Web-сайте.

Это тем более важно, поскольку быстроедействие системы, которое видит разработчик в процессе отладки создаваемых алгоритмов, совершенно не имеет ничего общего с реальным быстроедействием системы при полной нагрузке. Простой пример: разработчик создал документ и убедился, что он проводится быстро и правильно. Но когда несколько пользователей стали одновременно интенсивно создавать эти документы, выяснилось, что их одновременная работа просто невозможна, поскольку структура объектов метаданных, задействованных в алгоритме проведения, такова, что, проводя эти документы, пользователи блокируют одни и те же записи в таблицах базы данных.

Е.В. Левшунов (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **Е.Е. Пугачева**, ассистент

ОПИСАНИЕ ОСНОВНЫХ СВОЙСТВ IBM LOTUS NOTES ДЛЯ ИНТЕГРАЦИИ LOTUS NOTES С КАЛЕНДАРЯМИ GOOGLE

Выбор продуктов и сервисов, связанных с календарями, весьма широк. Среди них: Lotus Notes Calendar, Microsoft® Outlook Calendar, Google Calendar и Yahoo! Calendar. Можно использовать одновременно несколько продуктов и сервисов. Например, можно установить Lotus Notes Calendar для связи с клиентами и коллегами, а Google Calendar – в качестве органайзера личных дел, например, дней рождения или вечеринок.

В этом случае для отслеживания всех запланированных событий нужно будет переключаться с одного продукта или сервиса на другой, что достаточно неудобно. Уделим внимание созданию составных приложений Lotus Notes 8, которые интегрируют различные календари из распределенных источников, что дает интегрированное отображение, и в нем можно отследить любое событие в распределенных календарях.

LNC используется для составления в одном месте календарных записей: расписания и отслеживания встреч, мероприятий, занимающих целый день, важных событий, напоминаний и уведомлений о событиях. LNC является частью почтового приложения Lotus Notes.

Google Calendar – онлайн-календарный сервис совместного использования. Используя Google Calendar, можно добавлять события и приглашения, сообщать их своим друзьям и родственникам (или не сообщать), а также искать в сети события, в которых можно участвовать.

При установке Lotus Notes 8 нужно убедиться, что установлен редактор Composite Application Editor. Создаем почтовую базу данных Lotus Notes при помощи почтового шаблона Lotus Notes 8.

Для интеграции Lotus Notes Calendar Navigator, Lotus Notes Calendar View, Google Calendar Navigator и Managed Browser используется составное приложение. Все четыре компонента связаны между собой при помощи property broker (рисунок 1). Google Calendar Navigator извлекает информацию из Google Calendar Atom, добавляет в информационный файл схему Extensible Stylesheet Language Transformation (XSLT) и выдает Managed Browser команду для перевода.

Редактор Lotus Notes 8 Composite Application Editor содержит компоненты со встроенными свойствами и действиями. Если использовать почтовый клиент mail8.ntf, то в нем по умолчанию доступен компонент PIM (управление личной информацией). Для отображения компонентов PIM, которые можно добавить в приложение, в редакторе приложений разворачивается панель компонентов (Component Palette). На этой панели выделяется Lotus Notes Calendar Navigator.

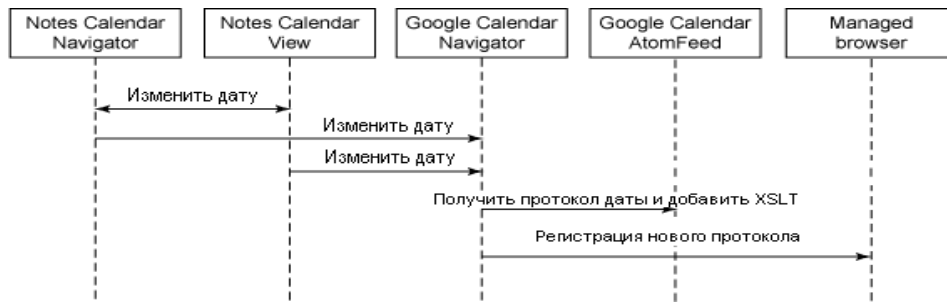


Рисунок 1 – Схема проекта

Компоненты в приложении связаны таким образом, что инициализация пользователем одного компонента приводит к программному действию другого компонента. В этом приложении, когда выделенная дата в Lotus Notes Calendar Navigator или Lotus Notes Calendar View изменяется, это изменение отображается и в другом компоненте.

Для создания связи между Lotus Notes Calendar Navigator и Lotus Notes Calendar View правой кнопкой щелкаем на навигационной панели Lotus Notes Calendar Navigator и выбираем Связывание (Wiring). Выбираем измененные свойства выделенных данных в Lotus Notes Calendar Navigator и перетаскиваем их в окно Set (Задать) выбранных данных в Lotus Notes Calendar View. Далее выбираем Lotus Notes Calendar View как связанный источник. И далее выбираем свойство Selected data change (изменить выделенные данные) в Lotus Notes Calendar Navigator и перетаскиваем их в действие Set selected date (Задать выбранную дату) в Lotus Notes Calendar View.

М.В. Лутков (УО «ГГУ им. Ф. Скорины», Гомель)
 Науч. рук. **В.Н. Леванцов**, ст. преподаватель

ГРУППОВАЯ РАЗРАБОТКА ПРИКЛАДНЫХ РЕШЕНИЙ В СИСТЕМЕ 1С: ПРЕДПРИЯТИЕ 8.2

Рост функциональности создаваемого прикладного решения приводит к тому, что требуется одновременная работа нескольких разработчиков для того, чтобы прикладное решение было создано в разумные сроки. При этом каждый из разработчиков, как правило, создает отдельную функционально законченную часть прикладного решения, которая взаимодействует с другими частями прикладного решения, созданными другими разработчиками.

Для того чтобы упростить и автоматизировать процесс разработки прикладного решения коллективом разработчиков, платформа «1С:Предприятие» содержит специальный механизм, позволяющий ве-

сти групповую разработку решений.

Основу механизма групповой разработки составляет хранилище конфигурации. Хранилище создается на каком-либо общем сетевом ресурсе, и в него помещается конфигурация, которую предполагается разрабатывать. Каждый из разработчиков создает на своей рабочей станции тестовую информационную базу и подключает ее к хранилищу. При подключении к хранилищу конфигурация тестовой базы автоматически заменяется конфигурацией, содержащейся в хранилище, и все объекты метаданных становятся недоступными для редактирования. Таким образом, каждый из разработчиков имеет возможность тестировать и отлаживать текущую разрабатываемую конфигурацию на своей рабочей станции и на своих тестовых данных, независимо от других разработчиков.

Ключевым моментом здесь является то, что, не предприняв специальных действий, разработчик не имеет возможности редактировать объекты конфигурации.

Дальнейшая работа строится следующим образом: если разработчику необходимо изменить какой-либо объект конфигурации, он захватывает его в хранилище, вносит изменения и затем помещает обратно в хранилище.

Таким образом, обеспечивается блокировка одновременного изменения объекта несколькими разработчиками: разработчик может захватить в хранилище только те объекты, которые не захвачены другими пользователями. Другими словами, захват объектов в хранилище выполняется эксклюзивно, и, захватив объект, разработчик может быть уверен, что этот объект не будет изменен другими разработчиками до тех пор, пока он не поместит его в хранилище или не отменит захват в хранилище.

Каждому пользователю хранилища может быть назначен индивидуальный набор прав, определяющий действия, которые данный пользователь может выполнять с хранилищем конфигурации. Существуют три права: Административные функции, Изменение состава версий и Захват объектов:

- право Административные функции позволяет создавать, изменять и удалять пользователей хранилища, отключать пользователей, подключенных к хранилищу, и отменять захват объектов хранилища, выполненный другими пользователями;

- право Изменение состава версий позволяет выполнять откат, сокращение и объединение версий конфигураций в хранилище;

- право Захват объектов позволяет захватывать объекты в хранилище и вносить изменения в конфигурацию хранилища.

Все эти права связаны друг с другом. Установка права Административные функции подразумевает, что пользователь может выполнять любые действия с хранилищем, поэтому при установке этого

права устанавливаются также и права Изменение состава версий и Захват объектов.

Установка права Изменение состава версий требует, чтобы пользователь имел возможность захвата и изменения объектов, поэтому вместе с этим правом устанавливается и право Захват объектов.

Помимо обеспечения захвата и блокировок объектов конфигурации, хранилище предоставляет целый ряд сервисных функций, которые упрощают ведение разработки и делают ее более автоматизированной.

Прежде всего, это возможность версионирования разработки и ведения истории изменения версий. Хранилище позволяет не только просматривать список версий, но и выводит отчеты в различных разрезах: по версиям, по объектам разработки, по комментариям к версиям.

Поддерживается возможность отката назад и удаления ненужных версий, опубликованных в хранилище, а также возможность удаления самых ранних ненужных версий путем сокращения до нужной версии.

Также хранилище позволяет разработчику просмотреть любую из версий, находящихся в хранилище, сравнить различные версии, загрузить выбранную версию в локальную отладочную базу, с которой работает разработчик, или сохранить в файл на диске.

В.Е. Малиновский (УО «ГГТУ им. П.О. Сухого», Гомель)

Науч. рук. **И.А. Мурашко**, д-р. тех. наук, профессор

ИСПОЛЬЗОВАНИЕ МЕТОДА АНАЛИТИЧЕСКИХ СЕТЕЙ В СИСТЕМАХ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

МАС (Метод аналитических сетей) – более общая форма метода анализа иерархий (МАИ), который используется в условиях мультикритериальности. МАИ структурирует решение проблемы в иерархию для определения критерия выбора и альтернативы, в свою очередь метод аналитических сетей (МАС) структурирует его в качестве аналитической сети, и затем используют систему парных сравнений для того, чтобы измерить вес компонентов структуры, и, наконец, ранжирует альтернативы в решении.

Существует ряд проблем, решение которых не может быть принято с помощью МАИ. Например, когда происходит взаимодействие элементов на высоком уровне с элементами более низкого уровня и их зависимости должны быть учтены. МАС предоставляет решение для проблем, которые не могут быть представлены иерархически. Опреде-

ление значения критерия играет роль не только для альтернатив; как и в иерархии, важность самих альтернатив определяет важность критериев.

Метод анализа иерархий является способом принятия решений, включающим в себя качественные факторы. В этом методе шкалы коэффициентов получаются из порядковых шкал, которые получают из отдельных суждений по качественным факторам использования матрицы парных сравнений. Метод аналитический сетей также использует матрицу попарных сравнений для того, чтобы получить соотношения масштабов. Разница между этими двумя способами проявляется в моделировании задачи и вычисления конечных приоритетов для альтернатив. МАИ моделирует принятие решения проблемы с помощью однонаправленного иерархического отношения между элементами решения. Однако МАС позволяет проводить более сложные взаимосвязи между элементами решения.

Оба метода используют одну основную формулу – определение приоритетов на основе вынесения заключений по парам элементов, или получение приоритетов благодаря нормализации прямых измерений. В МАИ элементы расположены в виде иерархической структуры с целью описание критериев выбора альтернатив, в МАС же элементы собраны в группы, одна из которых содержит альтернативы, которые содержат другие критерии или другие элементы решения.

В.Е. Малиновский (УО «ГГТУ им. П.О. Сухого», Гомель)
Науч. рук. **И.А. Мурашко**, д-р. тех. наук, профессор

МЕТОДЫ И ТЕХНОЛОГИИ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ОБЛАСТИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Для поддержки принятия решений с помощью информационных технологий, включая анализ и выработку альтернатив, в системах поддержки принятия решений (СППР) используются большое количество методов, каждый из которых имеет свои достоинства и недостатки.

Для хранения исходных данных СППР используется реляционная база данных, а в качестве системы управления базой данных – MicrosoftSQLServer.

В созданной системе поддержки принятия решений применяется метод аналитических сетей. Сеть состоит из 3 компонентов (в терминологии программы – кластеров), из которых «Цель» является компонентом-источником, а «Альтернативы» – компонентом-стоком. Центральную

роль играет кластер «Признаки», содержащий данные, на основе которых будет строиться решение задачи. Кластер «Альтернативы» содержит варианты, из которых лицу принимающему выбор (ЛПР) предстоит сделать выбор, основываясь на представленных программой рекомендациях. Каждый из признаков связан с каждой из альтернатив определенным коэффициентом. Соответственно, каждая из альтернатив связана с признаками с помощью коэффициентов значимости.

После создания структуры, для каждой выделенной взаимосвязи необходимо ввести попарные сравнения, определяемые ЛПР. Программа позволяет осуществлять ввод прямым способом. Также программа дает возможность просматривать вектор относительной значимости различных элементов задачи.

Исходные данные для программы будут представлены в виде количественных оценок, введенных пользователями. Реализована возможность просмотра влияния каждого из признаков на представленные альтернативы. Все значения приоритетов доступны в отдельном окне программы. После ввода и проверки значений всех приоритетов формируется суперматрица из полученных собственных векторов относительной значимости. Интерфейс программы позволяет построить суперматрицы: взвешенную матрицу (содержит приоритеты из матриц попарных сравнений умноженные на веса блоков суперматрицы) и предельную суперматрицу (взвешенная суперматрица приведенная к стохастическому виду и возведенная в предельную степень). В качестве результатов решения задачи пользователи получают графическое представление значений приоритетов исследуемых альтернатив.

А.П. Мартынов (УО «ГГУ им. Ф. Скорины», Гомель)

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

УПРАВЛЕНИЕ ФУНКЦИЕЙ ОСВЕЩЕНИЯ ПОМЕЩЕНИЙ С ПОМОЩЬЮ МОБИЛЬНЫХ УСТРОЙСТВ НА ПЛАТФОРМЕ ANDROID

В настоящее время прогресс не стоит на месте и население, очень часто, сейчас предпочитает пользоваться электронной системой управления освещения в квартирах. Ведь, прежде всего управление освещением обеспечивает новый уровень комфорта. Вам больше не будет надобности пользоваться выключателями света, всё это можно будет сделать, используя специальные команды.

В настоящее время существуют множество вариантов управления устройствами.

Пульты дистанционного управления на основе инфракрасного излучения. Основной недостаток малая дистанция управления и передатчик и приёмник должны находиться в зоне прямой видимости друг друга.

Управление посредством передачи команд и сигналов по стандарту Bluetooth (не важно какого стандарта). Намного лучше, чем ПДУ, но имеет очень важные недостатки, а именно также малый радиус приема и большое потребление энергии как приемником, так и передатчиком, что критически важно для мобильных устройств, работающих от аккумуляторной батареи.

Принцип управления по WI-FI заключается в том, что есть постоянная точка доступа, соединенная с центральным микропроцессорным устройством (сервером), которое взаимодействует с другими устройствами подключенными к нему либо по беспроводной связи, либо по иным протоколам, и клиентские точки, будь то телефоны, пульты (клиенты). При изменении состояния клиента, он передает информацию об изменении на сервер, тот в свою очередь реагирует на поступившую информацию, по заранее заложенному в него алгоритму. Основным недостатком передачи по Wi-Fi, является стоимость модулей. Из преимуществ можно и важно выделить большой радиус действия передатчика, малое энергопотребление, возможность совместной работы с другими устройствами, имеющими Wi-Fi.

Разрабатываемое устройство позволяет в полной мере управлять функцией освещенностью, как квартиры (дома), так и других объектов, где заботятся о рациональном использовании электроэнергии.

Устройство может включать освещение (все или по элементно) как в определенное время, так и по конкретным действиям (отработка датчиков), а управление яркостью, позволит настроить освещение под конкретный момент времени или под дизайн интерьера.

Использование последних беспроводных технологий, позволяет с легкостью применять различное управление: будь-то компьютер, будь-то мобильное устройство с необходимым ПО. Также это бесспорный факт легкой интеграции системы в существующие системы «умный дом», которые при не большой подстройке могут работать совместно и автономно.

Поддержка стандартных протоколов беспроводной связи открывает много возможностей, по интерактивному наблюдению и управлению за состоянием как ламп (горит или не горит), так и более крупных объектов, в системе умный дом – окон, дверей, электроприборов.

На рисунке 1 изображен пример устройства в виде блок схемы, где включение освещения управляется путем срабатывания датчика (датчик движения, датчик открытия двери, датчик температуры и т.д.).

Все устройство собрано в единый корпус, к которому подводится только переменное напряжение 220 В.

Состояние об объектах передается и отображается на дисплее мобильного телефона или планшета, работающего под управлением ОС Android. Приложение можно разработать абсолютно любое, начиная от тематики «Hello, Kitty» и до индивидуальных пожеланий. Удобность и простоту управления обеспечивает каждый сам себе.

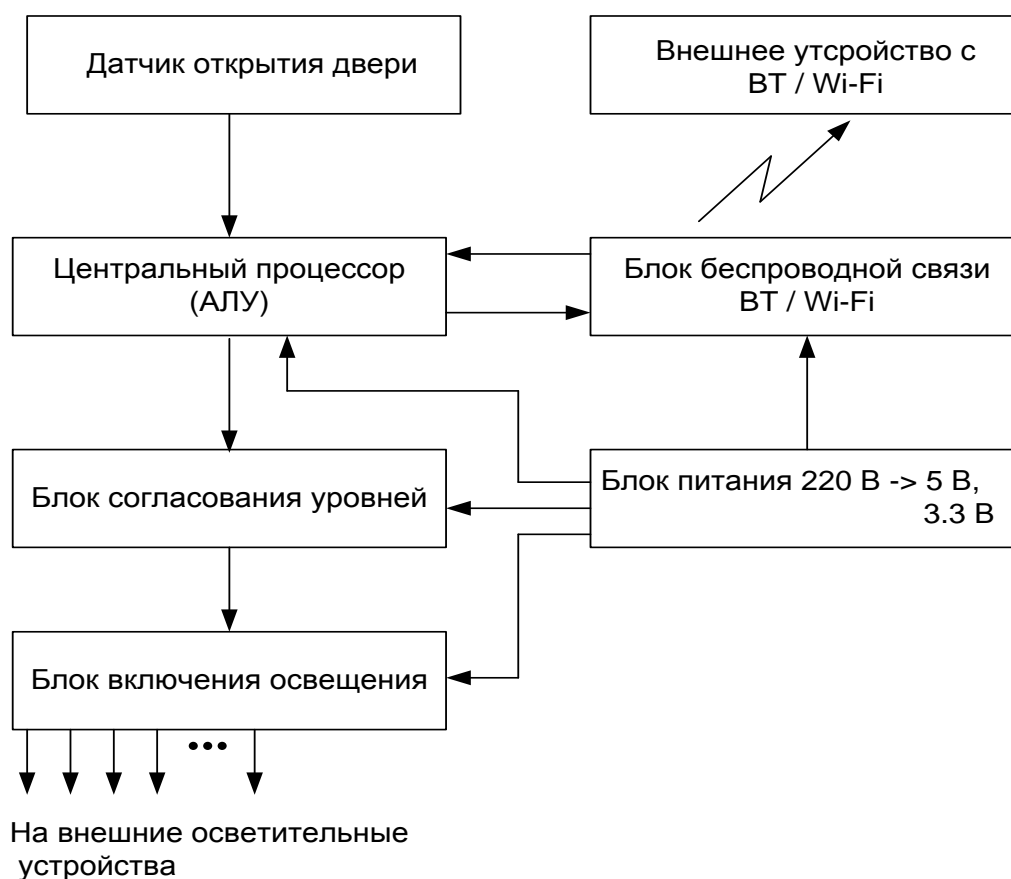


Рисунок 1 – Блок схема управления освещением.

Литература

1. Технология Wi-Fi – принципы работы, преимущества и недостатки [Электронный ресурс]. – 2014. – Режим доступа: http://www.mobility.ru/articlereview/wi-fi_technology – Дата доступа: 3.03.2014.

2. Фазоимпульсное управление силовым симистором. [Электронный ресурс].–2014.

Режим доступа: <http://www.embed.com.ua/mikrokontrollerniyi-konstruktor/fazoimpulsnoe-upravlenie-silovym-simistorom/> – Дата доступа: 3.03.2014.

Н.В. Масленченко (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА САЙТА «ЗАКАЗ БИЛЕТОВ»

Ведение бизнеса предполагает планирование рабочего дня до минут, так как время становится все более ценным элементом. С развитием информационных технологий появляются сервисы, которые предоставляют услуги, заменяющие рутинные операции, такие как предварительный заказ билетов. В настоящее время один из наиболее простых способов приобретения билетов – это бронирование билетов через Интернет. Авиаперелеты являются самым быстрым и удобным способом передвижения, особенно на большие расстояния. Но повышение цен на горючее и расходов на обеспечение безопасности заставляют стоимость билетов неуклонно расти. Сэкономить на цене авиабилета можно только заказав его через Интернет.

Электронный билет имеет значительные преимущества по сравнению с бумажным билетом. Электронный билет невозможно потерять, из-за того что вся информация сохранена надежно в компьютерах организации-перевозчика. Кроме того, электронный билет невозможно подделать, что служит залогом безопасности для покупателя. Покупка билета на самолет выполняется любым возможным для пассажира видом – за наличный расчет, банковским переводом, картой банка.

Заказ билетов по Интернету имеет и другие преимущества. Если вы покупаете билет онлайн, вы одновременно можете забронировать номер в отеле, благодаря чему заметно сэкономите. Многие гостиницы, при предварительном бронировании номеров, предоставляют клиентам скидки, но и здесь есть свои ограничения. В любом случае, заказав билеты и номера в гостинице через Интернет, вы сэкономите значительную сумму и, что, немаловажно, время.

В ходе проектирования разработки сайта были учтены следующие возможные варианты: перелеты с пересадками, заказ еды в самолете, регистрация багажа, выбор сидения в самолете, оплата карточками и популярными веб-сервисами, регистрация отелей и аренда машин.

Разработанный сайт удовлетворяет требованиям, поставленным на этапе постановки задачи и находится на этапе тестирования. В качестве дальнейшего совершенствования сайта предусмотрена возможность разработки и внедрения различных модулей.

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

Науч. рук. **А.И. Кучеров**, ст. преподаватель

АВТОМАТИЗАЦИЯ РАСЧЕТОВ СДЕЛЬНОЙ ЗАРАБОТНОЙ ПЛАТЫ ДЛЯ ПРЕДПРИЯТИЯ КСУП «ТЕПЛИЧНОЕ»

В растениеводстве для учета отработанного времени, объема выполненных работ и начисления заработной платы используется книжка бригадира. В первой части книжки указывается объем выполненных работ и начисленная заработная плата по культурам и видам работ. Во второй части отражается начисленная заработная плата каждому работнику за каждый отработанный день и за месяц.

Книжка бригадира по учету труда и выполненных работ (форма 505-АПК) – применяется для учета отработанного времени, объемов выполненных работ и начисления заработной платы работникам, занятым на конно-ручных работах. Книжка бригадира ф. 505-АПК состоит из «Листа учета выполненных работ», где учитывают выполненные работы и начисленную за них заработную плату по культурам и видам работ, и «Табеля учета рабочего времени и подсчета заработка на конно-ручных работах», в котором отражают отработанное время и объем работ, выполненный каждым исполнителем.

Табель учета рабочего времени и подсчета заработка на конно-ручных работах заполняет бригадир. Указывается фамилия, имя, отчество работника, табельный номер, категория, фактически отработанное время за каждый день, объем выполненных работ, заработная плата и доплаты. Учет рабочего времени и подсчет сдельной заработной платы рабочих на предприятии ведут бригадиры. Большую часть своего рабочего времени им придется тратить на выполнение данной, достаточно трудоемкой функции. Много времени занимает и у бухгалтеров ручной перенос сведений из формы 505-АПК в систему 1С Бухгалтерия. Автоматизация учета книжки бригадира 505-АПК позволит повысить производительность труда бригадиров цехов, отдела труда и заработной платы и расчетной группы при вводе и обработке данных о фактически отработанном времени, а также уменьшить вероятность ошибок при регистрации данных о фактически выполненной работе и заработной плате.

Наличие табелей учета рабочего времени также важно для учета трудовых затрат и в целях бухгалтерского, и в целях налогового учета. Затраты на оплату труда, подлежащие отнесению на отдельные объекты учета затрат, определяются на основании оформленных в установленном порядке первичных документов.

Решать данную задачу будем с помощью программного комплекса на базе платформы 1С: «Предприятие» версии 8.2. Так как автоматизируем

первичный учет по зарплате, который ведут бригадиры в цехах, для распределения в дальнейшем прав пользователей необходимо создать подсистему Расчет Сдельной Оплаты, в которую включим наши первичные документы по зарплате. К этой подсистеме будут иметь доступ бригадиры, при этом они не будут иметь доступ к другим подсистемам нашей конфигурации.

Создадим справочник Виды работ в который бригадир будет вносить виды работ с указанием шифра, нормы и разряда. Для того чтобы создать справочник, перейдем в ветвь Справочники дерева конфигурации и создадим в ней новый элемент. Для начала на закладке Данные нужно настроить длину кода и наименования элемента справочника. Девятизначный код способен вместить достаточное количество элементов. Код используется для идентификации элементов справочника и код каждого элемента уникален. Система сама следит за уникальностью кодов.

Теперь создадим документ Наряд на сдельную работу. Документы – это прикладные объекты, которые используются для ввода информации в систему. Нередко документы конфигураций 1С:Предприятие являются электронными аналогами бумажных документов. Как правило, прикладные конфигурации строят с учетом того, что информация о совершенной хозяйственной операции вводится в систему лишь один раз с использованием соответствующего документа, после чего используется в различных системных механизмах. Создавая конкретные документы, обычно ориентируются на хозяйственные операции, которые нужно отражать в системе с их помощью.

С помощью этого документа собираемся отражать факт выполненной работы овощеводами. В группе параметров, расположенную в нижней части вкладки Данные, можно добавлять в документ табличные части и настраивать их свойства. Табличную часть можно сравнить с обычной таблицей. Эту таблицу заполняют при создании документа. На этапе конфигурирования задаются реквизиты табличной части, а строки создаются на этапе работы с программой конечного пользователя. Можно добавлять в документ столько табличных частей, сколько нам нужно. В нашем случае добавим в документ одну табличную часть, которую назовем Выполненные Работы. Для добавления табличной части воспользуемся кнопкой Добавить табличную часть. После того, как табличная часть добавлена, можно добавлять ее реквизиты. При добавлении реквизитов табличной части их параметры настраиваются аналогично параметрам обычных реквизитов документа. Табличную часть назовем выполненные работы и создадим

реквизиты: *ФамилияИмяОтчество, ОбъемВыполненнойРаботы, Расценка, Сумма.*

Для того, чтобы система автоматически рассчитывала поле Сумма для каждого рабочего, нам понадобится написать обработчик событий изменений полей таблицы. А именно – при изменении поля Объем выполненной работы в поле Сумма должен подставляться результат умножения расценки на объем выполненной работы.

Создадим эту процедуру:

*Процедура ВыполненнаяРаботаКоличествоПриИзменении(Элемент)
СтрокаТабличнойЧасти=*

Элементы.ВыполненнаяРабота.ТекущиеДанные;

СтрокаТабличнойЧасти.Сумма=

*СтрокаТабличнойЧасти.ОбъемРаботы**

СтрокаТабличнойЧасти.Расценка;

КонецПроцедуры

Свойство табличного поля *ТекущиеДанные* содержит данные текущей строки. Для доступа к этим данным используем переменную *ТекущаяСтрока*. Создадим новый регистр накопления Сдельная Зарплата, в нем будут накапливаться данные, которые нам в дальнейшем будут необходимы для расчета сдельной зарплаты за месяц. Регистр накопления можно представить в виде многомерной системы координат, осями которой являются измерения, а ресурсы – это данные, которые хранятся в узлах этой системы координат.

Данные для хранения в регистры попадают из документов. В регистре накопления хранятся записи с нужным набором измерений, ресурсов и реквизитов. Каждой записи соответствует регистратор – документ, который сделал эту запись. Записи в регистры выполняются при проведении документа. Этот процесс еще называют формированием движений по регистру.

Регистр накопления Наряд на сдельную работу будет использоваться регистром расчета для начисления заработной платы. Регистры расчета позволяют организовать учет результатов вычислений: осуществляемых с некоторой периодичностью, тесно связанных друг с другом по некоторым правилам, взаимно влияющих друг на друга в пределах определенного периода. Такие особенности регистров расчета позволяют, например, реализовать регистрацию начислений в пользу физических лиц (оплата труда, оплата больничных листов, оплата дней отпуска и т.д.). Регистр расчета *Сдельной Заработной платы* будет подсчитывать обороты по регистру накопления *Наряд на сдельную работу* за интервал периода действия записи.

Р.В. Матюшин (УО «ГГУ им. Ф. Скорины», Гомель)

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

РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ СОДЕРЖИМЫМ ВЕБ-САЙТА ДЛЯ ЖЕЛЕЗНОДОРОЖНОГО (г. ГОМЕЛЯ) ОТДЕЛА ДЕПАРТАМЕНТА ОХРАНЫ

Интернет, как канал передачи информации в современном мире, выходит на лидирующие позиции по охвату аудитории, опережая традиционные средства массовой информации, такие как печатная продукция, телевидение и радиовещание.

Много информации выкладывается в сети, и ее поиск через интернет для людей становится все актуальнее. Это объясняется легкостью, удобством и доступностью. Использование Интернет-ресурса позволяет сократить количество времени, затрачиваемого на поиск необходимых сведений.

В связи с данной тенденцией все большее количество различных организаций обращаются к специалистам-разработчикам сайтов с целью размещения информации об их деятельности.

С помощью сайта организация получает возможность донести необходимую информацию до широкого круга пользователей сети интернет и взаимодействовать с ними в интерактивном режиме. Таким образом, как частным, так и государственным структурам необходимо создание онлайн-представительства организации в сети, которое позволит посетителям сайта получать актуальную информацию о ее деятельности и предоставляемых услугах. Еще одним преимуществом является предоставление онлайн сервисов для пользователей, с помощью которых организуется интерактивный диалог с администрацией организации.

В результате проделанной работы был разработан программный комплекс для организации интернет-сайта Железнодорожного отдела Департамента охраны г. Гомеля. Программный комплекс позволяет организовать и структурировать информацию об организации в рамках отдельного интернет-ресурса. На сайте можно ознакомиться со свежими новостями отдела, получить исчерпывающую информацию о деятельности организации, порядке оказываемых услуг, истории создания отдела, списке вакансий, а также контактной информации.

Разработанная система построена согласно концепции MVC (Model-View-Controller), которая позволяет разделить данные, представление и обработку на три отдельных компонента.

Модель (англ. Model). Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.

Представление, вид (англ. View). Отвечает за отображение информации (визуализацию).

Контроллер (англ. Controller). Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

Важно отметить, что как представление, так и контроллер зависят от модели. Однако модель не зависит ни от представления, ни от контроллера. Тем самым достигается назначение такого разделения: оно позволяет строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели.

В разработанной системе предусмотрены онлайн-сервисы для посетителей сайта. Пользователю, посетившему интернет-сайт, предоставляется возможность:

- получить актуальную информацию об услугах, предоставляемых отделом;
- ознакомиться с историей отдела;
- ознакомиться с руководством отдела;
- ознакомиться с актуальными вакансиями для устройства на работу;
- получить возможность рассчитать приблизительную стоимость постановки своей квартиры под охрану;
- заполнить заявление для постановки личной квартиры под охрану;
- задать интересующие вопросы.

Модуль обратной связи служит для организации связи между посетителем сайта и администрацией. Пользователь заходит на специализированную страницу и заполняет поля формы. После чего, полученная информация проходит проверку на корректность и отправляется администратору сайта на e-mail.

Пользователь может задать интересующий его вопрос и в кратчайшие сроки получить квалифицированный ответ. Пользователь может ознакомиться с другими вопросами, которые были заданы ранее посетителями ресурса.

Реализована возможность получить информацию о приблизительной стоимости средств и систем охраны, а также инженерно-технических средств защиты личной квартиры. Для этого можно воспользоваться калькулятором стоимости, который на основе введенных данных, рассчитывает стоимость постановки объекта под охрану.

Для облегчения службы приема заявок на постановку охраны объектов с личным имуществом граждан на сайте существует возможность заполнения бланка заявления, которое будет передано непосредственно данной службе для обработки.

Для реализации программного комплекса требуется использование Web-сервера Apache с PHP-интерпретатором и СУБД MySQL. Взаимодействие пользователя с системой происходит посредством веб-браузера.

Все данные сайта хранятся в структурированном виде под управлением реляционной СУБД. Исключения составляют файлы данных, предназначенные для просмотра и скачивания (изображения, видео, документы и т.п.). Такие файлы сохраняются в файловой системе, а в БД размещаются ссылки на них.

Пользовательский интерфейс сайта обеспечивает наглядное, интуитивно понятное представление структуры размещенной на нем информации, быстрый и логичный переход к разделам и страницам. Навигационные элементы обеспечивают однозначное понимание пользователем их смысла: ссылки на страницы снабжены заголовками, условные обозначения соответствуют общепринятым. Графические элементы навигации снабжены альтернативной подписью.

Разработанный программный комплекс позволяет быстро решать задачи построения информационного ресурса с интерактивными веб-сервисами для предоставления информации в сети интернет. Это особенно актуально в современном мире, учитывая развитие информационных технологий. Данный программный комплекс может быть применен для реализации других подобных проектов.

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

Науч. рук. **Г.Л. Карасёва**, канд. физ.-мат. наук, доцент

ОПТИМИЗАЦИЯ ДИНАМИЧЕСКИХ СИСТЕМ С ФАЗОВЫМИ ОГРАНИЧЕНИЯМИ

Теория оптимального управления возникла в начале 50-х годов в связи с появлением новых задач, поставленных практикой. Необходимость решения актуальных проблем теории автоматического регулирования, динамики полета, потребовала разработки эффективных методов решения экстремальных задач нового типа. Такие задачи стали называться неклассическими вариационными задачами или задачами оптимального управления.

Естественно, что среди непрерывных систем в первую очередь были изучены линейные системы, описываемые обыкновенными дифференциальными уравнениями:

$$\dot{x} = Ax + bu, \quad (0)$$

Для системы (0) качественная теория оптимального управления во многих своих разделах к настоящему времени достигла очень высокого уровня. Среди результатов этой теории выделяются принцип максимума Л.С. Понтрягина, динамическое программирование Р. Беллмана, метод моментов Н.Н. Красовского.

Уравнения (0) являются математическими моделями многих процессов в различных сферах человеческой деятельности. В них переменные $x(t)$, $t \in T$, представляют значения полного набора внутренних характеристик изучаемого процесса в момент времени t . Переменные $u(t)$, $t \in T$, называются переменными управления, A — $n \times n$ -матрица, характеризующая динамические свойства объекта, b — n -вектор параметров входного устройства.

Согласно теории дифференциальных уравнений, поведение объекта $x(t)$, $t \in T$, будет однозначным, если задать его начальное состояние

$$x(0) = x_0$$

и управляющее воздействие $u(t)$, $t \in T$, из класса кусочно-постоянных функций.

В целом система (0) представляет собой дифференциальный закон поведения процесса управления. Получение и использование законов поведения в дифференциальной форме широко распространено в современных научных исследованиях, так как она компактно и адекватно выражает фундаментальные свойства многих явлений.

Большие усилия были приложены к созданию эффективных методов решения задач оптимального управления. Однако до сих пор даже для линейных задач не предложено алгоритмов надёжно строящих оптимальное управление.

Линейная задача оптимального управления с фазовыми ограничениями формулируется в виде:

$$\begin{aligned} c'x(t_1) &\rightarrow \max, \\ \dot{x} &= Ax + bu, \quad x(0) = x_0, \quad Hx(t_1) = g, \\ \alpha_*(t) &\leq Dx(t) \leq \alpha^*(t), \quad f_* \leq u(t) \leq f^*, \quad t \in T = [0, t_1], \end{aligned}$$

где $x \in R^n$; $u \in R^r$; c, A, B, H, D — постоянные векторы и матрицы соответствующих размеров.

В теории оптимального управления наряду с качественной теорией, которая анализирует вопросы существования решения, необходимые и достаточные условия оптимальности, корректность постановки задачи, структуру решения и т. п., большое внимание уделяется

конструктивным вопросам, связанным с фактическим (аналитическим или численным) построением (синтезом) решения задач оптимального управления. Много исследований проведено в этой области, а именно А.Д. Иоффе, В.М. Тихомировым, Н.Н. Красовским и другими.

Появление принципа максимума Л.С.Понтрягина дало мощный толчок развитию новых численных методов решения задач оптимального управления, среди которых можно выделить несколько направлений. Прежде всего, это методы, основанные на сведении задачи оптимального управления к краевой задаче (при помощи принципа максимума). К отдельной группе методов можно отнести градиентные методы. На протяжении многих лет делается попытка реализовать численные методы на основе динамического программирования.

Следует отметить методы математического программирования в теории оптимального управления. Это направление, в котором задачу оптимального управления заменяют конечномерной задачей, и решают её хорошо разработанными методами математического программирования.

Несмотря на большие успехи, достигнутые в теории оптимального управления, многие ее проблемы остаются нерешенными до настоящего времени. Это, прежде всего, относится к задачам с фазовыми ограничениями. Поэтому создание эффективных методов решения задач с фазовыми ограничениями остается одним из приоритетных направлений современной теории оптимального управления.

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

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

АНСАМБЛЕВЫЙ ГИДРОДИНАМИЧЕСКИЙ ПРОГНОЗ ПОГОДЫ

Прогнозы погоды основаны на интерпретации результатов численного моделирования.

Подготовка прогнозов с помощью компьютеров начинается с описания состояния атмосферы, основанного на прошлых и текущих наблюдениях, в виде процесса, называемого усвоением данных, в котором используется модель численного прогнозирования погоды (ЧПП) для обобщения и экстраполяции по времени информации, извлеченной из прошлых наблюдений. Усвоение данных является весьма эффективным при неполном охвате данными наблюдений из различных источников, используемыми с целью создания логически согласованной оценки состояния атмосферы. Однако, подобно прогнозу, усвоение данных базируется на модели

ЧПП и не может непосредственно использовать наблюдения таких масштабов и процессов, которые не представлены в модели.

Однако атмосферные процессы изначально являются нелинейными, и не все физические процессы можно понять или представить в моделях ЧПП.

Имеется широкий спектр видов атмосферного движения – от планетарного масштаба до местной турбулентности. Это свойство атмосферы означает, что небольшие неопределенности состояния атмосферы также будут возрастать, и таким образом, в конечном итоге, окажется невозможным точно предсказать неустойчивые системы.

Традиционно прогнозирование текущей погоды концентрируется на анализе и экстраполяции наблюдаемых метеорологических полей с особым упором на мезомасштабные поля облаков и осадков, полученные по данным спутников и радиолокаторов. Однако интенсивность изменения явлений во времени является таковой, что простая экстраполяция важных характеристик приводит к тому, что продукция очень быстро ухудшается со временем, даже во временных масштабах порядка одного часа. Поэтому разрабатываются методы, в которых сочетаются методы экстраполяции с ЧПП, при этом как за счет смешения двух видов продукции, так и с помощью улучшенной ассимиляции подробных мезомасштабных наблюдений.

Численное прогнозирование погоды – это прогнозы с заблаговременностью, превышающей несколько часов, почти всегда полностью основываются на ЧПП. В действительности, большую часть улучшений в оправдываемости прогнозов погоды за последние 20 лет можно отнести за счет компьютерных моделей ЧПП, которые строятся с использованием уравнений, описывающих динамическое и физическое изменение атмосферы.

Модели ЧПП представляют атмосферу на трехмерной сетке. Точно можно предсказывать только погодные системы, которые в несколько раз превышают шаг сетки, и поэтому явления в меньших масштабах должны представляться в приближенном виде с использованием статистических и других методов. Эти ограничения в моделях ЧПП оказывают особое влияние на подробные прогнозы местных элементов погоды, таких, как облачность и туман, а также экстремальных явлений, таких, как интенсивные осадки и максимальные порывы ветра. Они также вносят вклад в неопределенности, которые могут хаотично возрастать и, в конечном итоге, ограничивать предсказуемость.

Обработка данных происходит в несколько этапов. Частный контроль полученных данных. После этого программа комплексного контроля проверяет невязки во всех частных контролях, а затем принимает решение об ошибочности или верности данного измерения. Затем

системы усвоения метеорологической информации используют разнообразные данные от самых разнообразных наблюдательных приборов и должны быстро их проверить и согласовать между собой. И в заключение проконтролированные данные интерполируются в правильную сетку точек на поверхности Земли или ее части. При этой интерполяции следует учесть статистические характеристики как реальных метеорологических полей, так и полей предыдущего прогноза на момент данного объективного анализа. Данные в геометрически правильной сетке используются в качестве начальных для системы уравнений в частных производных, описывающих динамику атмосферы. Это уравнения газовой динамики, в которые добавлены различные физические эффекты, не наблюдаемые в идеальном газе. Учитывается вращение Земли, ускорение силы тяжести, солнечная и отраженная от поверхности Земли радиация, фазовые переходы воды и многое другое.

Однако, даже при учете положительных качества ЧПП, неопределенность существует всегда, даже в наших сведениях о текущем состоянии атмосферы. Она хаотично возрастает во времени с поступлением большого количества новой информации, которая вначале не добавляет дополнительное значение, до тех пор, пока не остается только климатологическая информация.

Скорость роста этой неопределенности трудно оценить, поскольку она зависит от трехмерной структуры атмосферного потока. Решение этой проблемы состоит в использовании группы прогнозов – ансамбля – с рядом незначительно отличающихся начальных условий и группы моделей ЧПП с различными, но равновозможными приближениями. Если ансамбль хорошо построен, то его прогнозы охватят весь диапазон возможных результатов, включая ряд образований, где могут возрасти неопределенности. По этому комплекту прогнозов можно автоматически получить информацию о вероятностях, соответствующую требованиям пользователей.

Основная проблема ансамблевого прогнозирования исходит из ограничений ЧПП. Кроме того, поскольку группа прогнозов рассчитывается одновременно, то остается меньше компьютерных мощностей для каждого прогноза. В этой связи требуется увеличение шага сетки, что ведет к затруднениям в представлении некоторых суровых явлений погоды меньшего горизонтального масштаба. С учетом ограниченного количества прогнозов в ансамбле, трудно вычислить вероятности весьма экстремальных и редких явлений непосредственно по ансамблю. Более того, невозможно изменить модели ЧПП, используемые для должной выборки ошибок моделирования, и, таким образом, иногда во всех моделях будут допускаться одни и те же ошибки.

Таким образом, применение численного и ансамблевого методов прогнозирования дают возможность получить практически достоверную информацию об ожидаемой погоде.

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

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

ЗНАЧЕНИЕ КОМПЬЮТЕРНЫХ СЕТЕЙ В ПРОИЗВОДСТВЕ

Современное производство требует высоких скоростей обработки информации, удобных форм ее хранения и передачи. Необходимо также иметь динамичные способы обращения к информации, способы поиска данных в заданные временные интервалы; реализовывать сложную математическую и логическую обработку данных. Управление крупными предприятиями, управление экономикой на уровне страны требуют участия в этом процессе достаточно крупных коллективов. Такие коллективы могут располагаться в различных районах города, в различных регионах страны и даже в различных странах. Для решения задач управления, обеспечивающих реализацию экономической стратегии, становятся важными и актуальными скорость и удобство обмена информацией, а также возможность тесного взаимодействия всех участвующих в процессе выработки управленческих решений. С появлением ЭВМ эта задача была реализована в виде компьютерной сети и программного обеспечения, обеспечивающие обмен информации.

Компьютерные сети отнюдь не являются единственным видом сетей, созданным человеческой цивилизацией. Даже водопроводы Древнего Рима можно рассматривать как один из наиболее древних примеров сетей, покрывающих большие территории и обслуживающих многочисленных клиентов. Другой, менее экзотический пример – электрические сети. В них легко можно найти все компоненты любой территориальной сети: источники ресурсов – электростанции, магистрали – высоковольтные линии электропередач, сеть доступа – трансформаторные подстанции, клиентское оборудование – осветительные и бытовые электроприборы.

Установка сетей и сетевых компьютерных систем – это только два примера ориентированных на завтрашний день методов работы и концепций в области допечатных процессов. Функционирование систем и сетей и их применение обеспечивают сегодня эффективность как производственных, так и экономических показателей. Поэтому сетевым технологиям отводится все больше внимания, в том числе и в полиграфической

промышленности. Прежде всего представляется все более насущной задача объединения в сеть всех участков допечатной стадии (а для создания «сетевой полиграфии» объединение в сеть всех подразделений типографии).

Компьютерные сети, называемые также вычислительными сетями, или сетями передачи данных, являются логическим результатом эволюции двух важнейших научно-технических отраслей современной цивилизации – компьютерных и телекоммуникационных технологий. С одной стороны, сети представляют собой частный случай распределенных вычислительных систем, в которых группа компьютеров согласованно выполняет набор взаимосвязанных задач, обмениваясь данными в автоматическом режиме. С другой стороны, компьютерные сети могут рассматриваться как средство передачи информации на большие расстояния, для чего в них применяются методы кодирования и мультиплексирования данных, получившие развитие в различных телекоммуникационных системах.

Компьютерная (вычислительная) сеть – совокупность компьютеров и терминалов, соединённых с помощью каналов связи в единую систему, удовлетворяющую требованиям распределённой обработки данных. Основное назначение любой компьютерной сети – предоставление информационных и вычислительных ресурсов подключенным к ней пользователям. Компьютерные сети являются высшей формой многомашинных ассоциаций. Выделим основные отличия компьютерной сети от многомашинного вычислительного комплекса.

Первое отличие – размерность. В состав многомашинного вычислительного комплекса входят обычно две, максимум три ЭВМ, расположенные преимущественно в одном помещении. Вычислительная сеть может состоять из десятков и даже сотен ЭВМ, расположенных на расстоянии друг от друга от нескольких метров до десятков, сотен и даже тысяч километров.

Второе отличие – разделение функций между ЭВМ. Если в многомашинном вычислительном комплексе функции обработки данных, передачи данных и управления системой могут быть реализованы в одной ЭВМ, то в вычислительных сетях эти функции распределены между различными ЭВМ.

Третье отличие – необходимость решения в сети задачи маршрутизации сообщений. Сообщение от одной ЭВМ к другой в сети может быть передано по различным маршрутам в зависимости от состояния каналов связи, соединяющих ЭВМ друг с другом.

Объединение в один комплекс средств вычислительной техники, аппаратуры связи и каналов передачи данных предъявляет специфические

требования со стороны каждого элемента многомашинной ассоциации, а также требует формирования специальной терминологии.

Абоненты сети – объекты, генерирующие или потребляющие информацию в сети. Ими могут быть отдельные ЭВМ, комплексы ЭВМ, терминалы, промышленные роботы, станки с числовым программным управлением и т.д. Любой абонент сети подключается к станции. Станция – это аппаратура, которая выполняет функции, связанные с передачей и приёмом информации. Совокупность абонента и станции принято называть абонентской системой

Для организации взаимодействия абонентов необходима физическая передающая среда. На базе физической передающей среды строится коммуникационная сеть, которая обеспечивает передачу информации между абонентскими системами. Такой подход позволяет рассматривать любую компьютерную сеть как совокупность абонентских систем и коммуникационной сети.

В зависимости от территориального расположения абонентских систем вычислительные сети можно разделить на три основных класса:

- глобальные сети (WAN – Wide Area Network);
- региональные сети (MAN – Metropolitan Area Network);
- локальные сети (LAN – Local Area Network).

Глобальная вычислительная сеть объединяет абонентов, расположенных в различных странах, на различных континентах. Взаимодействие между абонентами такой сети может осуществляться на базе телефонных линий связи, радиосвязи и систем спутниковой связи. Глобальные вычислительные сети позволяют решить проблему объединения информационных ресурсов всего человечества и организации доступа к этим ресурсам.

Региональная вычислительная сеть связывает абонентов, расположенных на значительном расстоянии друг от друга. Она может включать абонентов внутри большого города, экономического региона, отдельной страны. Обычно расстояние между абонентами региональной вычислительной сети составляет десятки – сотни километров.

Локальная вычислительная сеть объединяет абонентов, расположенных в пределах небольшой территории. В настоящее время не существует четких ограничений на территориальный разброс абонентов локальной вычислительной сети. Обычно такая сеть привязана к конкретному месту. К классу локальных вычислительных сетей относятся сети отдельных предприятий, фирм, банков, офисов и т.д. Протяженность такой сети можно ограничить пределами 2–2,5 км.

Объединение глобальных, региональных и локальных вычислительных сетей позволяет создавать многосетевые иерархии. Они

обеспечивают мощные, экономически целесообразные средства обработки огромных информационных массивов и доступ к неограниченным информационным ресурсам. Локальные вычислительные сети могут входить как компоненты в состав региональной сети, региональные сети – объединяться в составе глобальной сети и, наконец, глобальные сети могут также образовывать сложные структуры.

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

Науч. рук. **Г.Л. Карасёва**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА ФОРУМА «STARTPOINT»

Всемирная сеть Интернет представляет собой объединение миллионов компьютеров по всему миру, и позволяет получить доступ к практически неограниченному объему информации. Поэтому, даже на заре интернета, когда сеть была не такой всеохватывающей структурой, найти нужную информацию или единомышленников в ней было довольно проблематично. Тогда, в числе прочих новшеств, появилась идея создания тематических сайтов – форумов, где посетители могли общаться на некоторые конкретные интересующие их темы.

Хотя, в наше время, форумы испытывают большую конкуренцию со стороны социальных сетей, они по-прежнему занимают свою нишу тематических ресурсов.

Классический интерфейс форума представляет собой страницу со списком ссылок на основные ветви тем, которые образуют собой древовидную иерархию. Как правило, на форумах пользователи могут сами создавать новые темы, если они не нашли интересующей их. Зачастую новички на форуме создают темы, игнорируя иерархию, или не учитывают тематику данной ветви, чем мешают остальным пользователям нормально использовать ресурс. Поэтому на форумах принята система модерации, т.е. из числа постоянных пользователей форумом назначаются самые добросовестные, которые следят за соблюдением тематичности и, в общем, поддерживают порядок.

Техническая реализация форумов, как правило, относительно проста. Как уже говорилось выше, основной интерфейс форума представляет собой дерево тем, ну и, собственно, страницу обсуждения выбранной темы. Сама страница обсуждения представляет собой заголовок темы и последовательно расположенные друг под другом записи пользователей. Обычно на форумах есть возможности цитирования, ответа или еще чего, на что хватит фантазии разработчика, но все равно логика и характер общения на форумах – едины. Обновление информации на

странице обсуждения темы обычно инициируется самим пользователем при создании своего ответа, либо при ручном обновлении страницы. Как видно из описания интерфейса форума, реализация его классического варианта – элементарна и предъявляет минимум требований как к технической части, так и к необходимому ПО. Поэтому язык и прочие средства разработки форума можно выбирать исходя из их простоты и доступности, не заботясь о поиске высокопроизводительных систем.

В наше время существует множество сред и средств автоматизации разработки сайтов и форумов в том числе. Но любые системы на выходе дают совокупность скриптовых файлов, которые обрабатывает сервер, в свою очередь возвращающий пользователю в ответ на запрос html-страницы с вкраплениями javascript. Основные языки скриптов на стороне сервера: ASP от компании Microsoft и PHP, разработанный Расмусом Лердорфом. ASP, в связи с политикой компании-разработчика, работает только на серверах IIS собственного производства, которые, в свою очередь, устанавливаются только на их же ОС семейства windows, что значительно сужает сферу его использования. PHP же, наоборот, разрабатывался для свободных unix подобных систем и распространяется под свободной лицензией. Простота, функциональность и свободная лицензия сделали PHP мультиплатформенным и, как следствие, самым распространенным языком построения сайтов.

Таким образом, для разработки форума необходимо и достаточно знать: язык скриптов на стороне сервера – PHP, язык запросов для СУБД MySQL, и скриптовый язык, работающий на стороне клиента, для расширения функционала HTML-страниц – JavaScript. Ну и, разумеется, не обойтись без самого языка разметки страниц – HTML, который будет предоставлять интерфейс, объединяющий все средства и единое интернет-приложение, работающее на стороне клиента в его браузере. Целью данной курсовой работы будет ознакомление с данными средствами разработки.

Отличительная особенность форумов, в сравнении с обычными веб-страницами, заключается в том, что наряду с отображением доступных статей они будут отображать взаимосвязь между статьями. Т.е., пользователь может видеть, какие статьи являются ответами на другие статьи (и за какой статьей они следуют), а какие – новыми темами обсуждения.

Способ отображения этих взаимосвязей должен быть тщательно продуман. Пользователь должен иметь возможность просматривать отдельное сообщение, цепочку беседы с показанными взаимосвязями или все цепочки в системе.

Разработка форума – довольно увлекательное занятие. А PHP и MySQL помогают максимально упростить задачу. Сам же проект фору-

ма «StartPOInT» планируется развивать и в дальнейшем разместить на бесплатном хостинге. Основной тематикой данного форума будет обмен своими идеями в области программирования, а также помощь студентам в выполнении учебных заданий по программированию.

М.В. Моцар (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **В.А. Дробышевский**, ст. преподаватель

СОЗДАНИЕ ЦВЕТОВОЙ СХЕМЫ WEB-САЙТА

Эмоциональное или психологическое воздействие цвета не так легко проанализировать, как физиологические процессы, возникающие в результате цветовосприятия, а между тем, большинство из нас отдает предпочтение определенным цветам и считает, что цвет влияет на настроение. Многие находят затруднительным жить и работать в помещениях, цветовое оформление которых кажется им неудачным. Цвета разделяют на сильные и слабые, успокаивающие и возбуждающие, даже на тяжелые и легкие.

Хотя отношение к цвету во многих случаях носит чисто субъективный характер, исследования показывают, что существуют и общие точки зрения. Но мы без труда забываем о своем отношении к цвету под влиянием других ощущений, и границы, которые мы проводим, приписывая тому или иному цвету определенное качество, условны и расплывчаты, поэтому ученые часто получают во время опытов противоречивые данные.

Во время первых нескольких секунд посещения сайта у пользователя может сложиться негативное впечатление. Хорошо, если цвета нового сайта гармонируют с его текстовым и графическим наполнением и отвечают вкусам аудитории, на которую сайт рассчитан. Призвание цветовой палитры сайта подчеркнуть его выразительность, но не отвлекать гостей сайта от цели, с которой они зашли – изучить содержание Web-ресурса. Для этого Вы и заказываете проект – донести как можно большему количеству людей определенную информацию. С помощью цветовой гаммы можно сделать акцент на основные разделы сайта, например, выделить навигацию. Интересное сочетание цветов заострит внимание на любой точке сайта. Ведь люди реагируют на цвета и остальные зрительные явления бессознательно. Цветовая гамма сайта с первой секунды расскажет посетителю, что это за сайт (развлекательный портал, интернет шопинг или онлайн-газета). Поэтому интересные, умело, смешанные краски на сайте занимают не последнее место для того, кто хочет популяризировать свой проект.

Очень много маркетинговых кампаний строиться на цвете. Маркетологи, дизайнеры и психологи верят в то, что используя определенный цвет можно внушить что-то потребителю и заставить его купить больше. Правильная цветовая схема – фактор не решающий, но однозначно важный. Примеры цветовых гамм известных брендов.

– Cadbury. Оттенки фиолетового выделяют Cadbury среди «шоколадных братьев». Если задача была именно такова – уйти от традиционной шоколадной гаммы, то выполнена она очень хорошо.

– Coca-Cola. Красно-бело-серая гамма. Белый и светлые оттенки серого концентрируют на красном. «Чистый» красный в таком сочетании энергичный, создает чувство драйва, возбуждения.

– Colgate. Красный, синий и белый – цвет полосок зубной пасты Colgate. Красный – энергия, белый – блеск и сияние, синий – свежесть.

– Duracell. В гамме два основных цвета – черный и оранжевый. Оранжевый – энергичный цвет, поэтому отлично подходит такому продукту, как «батарея». Черный, в свою очередь, добавляет солидности.

– Gillette. Лучше для мужчины нет! Синий – мужской цвет. К тому же это цвет свежести, поэтому отлично гармонирует с водно-гигиеническими процедурами. Оранжевый используется для акцентов.

– Gucci. На сайте Gucci основная гамма – светло коричневая. Эти цвета ассоциируются с кожей, чем-то натуральным. Кроме того, дизайнерам удалось это сделать изысканно и стильно.

– Ikea. Сайт Ikea выполнен в светлых, ярких тонах, в духе философии компании – просто, но со вкусом.

– L’Oreal. Пурпурная гамма – стильность, исключительность, уверенность в себе, сексуальность. В общем, все то, что следует внушить женщине.

– M&M’s. Основные цвета бренда M&M’s – красный и желтый. Как известно красно-желтая гамма возбуждает аппетит. Возьмите тот же McDonalds! Но, в данный момент, маркетинговые изыскания привели к тому что на ряду с основными, используются практически всевозможные цвета.

– Nestle. Бело-серо-голубое решение. Как представляется, внушает доверие, создает имидж надежной компании. Что и требуется таком гиганту.

– Nokia. Бело-сине-зеленая гамма свежая, чистая и ясная. Она достаточно серьезная, но не строгая, без явной половой принадлежности.

Существует несколько методов создания цветовой схемы для дизайна будущего сайта:

– создание цветовой схемы, используя классический цветовой круг. Данный метод довольно сложный и применяется на практике только опытными и профессиональными дизайнерами и художниками. Для

начинающего Web-разработчика данный метод вряд ли подойдет, из-за своей сложности. Сама методика создания цветовой схемы заключается в том, что, имея основной (главный) цвет цветовой схемы мы создаем несколько цветовых схем состоящих из 4 - 6 цветов, основываясь на принципах построения классических цветовых схем. При этом 2 цвета в каждой цветовой схеме бронируются для текстовых фонов и цвета основного шрифта. Далее из всего перечня полученных схем выбирается наиболее понравившаяся схема. При этом не маловажную роль будут играть личные качества человека, в основном творческие. Стоит понимать, что ошибки при таком методе создания цветовой схемы могут негативно отразиться на дизайне в целом. Поэтому использование метода на основе классического цветового круга при создании цветовой схемы, должно происходить лишь в тех случаях, когда Вы как разработчик имеете достаточный опыт и знания в разработке цветовых схем для web-дизайна;

– создание цветовой схемы, используя компьютерные программы по созданию цветовых схем. Данный способ не менее эффективный и используется Web-разработчиками гораздо чаще, чем предыдущий. На сегодняшний день программ для разработки цветовых схем в интернете можно найти очень большое количество. Причем предоставляются они в основном бесплатно. Принцип работы, каждой из таких программ строится на алгоритме создания классических цветовых схем. Однако более сложные в разработке программы позволяют использовать не только компьютерные алгоритмы, но и фантазию дизайнера;

– создание цветовой схемы, используя фотографию либо другое изображение. Данный способ, как и первый способ в этом перечне, довольно сложен в применении, поэтому требует определенных навыков. Однако, используя данный метод можно наиболее точно подобрать все необходимые сочетания цветов и их оттенков. Для реализации данного метода можно использовать самые различные фотографии или изображения, основным цветом на которых является выбранный вами основной цвет или близкий к нему оттенок будущей цветовой схемы. Предпочтение при выборе фотографий в основном отдается фотографиям цветов, бабочек, различным композициям и т.д. Иногда это может быть просто стопка отглаженного белья или одежды, поэтому не стоит себя ограничивать в выборе, а предоставлять волю интуиции и вкусу;

– создание цветовой схемы, на основе уже существующих цветовых схем. Данный метод тоже имеет право на существование, и ни как не является плагиатом. Доработанная цветовая схема уже существующего сайта или понравившегося другого дизайна может превзойти своего предшественника, как в эффективности, так и в дизайнерском решении.

Однако не стоит увлекаться заимствованием, не редко это приводит к тому, что Web-разработчик, увлекшись процессом, не замечает, как копирует не только цветовую схему понравившегося сайта, но и дизайн данного сайта в целом.

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

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

РАЗРАБОТКА ОНЛАЙН СИСТЕМЫ АВТОМАТИЗАЦИИ ОБРАБОТКИ ЗАЯВОК ДЛЯ ПРЕДПРИЯТИЯ

Роль интернета в жизни современного человека сложно переоценить. Интернетом в наше время пользуется более 30% населения Земли, а это немного немало около 1,500,000,000 человек. Еще в 1992 году им пользовалось всего 100 человек.

Число пользователей интернета растет высокими темпами. К 2018 году интернет будет почти в каждой семье. Телевидение уйдет в прошлое. Через интернет будут оплачивать коммунальные услуги, заказывать еду на дом, хотя, в принципе, это уже возможно и сейчас. А главное, в будущем многие будут выполнять свою работу, не выходя из дома, экономя этим свое время, которое можно будет провести с близкими людьми. Эти времена уже не за горами.

Пользователи уже сегодня стремятся решать свои вопросы и проблемы не выходя из дома. Одной из таких проблем является поиск работы. Если раньше для решения этой задачи необходимо было изучать тематические издания на предмет появления нужных вакансий, искать интересные предложения на рынке труда, то сейчас в большинстве случаев, можно свести свои трудозатраты к минимуму, благодаря появлению специализированных сайтов. На таких сайтах пользователи могут искать интересные их вакансии, а также публиковать свои резюме для потенциальных работодателей.

Одной из таких проблем является поиск работы. Если раньше для решения этой задачи необходимо было изучать тематические издания на предмет появления нужных вакансий, искать интересные предложения на рынке труда, то сейчас в большинстве случаев, можно свести свои трудозатраты к минимуму, благодаря появлению специализированных сайтов. На таких сайтах пользователи могут искать интересные их вакансии, а также публиковать свои резюме для потенциальных работодателей.

В следствие вышесказанного был разработан программный комплекс для автоматизации приема заявок от пользователей. Заявки делятся на три типа: вакансии, резюме и курсы. Любой пользователь ресурса,

заполнив форму, может добавить на сайт вакансию, резюме или курс, информация о котором сохранится в базе данных. Например, в форме добавления вакансии пользователю предлагается указать должность, требования к кандидату, контактную информацию и выбрать нужную категорию, в которой будет опубликовано объявление. Для защиты от автоматических заполнений формы реализована защита с помощью CAPTCHA. После прохождения модерации, добавленная информация появляется на сайте.

Всем пользователям ресурса также доступна возможность просмотра списка добавленных ранее заявок. Реализована возможность как просмотра полного списка объявлений, так и вывод заявок определенной категории/направления. Вывод объявлений производится постранично.

Для отображения другого информационного контента реализована возможность создавать на сайте статические страницы. Например, это могут быть страница контактов организации, страница с ценами на размещение платных вакансий и курсов, а также другие информационные страницы с полезной информацией.

Для организаций предусмотрена возможность размещения информации на платной основе. После заключения договора и оплаты размещения, информация о заявке добавляется на сайт. Такая заявка имеет ряд преимуществ перед бесплатным размещением. Во-первых, платные объявления всегда отображаются вверху страницы и чаще попадают на глаза посетителям ресурса. Во-вторых, платное объявление имеет более гибкие возможности в оформлении: здесь можно добавить логотип организации, ссылки на страницы своего сайта, а также указать несколько позиций, по которым производится поиск кандидатов.

Для администратора интернет-ресурса предусмотрен отдельный интерфейс. После успешного прохождения авторизации, администратор попадает в панель управления сайтом. В разделе модерации показаны добавленные посетителями объявления, которые еще не опубликованы на сайте. Администратору доступны функции одобрения, удаления, а также редактирования информации, указанной в объявлении.

Также, администратору доступны функции добавления, редактирования и удаления платных объявлений. При их добавлении необходимо указать срок размещения, по истечении которого, объявление будет снято с публикации на сайте.

В администраторском интерфейсе реализована возможность просмотра списка опубликованных статических страниц с возможностью их редактирования или удаления. Для добавления новой статической страницы, администратору нужно указать ее заголовок, описание и добавить контент страницы. Для добавления контента используется wysiwyg

редактор, который позволяет форматировать контент, не имея специальных навыков HTML-разметки.

Для функционирования программного комплекса требуется веб-сервер Apache. Программный комплекс реализован на PHP5, в качестве базы данных используется MySQL.

Для более быстрого взаимодействия с сервером баз данных используется расширение mysqli. MySQLi (MySQL Improved) – расширение драйвера реляционных баз данных, используемого в языке программирования PHP для предоставления доступа к базам данных MySQL. MySQLi является обновленной версией драйвера PHP MySQL, и дает различные улучшения в работе с базами данных. Разработчики языка программирования PHP рекомендуют использование MySQLi при работе с сервером MySQL версий 4.1.3 или новее (они используют новую функциональность). Основным преимуществом MySQLi является то, что при помощи предварительно объявляемых операторов, могут быть предотвращены так называемые SQL-инъекции.

Архитектура построена согласно MVC-концепции. MVC(Model-view-controller, «Модель-представление-поведение», «Модель-представление-контроллер») – это шаблон проектирования приложений, при котором управляющая логика поделена на три отдельных компонента таким образом, что модифицирование одного из них дает минимальное влияние на остальные. Шаблон MVC хорошо применять при создании сложных проектов, где необходимо отделить работу php программиста (или разделить группу программистов на отделы), дизайнера, верстальщика, и т.д. Все файлы функционально разделены на файлы-контроллеры (Controllers), файлы моделей (Models) и файлы представления (Views).

Исходя из вышесказанного, можно сделать вывод о том, что разработанный программный комплекс подходит для решения задачи построения интернет-ресурса с реализацией автоматического приема заявок от пользователей и структурированного вывода их на сайте. Разработанный программный комплекс поддается масштабированию и может быть использован для построения аналогичных сервисов в сети интернет.

Литература

1. Макконнелл, С. Совершенный код. Мастер-класс. / С. Макконнелл – М.: «Русская редакция», 2008. – 896 с.
2. Зандстра, М. PHP: объекты, шаблоны и методики программирования / М. Зандстра – М.: «Вильямс», 2013 – 560 с.
3. Конверс, Т. PHP 5 и MySQL. Библия пользователя/ Т. Конверс – М.: «Вильямс», 2007. – 1216 с.
4. Котеров, Д. PHP 5 В Подлиннике / Д. Котеров, А. Костарев – СПб.: «БХВ-Петербург», 2012. – 1104 с.

О.В. Никитина (УО «ГГУ им. Ф. Скорины», Гомель)

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

РАЗРАБОТКА САЙТА ПРЕДПРИЯТИЯ В JOOMLA 3

На сегодняшний день существует множество самых разных систем управления. Одни более простые в управлении, но их возможностей хватает только для создания блогов и сайтов визиток с небольшим количеством страниц. Другие сложные, но вместе с тем функциональные, позволяющие реализовать создание интернет-магазина со множеством функций, возможностей и большим каталогом товаров. Сейчас мы предлагаем рассмотреть плюсы и минусы системы, которая сочетает в себе относительную простоту использования и большой функционал – это Joomla. Самая популярная система управления в мире, была создана на основе Mambo 4.5.2.3 и являет собой набор написанных на языке PHP скриптов.

Преимущества CMS Joomla: бесплатный доступ, открытый код системы, система конструктора модулей, гибкость, возможность свободной реализации разных дополнений в любой системе и их взаимодополняемость, доступность справочной информации, низкое ресурсопотребление, локализация версии.

Бесплатный доступ это важное преимущество для начинающих веб-мастеров, а так же для тех, кто не хочет или не видит смысла в том, чтобы платить за обслуживание.

Открытый код системы при наличии определенных умений можно легко изменить сам движок.

Отличная реализация управления и работы с модулями, шаблонами, компонентами и расширениями – это позволяет в короткие сроки и с минимумом усилий со стороны разработчика создать интернет-магазин, большой корпоративный сайт для крупной компании или информационный портал.

Гибкость – используя один и тот же шаблон можно легко поменять вид одного компонента или целого модуля без внесения исправлений в код самого движка, но для этого потребуются знания PHP, CSS и JS. При этом все внесенные изменения сохраняются в шаблоне.

Многообразие шаблонов, модулей и дополнений – и этот плюс вряд ли нуждается в комментариях, поскольку даже начинающий разработчик сразу его оценит.

Для некоторых пользователей сайт на Joomla имеет преимущество в виде наличия вспомогательных функций, упрощающих работу.

Следующие плюсы CMS Joomla менее значимы, но играют весомую роль при выборе системы управления.

Большое количество обучающих материалов – сюда входит не только официальная документация, но также видео, электронные пособия, форумы и сообщества по технической поддержке.

Низкое потребление ресурсов – это важно, когда требуется не простой сайт-визитка, а какой-то глобальный проект с большим количеством разнообразной информации.

Русификация (локализация) – перевод множества модулей на русский язык.

Отрицательные моменты системы: некоторая несопоставимость некоторых модулей в разных версиях, избыточность кода, в следствие избыточности кода увеличивается время загрузки, существует необходимость доработки программного кода и вида шаблона (средствами PHP и CSS и JS).

Литература

1. Декстер, М. Joomla! Программирование / М. Декстер, Л. Лэндри – М.: ООО «И.Д. Вильямс», 2013. – 592 с.

2. Клименко, Р. А. Веб-мастеринг на 100% / Р. А. Клименко – СПб.: Питер, 2013. – 512 с.

3. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript / Р. Никсон. – СПб.: Питер, 2013. – 496 с.

4. Фримен, Э. Изучаем HTML, XHTML и CSS / Э. Фримен. – СПб.: Питер, 2013. – 656 с.

Д.А. Новик (УО «ГГУ им. Ф. Скорины», Гомель)

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

МОДЕРНИЗАЦИЯ ЛВС ДЛЯ ОТДЕЛА ОБРАЗОВАНИЯ, СПОРТА И ТУРИЗМА ЖЕЛЕЗНОДОРОЖНОГО РАЙОНА г. ГОМЕЛЯ

Данный проект посвящен модернизации уже существующей сети. Практика показывает, что модернизацию локальных вычислительных сетей необходимо проводить приблизительно каждые 3-6 лет. Это обусловлено не только развитием новых технологий и увеличением объемов передачи данных, но и необходимостью решать задачи, стоящие перед предприятием в данный период развития. Модернизацию необходимо проводить в свете уже существующих ИТ-решений в перспективе горизонтов развития.

Однако часто причиной для модернизации является неправильный подход при изначальном построении сети. Это связано с тем, что большинство предприятий при построении первой сети не имели четко сформулированных задач с позиции развития ИТ-инфраструктуры.

Постепенное увеличение количества обслуживающих компьютеров приводит к бессистемному «латанию дыр» в попытке избежать перегрузок сети в узких местах. В результате, это может привести к тому, что данную сеть надо переделывать, а не модернизировать.

Правильный подход к построению сетей лежит в закладывании ресурсов для последующего развития, исходя из будущих возможных потребностей. Правильно построенная сеть изначально масштабируема. Сеть должна быть наращиваемой для подключения, например, более мощного сервера, для облегчения обслуживания компьютеров, новых приложений или систем. Все факторы необходимо учитывать, чтобы заложить некий потенциал роста.

Очень часто именно системный администратор заинтересован в развитии сети и усовершенствовании ИТ-инфраструктуры, именно он, зачастую, подталкивает руководство к принятию решения о модернизации локальной сети. Но, в первую очередь, в результате модернизации ЛВС повышается эффективность работы руководителя, позволяя видеть более целостную картину работы предприятия, Это дает возможность оперативно собирать информацию, автоматизировать регулировку процессов.

Для начала опишем уже существующую сеть данного предприятия.

Локальная сеть, основана на протоколе IPv4, которая использует специальные адреса, диапазон адресов 192.168.0.0 – 192.168.255.255, то есть адреса класса С.

При анализе сети была использована программа 10-страйк, в процессе ее работы была получена схема сети данного предприятия (рисунок 1).

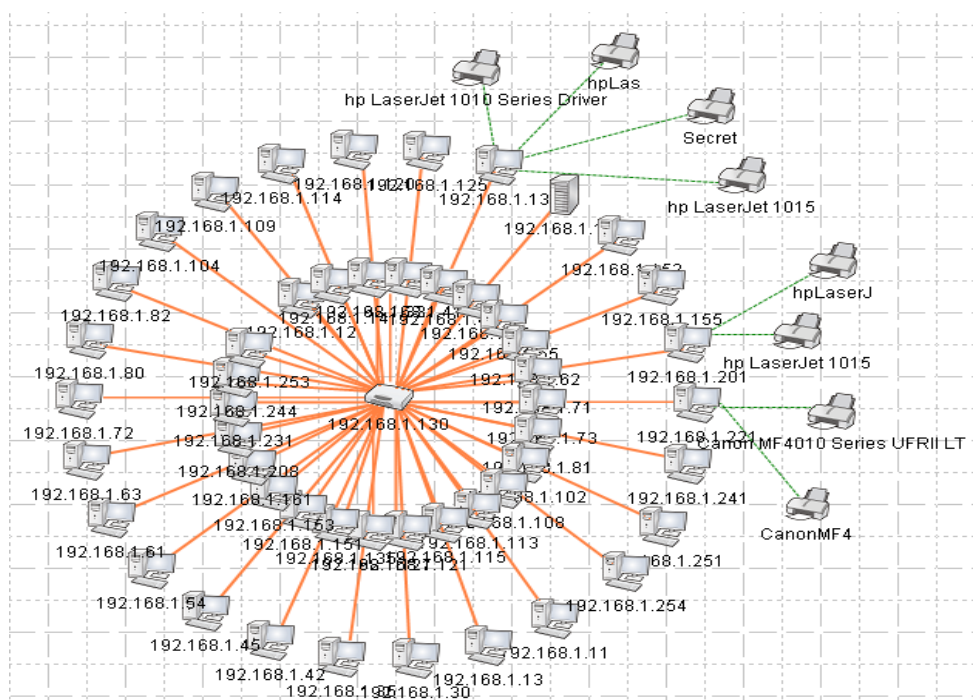


Рисунок 1 – Схема сети

Также при анализе данной сети, были собраны все компьютерные mac-адреса.

При модернизации сети можно выделить несколько этапов:

- первичное исследование;
- разработка, составление технического задания;
- подготовка оборудования;
- монтаж ЛВС;
- последующее обслуживание.

Стоит отметить, что при построении любой сети основополагающим фактором является стоимость. Поэтому целесообразно при ее модернизации использовать уже имеющееся сетевое оборудование, в крайнем случае, добавлять необходимые элементы сети. Также стоит отметить, что в данной сети имеется несколько серверов баз данных, которые, впоследствии, можно будет вынести на отдельный сервер для оптимальной работы всей системы. Модернизация данной системы необходима для того, чтобы систематизировать все данные предприятия и улучшить работоспособность модернизируемой сети. Во-первых, необходимо будет задать определенный диапазон для каждого кабинета с расчетом увеличения рабочих мест в будущем.

Для того чтобы построить локальную вычислительную сеть (ЛВС), необходимо, в первую очередь, выбрать топологию построения, от которой будут зависеть характеристики планируемой сети. Термин «топология» отражает физическое расположение серверов, рабочих станций, компьютеров, кабелей, а при наличии, также и расположение коммутаторов, концентраторов и маршрутизаторов. Фактически, говоря простым языком, это «карта» сети, которая выбирается в зависимости от потребностей пользователей. Выбор топологии влияет на состав и технические характеристики сетевого оборудования, способы управления системой и возможность дальнейшего расширения сети.

Базовыми топологиями для построения ЛВС являются топологии «шина» (bus), «кольцо» (ring) и «звезда» (star).

Необходимо искать оптимальные варианты долговременного использования локальной вычислительной сети. Вынужденные простои обходятся дорого, поэтому необходимо использовать специальные приборы и инструменты, повышающие надежность ЛВС.

Гарантией надежной и эффективной работы предприятия и сохранности данных является защищенность локальных вычислительных сетей. Грамотно спроектированные локальные вычислительные сети должны обеспечивать защиту от несанкционированного доступа через телефонную линию или интернет.

Также необходимы мощные средства мониторинга локальной вычислительной сети, используемые для оперативной диагностики и

устранения помех и неисправностей, чтобы исключить возможные простои, упомянутые выше. Существует много продуктов, применяемых для сбора данных о состоянии ЛВС и ее параметрах, например, средства RMON. Web-интерфейс обеспечивает возможность управления сетью практически в любом месте с помощью удаленного доступа.

Поставленные задачи:

- возможность совместного использования ресурсов сети (файлов, принтеров, модемов и т.д.);
- оперативный доступ к любой информации сети;
- надежные средства резервирования и хранения информации;
- защита информации от несанкционированного доступа;
- возможность использования современных технологий, в частности, системы электронного документооборота, сетевых баз данных, приема/передачи факсов, доступа в Интернет.

В конечном итоге, можно сделать вывод, что к задаче по модернизации сети надо подходить комплексно. При этом необходимо учитывать всевозможные факторы, часто лежащие в совершенно разных плоскостях, но являющиеся дополнениями друг другу, и игнорирование одних может свести на нет эффективность других.

Литература

1. Ильина, О.П. Вычислительные системы, сети и телекоммуникации. / О.П. Ильина, В.Л. Бройдо – СПб: Питер, 2008. – 560 с.
2. Новиков, Ю.В. Локальные сети: архитектура, алгоритмы, проектирование. / Ю.В. Новиков, С.В. Кондратенко – М.: ЭКОМ, 2000. – 780 с.
3. Уолрэнд, Дж. Телекоммуникационные и компьютерные сети./ Дж. Уолрэнд. – М.: Постмаркет, 2007. – 659 с.
4. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы./ В.Г. Олифер, Н.А. Олифер. – СПб.: "Питер", 2004. – 568 с.
5. Нанс, Б. Компьютерные сети: Пер. с англ. / Б. Нанс. – М.: "БИНОМ", 2006. – 789 с.

А.А. Носиков (БТЭУ ПК, Гомель)

Науч. рук. **С.В. Карпенко**, канд. эконом. наук, доцент

МОДЕЛИРОВАНИЕ ПРОЦЕССА РАЗРАБОТКИ САЙТА

Сайт – совокупность электронных документов частного лица или организации в компьютерной сети, объединённых под одним адресом.

Все сайты в совокупности составляют Всемирную паутину, где коммуникация объединяет сегменты информации мирового сообщества в единое целое – базу данных и коммуникации планетарного масштаба. Для прямого доступа клиентов к сайтам на серверах был специально разработан протокол HTTP.

В настоящее время в Беларуси существует множество фирм по созданию сайта. Наиболее распространены следующие:

- ArtisMedia (<http://www.artismedia.by/>)
- СайтоДром (<http://www.saitodrom.by/>)
- Иквadrat (<http://iquadart.by/>)
- МастерЛинк (<http://masterlink.by/>)

ArtisMedia. Это студия интернет-проектов для компаний-лидеров в своей области, которые желают открыть новые и эффективно использовать существующие возможности Интернета для своего развития. Благодаря своему уникальному опыту успешных интернет-проектов ArtisMedia уже сегодня обеспечивает своих заказчиков доступными по цене качественными интернет-услугами и создает для них значительное конкурентное преимущество, что дает им еще один источник поступления средств и уверенность в будущем.

СайтоДром. Сайтодром предоставляет своим клиентам следующие услуги:

- Создание или заказ сайта;
- Создание каталога товаров;
- Создание интернет-магазина;
- Создание доски объявлений;
- Создание фотогалереи.

Иквadrat. Процесс создания сайта в этой студии состоит из 4-х этапов, которые позволяют максимально быстро предоставить заказчику необходимый продукт. 1) Планирование. Проект-менеджер разрабатывает техническое задание и план проекта. 2) Дизайн. В первую очередь создаются модульные сетки страниц, которые затем превращаются в дизайн. Модульные сетки, так же как и дизайн страниц, предоставляются заказчику на утверждение. 3) Верстка и программирование. Мы разрабатываем сайт, согласно функционалу, определенному на этапе планирования. 4) Тестирование. При тестировании сайта анализируются:

- оптимизированность под поиск;
- скорость загрузки;
- кроссбраузерность;
- валидность.

RНР-версия (запрограммированная версия сайта) проходит проверку на тестовом сервере, а затем финальную проверку на сервере заказчика.

МастерЛинк. Masterlink.by – Создание и продвижение прибыльных сайтов. Создание сайтов и продвижение сайтов – вот что делает эта компания действительно хорошо. Компания Masterlink.by – надёжный партнёр. Реализовав более 200 проектов в различных тематиках в Беларуси, России, Украине, Великобритании и Корее, они продолжают концентрироваться на выгодах, которые получают их клиенты, клиенты клиентов клиентов, а значит и они сами.

Модель процесса разработки сайта в программной среде AllfusionProcessModeler представлена на рисунке 1.

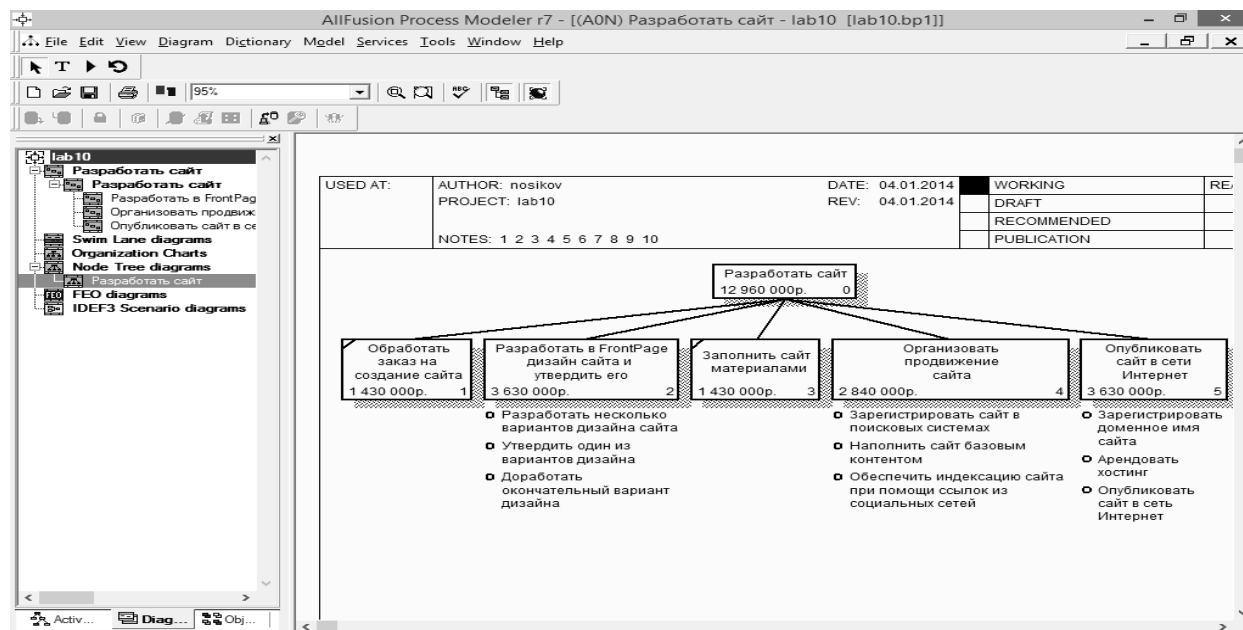


Рисунок 1 – Моделирование бизнес-процесса создания сайта

Процесс разработки сайта начинается с поступления и обработки заказа от клиента на создание сайта. Основная часть работы – получение информации от клиента о требованиях, предъявляемых к сайту, разъяснение интересующих клиента вопросов. Далее необходимо получить информацию, которая будет служить основой для создания сайта.

Разработка и утверждение дизайна сайта. На этом этапе происходит разработка нескольких вариантов дизайна сайта. После учитываются предпочтения клиента и выбирается конкретный вариант дизайна сайта. В последнем моменте выбранный вариант дизайна сайта дорабатывается сотрудниками фирмы, после чего и получается сайт, готовый заполнения нужными материалами.

Наполнение сайта информационными материалами по предпочтениям самого клиента. Этот этап является немаловажным этапом создания сайта. Контент сайта может включать в себя большое количество материала, представленного в различной форме и максимально

адаптированного для восприятия. Создание уникального контента – трудоемкий и щепетильный процесс, требующий немало сил и времени. Это один из важных этапов разработки сайта, ведь сайт, не отличающийся информативностью, не имеет шансов на успех.

Продвижение сайта. Только разработать и разместить сайт в интернете мало. Для абсолютного большинства сайтов продвижение должно начинаться с поисковой оптимизации, или seo. Её целью является повышение рейтинга ресурса у поисковиков.

Публикация сайта в сети Интернет. Вначале на специальных сайтах регистрируется доменное имя или же адрес нашего сайта. После арендуется пространство для хранения содержимого нашего сайта и самого сайта соответственно, так называемый хостинг. В конечном итоге на данном этапе происходит само непосредственно опубликование сайта в сеть Интернет.

В процессе моделирования решены следующие задачи:

1. Моделирование бизнес-процесса AS-IS с помощью AllFusion Process Modelerr7.

2. Функционально-стоимостной анализ. Построены матрицы парного сравнения для функций модели. Они позволили получить количественную оценку качественного признака и определить долю каждой из анализируемых функций в списке функций.

3. Построена функционально-стоимостная диаграмма, наглядно отображающая соотношение затрат и значимостей работ. Она позволила выявить два процесса для перепроектирования: процесс «Обработать заказ на создание сайта» и процесс «Заполнить сайт материалами».

4. Моделирование бизнес-процесса TO-BE позволило получить следующий экономический эффект. Общие затраты на выполнение всего процесса создания сайта снизились на 400000 бел. руб.

Литература

1. Создание сайтов – ООО АртисМедиа. Электронный ресурс: <http://www.artismedia.by/>. Дата доступа 07.01.2014

2. Создание сайтов в Беларуси: конструктор сайтов Сайтодром - возможность создать сайт бесплатно. Электронный ресурс: <http://www.saitodrom.by/>. Дата доступа 07.01.2014

3. «Икварт» = разработка сайтов, создание сайтов в Беларуси, разработка флеш-анимации, разработка фирменного стиля. Электронный ресурс: <http://iquadart.by/>. Дата доступа 07.01.2014

4. Создание и продвижение сайтов. Электронный ресурс: <http://masterlink.by/>. Дата доступа 07.01.2014

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

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

АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ ПЕРСОНАЛОМ НА ПРЕДПРИЯТИИ

На современном этапе развития общества, квалифицированные кадры играют важнейшую роль, являясь одним из главных производственных активов предприятия. На предприятии любого масштаба необходимо вести точный учет кадров, оптимально распределять нагрузку между специалистами, мотивировать персонал на выполнение поставленной задачи, организовывать отлаженную работу, имеющую целью конечный результат, значимый для всего предприятия.

Эффективное руководство кадрами – это четко выстроенная система использования кадровых ресурсов предприятия. При наличии оптимального количества квалифицированных сотрудников такая система позволяет достигать поставленных целей в кратчайшие сроки и с минимальными затратами. Она помогает увеличивать продуктивность работы, расширять инновационную деятельность, повышать удовлетворенность клиентов, обеспечивать целостность бизнес-процессов.

Основная проблема, с которой сталкиваются сотрудники отдела кадров – это большая трудоемкость управления, огромное количество задач, функций, процессов, которыми необходимо оперативно и качественно управлять.

Все вышеперечисленные проблемы практически невозможно решить без применения современных средств сбора и обработки информации. Именно с использованием современных программных комплексов достигаются высокие результаты в скорости получения информации и удобства работы с ней. Весь этот процесс перехода от старых принципов работы к информационным технологиям называется автоматизацией.

Автоматизация управления персоналом предполагает создание общей системы, при которой осуществление операций было бы простым и высокопроизводительным. Для этого настраиваются необходимые программы, задействованные в процессе управления сотрудниками.

Возможности автоматизации:

1 Получение руководством предприятия полной аналитической информации, необходимой для принятия решений.

2 Снижение затрат на основные HR-функции для руководителей службы персонала.

3 Автоматизация работы отдела кадров трудоемкие по учету персонала и ведению документации.

4 Автоматическое формирование необходимых отчетов.

Автоматизация управления персоналом решает следующие задачи:

1 Работа с различными типами структур и ведение штатного расписания.

2 Внедрение технологий для упрощения ведения должностных обязательств и требований к сотрудникам, а также автоматизация создания и изменения всех должностных инструкций.

3 Ведение и управление информацией о сотрудниках.

4 Разработка методов материального стимулирования различных категорий сотрудников.

5 Планирование и контроль над деятельностью и взаимодействием сотрудников, а так же создание сложных планов.

6 Контроль квалификации сотрудников.

7 Анализ эффективности денежных вложений и финансовых затрат по персоналу.

8 Формирование отчетов для руководства предприятия.

Преимущества автоматизации:

1 Поддержка различных бизнес-процессов для управления персоналом.

2 Организация информационного пространства документооборота в подразделениях, исключение дублирования информации.

3 Синхронизация данных между отделами, централизация хранения и обновления информации.

4 Получение отчетности по всему предприятию и упрощение работы по формированию списков и справок.

5 Разграничение прав доступа к данным, фиксирование времени и места внесения изменений в информационную базу.

6 Легкость внедрения и освоения работы с системой для любого пользователя.

Таким образом, автоматизация управления персоналом является действенным инструментом, облегчающим не только действия отдела кадров, но и обеспечивающим включенность каждого сотрудника в работу предприятия.

С.С. Пигуль (УО «ГГУ им. Ф. Скорины», Гомель)

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

КОМПЬЮТЕРНАЯ ГРАФИКА В СРЕДЕ 3D STUDIO MAX

История развития компьютерной графики началось уже в 20 веке и продолжается сегодня. Компьютерная графика в настоящее время сформировалась как наука об аппаратном и программном обеспечении для разнообразных изображений от простых чертежей до реалистичных

образов естественных объектов. Компьютерная графика используется почти во всех научных и инженерных дисциплинах для наглядности и восприятия, передачи информации. Применяется в медицине, рекламном бизнесе, индустрии развлечений и т. д. Без компьютерной графики не обходится ни одна современная программа.

Конечным продуктом компьютерной графики является изображение. Это изображение может использоваться в различных сферах, например, оно может быть техническим чертежом, иллюстрацией с изображением детали в руководстве по эксплуатации, простой диаграммой, архитектурным видом предполагаемой конструкции или проектным заданием, рекламной иллюстрацией или кадром из мультфильма.

Под компьютерной графикой обычно понимают автоматизацию процессов подготовки, преобразования, хранения и воспроизведения графической информации с помощью компьютера. Компьютерная графика нашла широкое применение как средство художественного оформления, в частности для создания художественных кинофильмов, телерекламы, видеоклипов музыкальных исполнителей.

Объектом исследований при создании компьютерной графики являются трехмерные объекты и их представление в составе сцен. Инструментом для создания трехмерных анимационных роликов служит ЭВМ. Для создания таких роликов требуется специальное программное обеспечение (ПО). В настоящее время, в качестве наиболее распространенного, эффективного и доступного ПО такого рода является пакет 3D Studio Max, который предлагает самые совершенные возможности современных анимационных систем. Эта программа позволяет создавать анимационные заставки, рекламные ролики, анимационные сцены в Internet, музыкальные клипы, анимационные фильмы; обрабатывать видеоматериалы, создавать видео монтаж, отображать движения в анимации; позволяют строить анимационные сцены и сюжеты, работать с видеоматериалом и звуковыми файлами. Целью преподавания дисциплины является доступное изложение теоретических основ и программного обеспечения для создания, преобразования и анимации трехмерных графических объектов и сцен.

А.Н. Пинчук (УО «ГГУ им. Ф. Скорины», Гомель)

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

АВТОМАТИЗАЦИЯ ДЕБИТОРСКОЙ КРЕДИТОРСКОЙ ЗАДОЛЖЕННОСТИ ДЛЯ ОАО «СТРОИТЕЛЬНЫЙ ТРЕСТ № 14»

В настоящее время персональный компьютер играет очень важную роль во всех сферах деятельности человека. Он позволяет решать

широкий спектр задач, начиная с компьютерных игр и заканчивая сложнейшими расчетами. При этом наибольший успех имеет тот, кто может качественно обработать нужную ему информацию за короткий промежуток времени. Однако до сих пор нет средств, позволяющих в достаточной мере автоматизировать процесс ведения документации и отчетности.

Одной из составных задач можно рассматривать проблему составления статистической отчетности, а так же оперативную корректировку бухгалтерской задолженности при возникновении необходимости в этом.

О своевременности и актуальности рассматриваемой проблемы говорит тот факт, что протяжении длительного времени, было достаточно сложно получить оперативную информацию, если при этом в формировании этой информации участвовали несколько структурных подразделений или отделов. Для любой организации получение достоверной информации в сжатые сроки очень важно. Таким образом, возникает необходимость приобретения или создания такого программного продукта, который удовлетворил бы запросы всех заинтересованных. Так как рынок предлагает ограниченное количество громоздких и дорогих решений, то в данной сфере гарантирована высокая 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).

База данных является ядром информационной системы. Применение количественных метрик физических схем БД позволяет разработчикам БД:

- изучить сложность разработанной физической схемы базы данных;
- оценить объем работ, выполненных разработчиком физической схемы БД;
- оценить усилия по реализации физической схемы БД;
- выбрать наилучшую физическую схему БД из нескольких альтернативных вариантов.

СУБД вместе с прикладными программами называют банком данных.

Одно из основных назначений СУБД – поддержка программными средствами представления, соответствующего реальности.

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.

Концептуальное проектирование БД абсолютно не зависит от таких подробностей ее реализации, как тип выбранной целевой СУБД, набор создаваемых прикладных программ, используемые языки программирования, тип выбранной вычислительной платформы, а также от любых других особенностей физической реализации.

Физическое проектирование БД – это процесс подготовки описания реализации БД на вторичных запоминающих устройствах; на этом этапе рассматриваются основные отношения, организация файлов и индексов, предназначенных для обеспечения эффективного доступа к данным, а также все связанные с этим ограничения целостности и средства защиты.

Для ввода в ПК полученное описание должно быть представлено в терминах специального языка описания данных, который входит в комплекс средств СУБД.

Наиболее перспективным на сегодняшний день является использование для проектирования БД объектно-реляционных систем управления базами данных, сочетающих возможности объектного и реляционного моделирования.

Проектирование данных играет ключевую роль в жизненном цикле информационных систем. Также важно знать, что представляют собой процессы логического (концептуального) и физического моделирования данных.

Таким образом, основная цель проектирования БД – сокращение избыточности хранимых данных и, следовательно, экономия объёма используемой памяти, уменьшение затрат на многократные операции, обновлении избыточных копий и устранение возможности возникновения противоречий из-за хранения в разных местах сведений об одном и том же объекте.

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

Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА WEB-САЙТА АВТОМАТИЗАЦИИ ПРОЦЕССА КУПИ-ПРОДАЖИ УЧЕБНЫХ МАТЕРИАЛОВ

Разрабатываемый сайт позволит:

- покупать на нем готовые работы, которые находятся в каталоге предметов;

- загружать на сайт работ, с указанием собственной цены, которую в будущем смогут покупать другие пользователи сайта;

- заказывать работу: написание диплома, курсовой работы, реферата;

- зарегистрироваться в качестве исполнителя (автора) и выполнять поступающие заказы по определенным специализациям;

- администрация сайта имеет информацию (может ее просматривать, частично изменять, удалять) о зарегистрированных пользователях, авторах, о заказах на разработку (дипломов, курсовых и т.д.), о заказах на покупку готовых продуктов (дипломов, курсовых и т.д.), о заявках на вывод средств.

- принимать оплату несколькими способами: электронными деньгами WebMoney, а так же пластиковыми карточками Visa/MasterCard

Web-сайт разработан на PHP-фреймворке с использованием технологии ZendFramework, который позволяет разрабатывать веб-приложения и веб-сервисы. ZendFramework следует принципу веб-разработки MVC (ModelViewController), который в настоящее время, считается одним из лучших принципов веб-разработки. Все компоненты технология ZendFramework строится на принципе ООП, имеют возможность работать с многопоточностью, а так же поддерживают большое количество СУБД, включая MariaDB, MySQL, Oracle, IBM DB2, Microsoft SQL Server и др.

В ходе разработки проекта реализованы необходимые контроллеры и модули, при помощи которых сайт позволит регистрироваться и авторизоваться пользователям, заказывать учебные материалы, покупать готовые работы, просматривать отчеты по финансам в административной панели. Так же реализованы представления, которые отображают HTML информацию в браузере в виде страниц сайта.

Е.В. Пищик (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **В.Д. Левчук**, канд. тех. наук, доцент

РЕАЛИЗАЦИЯ ФУНКЦИЙ УЧЕТА ТРУДА В КОНФИГУРАЦИИ 1С:ПРЕДПРИЯТИЕ «ПЕРСОНАЛ. КАДРЫ. ЗАРПЛАТА»

В любой компании наиболее ценный ресурс – персонал. Поэтому как любой ресурс он требует учета и эффективного управления. Для управления человеческими ресурсами был разработан современный инструмент автоматизации задач управления персоналом, ведения кадрового учета и расчетов в соответствии с законодательством – «Зарплата и Управление Персоналом для Беларуси».

Данное прикладное решение является типовым решением и обеспечивает следующие функциональные возможности: планирование потребностей в персонале и затрат на персонал, обеспечение бизнеса кадрами, управление компетенциями и аттестация работников, управление обучением персонала, управление финансовой мотивацией персонала, эффективное планирование занятости персонала, учет кадров и анализ кадрового состава, трудовые отношения, в том числе кадровое делопроизводство, расчет заработной платы персонала, управление денежными расчетами с персоналом, включая депонирование, исчисление регламентированных законодательством налогов и взносов с фонда оплаты труда, отражение начисленной заработной платы и налогов в затратах предприятия.

Т.к. данное прикладное решение является типовым, то чтобы его внедрить в какую-либо компанию, его необходимо адаптировать с учетом всех требований, потребностей, особенностей и спецификой управления данного предприятия. Так, например, возникла необходимость реализовать более эффективные функции оперирования штатным расписанием.

Штатное расписание – это организационно-распорядительный документ, в котором сгруппированы структурные подразделения организации, должности (профессии) с указанием их количества с учетом одноименных наименований, квалификационных разрядов (классов, категорий), размеров должностных окладов (тарифных ставок), надбавок, доплат. Штатное расписание как нормативный документ предприятия фиксирует в сводном виде сложившееся на предприятии разделение труда между работниками, описанное в должностных (рабочих) инструкциях. Для работников служб персонала штатное расписание – документ для подбора кадров на вакантные места.

Все изменения каждой единицы штатного расписания фиксируются и сохраняются таким образом, что можно просмотреть всю историю изменений. Однако не предусмотрено возможности просмотра всех изменений всех единиц штатного расписания за определённый период.

Поэтому необходимо реализовать отчет, в котором будут отображены все изменения всех штатных единиц за выбранный период: параметры до изменения, дата изменения и параметры после изменения (номер единицы штатного расписания, количество ставок, размер тарифной ставки первого разряда, тарифный разряд, тарифный коэффициент, тарифный оклад, тарифный должностной оклад, надбавки).

В ходе создания отчета на закладке Данные были добавлены четыре необходимых реквизита: Название Организации, Название Подразделения, Начало Периода и Конец Периода. Тем самым в дальнейшем пользователю обеспечен выбор наименования организации, подразделения и периода, за который необходимо просмотреть изменения штатных единиц.

Также создана форма отчета, на которой необходимо разместить поля ввода для параметров наименования организации, подразделения организации, начала и конца периода, а также табличную часть, в которую выводится макет с результатом отчета, и непосредственно кнопку Сформировать. В модуль формы помещены обработчики событий, связанные с действиями формы (выбор параметров, нажатие кнопок и т.д.).

Макет целесообразно создавать не с помощью схемы компоновки данных, а вручную. Тип создаваемого макета – табличный документ. В табличном документе созданы заголовки отчета, с выводом общих параметров, шапка таблицы, в соответствии с которой выводятся данные в строки отчета, для строк отчета заданы все параметры для выводимых данных. Следует обратить внимание, что одной единице штатного расписания соответствуют две строки – информация до изменения и информация после изменения. Таким образом, удобно сравнивать какие именно изменения были произведены. Макет разбит на области, которым назначены имена. В модуле объекта отчета созданы процедуры для выборки, обработки, расчета и вывода данных.

Нужно создать процедуру, которая выводит строки по подразделениям очередного уровня. Для решения задачи была создана таблица повышений со следующими колонками: вид коэффициента, размер коэффициента, вид повышения, размер повышения, вид надбавки, процент надбавки, результат надбавки, тарифная ставка, расчетная тарифная ставка. В запросе к регистру сведений Повышения Тарифного Коэффициента выбраны поля вид коэффициента и размер коэффициента, установив, что поле Период данного регистра сведений должен быть больше параметра & Начало Периода и меньше параметра & Конец Периода. Аналогичным образом созданы запросы к регистрам сведений Повышения Тарифной Ставки и Надбавки По Штатному Расписанию Организаций, где выбираются соответствующие поля и задано то же условие.

Аналогично создана процедура по формированию отчета. Для этого создан запрос, который включает в себя данные таблиц регистров сведений Размер Тарифных Ставок и Штатное Расписание Организаций, документ Штатное Расписание и табличную часть Единицы Штатного Расписания документа Изменение Штатного Расписания. Данные таблицы необходимо связать для получения корректных данных, а также указать условия, налагаемые на таблицу (период должен находиться в интервале, указанном пользователем, а также должен осуществляться отбор по организации и подразделению). Из документов-регистраторов получено действие, которое было применено к единице штатного расписания при изменении.

Результаты тестирования процедур обсуждаются в докладе.

Литература

1. Габец, А.П. Реализация прикладных задач в системе 1С:Предприятие 8.2/ А.П. Габец, Д.В. Козырев, Д.С. Кухлевский, Е.Ю. Хрусталева – М.: ООО «1С-Публишинг», 2010 – 713 с.
2. 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.
3. Хрусталева, Е.Ю. Разработка сложных отчетов в 1С: Предприятие 8. Система компоновки данных/ Е.Ю. Хрусталева – М.: ООО «1С-Публишинг», 2009 – 513 с.

Е.В. Пищик (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **В.Д. Левчук**, канд. тех. наук, доцент

АВТОМАТИЗАЦИЯ ПОСТРОЕНИЯ ОТЧЕТОВ В КОНФИГУРАЦИИ 1С:ПРЕДПРИЯТИЕ «ПЕРСОНАЛ. КАДРЫ. ЗАРПЛАТА»

Конфигурация «Персонал. Кадры. Зарплата» является типовой. Чтобы ее внедрить в какую-либо компанию, необходимо адаптировать с учетом всех требований, потребностей, особенностей и спецификой управления данного предприятия.

В частности возникла необходимость реализовать отчет, в котором будут отображены все изменения всех штатных единиц за выбранный период: параметры до изменения, дата изменения и параметры после изменения (номер единицы штатного расписания, количество ставок, размер тарифной ставки первого разряда, тарифный разряд, тарифный коэффициент, тарифный оклад, тарифный должностной оклад, надбавки и т.д.). В ходе создания отчета необходимо создать четыре реквизита, которые в дальнейшем будут использоваться.

Также создана форма отчета, на которой размещены поля ввода для параметров а также табличную часть, в которую выводится макет с результатом отчета. В модуле формы определены обработчики событий, связанные с действиями формы.

Макет создан на основе табличного документа. В табличном документе создан заголовок отчета с выводом общих параметров, шапка таблицы, в соответствии с которой выводятся данные в строки отчета. Следует обратить внимание, что одной единице штатного расписания соответствует две строки: информация до изменения и информация после изменения. Таким образом, удобно сравнивать какие именно изменения были произведены.

В модуле объекта отчета созданы процедуры для выборки, обработки, расчета и вывода данных. Создана процедура, которая выводит строки по подразделениям очередного уровня, а также проверяет нужно ли осуществлять переход на следующую страницу. Далее реализован вывод строк в отчет и процедура расчета тарифного и должностного окладов. Аналогичным образом выведены строки с данными по единицам штатного расписания до изменения. Для этого достаточно получить данные регистра сведений, записанные за предыдущий день относительно даты, попавшей в выбранный период.

Таким образом, создан отчет, который необходим отделу кадров предприятия, использующего данную конфигурацию.

А.В. Плесский (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **Е.А. Ружицкая**, канд. физ.-мат. наук, доцент

РАЗРАБОТКА WEB-СЕРВИСА ДЛЯ ONLINE ЗАКАЗА БИЛЕТОВ

В современном мире человек каждый день нуждается в приобретении различного рода билетов: на городской, региональный и международный транспорт; на поезда и самолеты; на выставки и концерты, в театры и кино. Такие билеты семантически не связаны, однако, для пользователей будет удобно видеть и заказывать их все в одном месте.

В таком случае необходима некоторая иерархия, где все билеты будут унаследованы от какого-то абстрактного билета с минимальным набором свойств: цена, дата и время. Для создания подобной иерархии, а также ее хранения хорошим и современным подходом является использование технологии RDF (Resource Description Framework) – разработанная консорциумом Всемирной паутины (W3C) модель для представления данных. Ресурсом в RDF может быть любая сущность – как информационная, так и неинформационная. Утверждение, высказываемое о ресурсе, имеет вид «субъект–предикат–объект» и называется триплетом.

Разработан RESTweb-сервис на языке программирования Java, который предоставляет пользователю информацию о различных видах билетов, с возможностью их заказа.

RDF модель (словарь) данных разработана при помощи RDFS-схемы и языка OWL (The Ontology Web Language) и загружена в хранилище. В качестве хранилища была выбрана OpenRDFSesame, позволяющая работать с данными в таких форматах, как JSON и XML. Данные, добавляемые и хранящиеся в ней, являются триплетами. Запросы к такому хранилищу выполняются на языке запросов SPARQL (SPARQL Protocol and RDF Query Language). Связующим звеном между back-end частью

приложения и данными в RDF-формате была выбрана библиотека ApacheJena. С ее помощью устанавливается связь с хранилищем, а также выполняются SPARQL-запросы.

Разработанный REST web-сервис предоставляет клиентам CRUD-методы (Create, Read, Update, Delete) для работы с данными. Пользователь также имеет возможность совершать поиск по категории, времени события, а также по дополнительным параметрам в зависимости от выбранной категории. Для администраторов сервиса предусмотрена возможность загружать, изменять и удалять данные.

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

Науч. рук. С.А. Лукашевич, ст. преподаватель

ПРИМЕНЕНИЕ АЛГОРИТМА ВЫЧИСЛЕНИЯ ОПН ДЛЯ РАЗРАБОТКИ ПРИЛОЖЕНИЯ «ИНЖЕНЕРНЫЙ КАЛЬКУЛЯТОР» ПОД ОС ANDROID

Одной из главных причин, лежащих в основе появления языков программирования высокого уровня, явились вычислительные задачи, требующие больших объёмов рутинных вычислений. Поэтому к языкам программирования предъявлялись требования максимального приближения формы записи вычислений к естественному языку математики. В этой связи одной из первых областей системного программирования сформировалось исследование способов выражений. Наибольшее распространение получил метод трансляции с помощью обратной польской записи, которую предложил польский математик Я. Лукашевич. Обратная польская нотация была разработана австралийским философом и специалистом в области теории вычислительных машин Чарльзом Хэмблином в середине 1950-х на основе польской нотации, которая была предложена в 1920 году польским математиком Яном Лукашевичем.

Обратная польская нотация (ОПН) – форма записи математических выражений, в которой операнды расположены перед знаками операций. Инфиксная нотация – это форма математических записей, которую использует большинство людей (например, $3 + 4$ или $3 + 4 * (2 - 1)$). Чтобы дать индуктивное определение постфиксной нотации, обозначим выражения в инфиксной нотации E, E_1, E_2 , эквивалентные имвыражения в постфиксной нотации $\dot{E}, \dot{E}_1, \dot{E}_2$ соответственно; o – произвольный бинарный оператор, тогда:

1. Если E – переменная или константа, то \dot{E} есть E .
2. Если E – выражение вида $\dot{E}_1 o \dot{E}_2$, то \dot{E} есть $\dot{E}_1 \dot{E}_2 o$
3. Если E – выражение вида (E_1) , то \dot{E} есть E_1 .

Отличительной особенностью обратной польской нотации является то, что все аргументы (или операнды) расположены перед знаком операции. В общем виде запись выглядит следующим образом:

- запись набора операций состоит из последовательности операндов и знаков операций;
- выражение читается слева направо. Когда в выражении встречается знак операции, выполняется соответствующая операция над двумя последними встретившимися перед ним операндами в порядке их записи. Результат операции заменяет в выражении последовательность её операндов и её знак, после чего выражение вычисляется дальше по тому же правилу;
- результатом вычисления выражения становится результат последней вычисленной операции.

Например, рассмотрим вычисление выражения $7\ 2\ 3\ *$ – (эквивалентное выражение в инфиксной нотации: $7-2*3$).

Первый по порядку знак операции – «*», поэтому первой выполняется операция умножения над операндами 2 и 3 (они стоят последними перед знаком). Выражение при этом преобразуется к виду $7\ 6$ – (результат умножения – 6, – заменяет тройку «2 3 *»).

Второй знак операции – «-». Выполняется операция вычитания над операндами 7 и 6. Вычисление закончено. Результат последней операции равен 1, это и есть результат вычисления выражения. Эдсгер Дейкстра изобрёл алгоритм для преобразования выражений из инфиксной нотации в ОПН. Алгоритм получил название «сортировочная станция», за сходство его операций с происходящим на железнодорожных сортировочных станциях. Как и алгоритм вычисления ОПН, алгоритм сортировочной станции основан на стеке. В преобразовании участвуют две текстовых переменных: входная и выходная строки. В процессе преобразования используется стек, хранящий ещё не добавленные к выходной строке операторы. Преобразующая программа читает входную строку последовательно символ за символом (символ – это не обязательно буква), выполняет на каждом шаге некоторые действия в зависимости от того, какой символ был прочитан,

Таким образом, алгоритм вычисления ОПН и алгоритм «сортировочная станция» реализованные в коде на языке программирования Java, с помощью среды разработки Eclipse, позволяют создать калькулятор для устройств с ОС Android. Логика алгоритма вычисления ОПН позволяет использовать множество других операций над числами, помимо элементарных арифметических операций. В калькуляторе реализована возможность вычисления тригонометрических функций, натурального и десятичного логарифма, возведение в степень, извлечение из-под корня и др.

Литература

1. Пратт, Т. Языки программирования: разработка и реализация / Т. Пратт, М. Зелковиц. – 4-е изд. – Питер, 2002. – 688 с.

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

Науч. рук. **В. Д. Левчук**, канд. тех. наук, доцент

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ СЕРВИСА УЧЁТА ЗАКАЗОВ В ИТ-КОМПАНИИ

Заказы сотрудниками ИТ-компаний продуктов и офисного оборудования – один из ежедневных и рутинных процессов в процессе жизнедеятельности компаний. Как правило, такие заказы выполняются при помощи не предназначенных для этого средств – например, электронной почты. Такой подход имеет ряд недостатков: отсутствие систематизированности и централизованности, неадаптированность под задачи, неудобство пользования.

Внедрение специального сервиса для автоматизации обработки заказов позволяет решить все эти проблемы, ускорить время обработки и выполнения действий в рамках заказов и стандартизировать процесс работы с заказами.

При разработке приложения были реализованы функции, необходимые для эффективного использования его сотрудниками:

- просмотр активных заказов и добавление новых с указанием требуемого количества нужного продукта или единиц оборудования;
- редактирование и отмена заказов с изменением заказанного количества продукта и с указанием причины редактирования или отмены;
- просмотр и добавление комментариев к каждому из заказов;
- просмотр истории по всем заказам;
- просмотр подробной истории по каждому из заказов с указанием выполненных действий, даты и времени, а также пользователей, инициировавших каждое из выполненных действий;
- управление заказами со стороны менеджера (подтверждение, задержка, отмена, перевод в статус «в процессе»);
- возможность указания доставки продуктов в офис без их предварительного заказа сотрудниками (дистрибуция);
- управление списком и свойствами продуктов и оборудования, доступных для заказа;
- управление правами пользователей и разграничение доступа к системе в зависимости от ролей;

- фильтрация данных по определённым характеристикам.

Перечисленные возможности успешно реализованы в проекте по разработке системы управления, учёта и автоматизации заказов сотрудников ИТ-компании.

Система представляет собой локальный сайт, реализованный при помощи серверных и клиентских технологий и доступный из сети предприятия. В приложении успешно реализованы все перечисленные функции.

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

Науч. рук. **В.Д. Левчук**, канд. тех. наук, доцент

АВТОМАТИЗАЦИЯ УЧЁТА ЗАКАЗОВ СОТРУДНИКОВ В ИТ-КОМПАНИИ

Задача проекта – создание и внедрение продукта, предназначенного для автоматизации учёта заказов сотрудниками в офисы продуктов питания и дополнительного оборудования и автоматизации работы офис-менеджеров по учёту и доставке заказанных продуктов. Продукт внедряется на предприятие «Эпам Системз». Практический результат проекта – веб-приложение, доступное из сети компании и включающее клиентскую и серверную части. Автоматизированы процессы размещения и обработки заявок на доставку продуктов и оборудования. В приложении реализованы функции, облегчающие процесс заказа и обработки:

- управление статусами и состояниями заказов;
- добавление комментариев к заказам;
- ведение и просмотр истории всех заказов и событий для каждого заказа в отдельности;
- управление списком продуктов/оборудования и редактирование их свойств;
- фильтрация продуктов и действий;
- управление правами доступа пользователей к сервису.
- Основные средства автоматизации – языки программирования Java (для серверной части) и JavaScript (для клиентской части).

Цели, достигаемые внедрением проекта:

- использование для создания и обработки заказов специально разработанного для этого инструментария, учитывающего особенности процессов в предметной области;
- увеличение скорости работы с заказами;

- стандартизация действий и процессов, характерных для предметной области;
- концентрация информации по всем заказам и процессам в рамках одного сервиса.

Обмен данными между серверной и клиентской частями реализован при помощи RESTful-интерфейса. При этом данные передаются в формате JSON по протоколу HTTP. На стороне сервера для создания REST-объектов для передачи используется стиль DTO (Data Transfer Object). На клиентской стороне данные обрабатываются при помощи языка JavaScript в реализованных на нём библиотеках – для языка JavaScript формат JSON является стандартным форматом для обмена данными [1].

В приложении реализована система разграничения пользовательских прав. Существует несколько ролей, представители каждой из которых отвечают за выполнение определённых действий и имеют соответствующий набор прав. На рисунке 1 показана диаграмма прецедентов, отражающая возможные последовательности процессов внутри приложения, инициируемых пользователями, имеющими определённые роли.

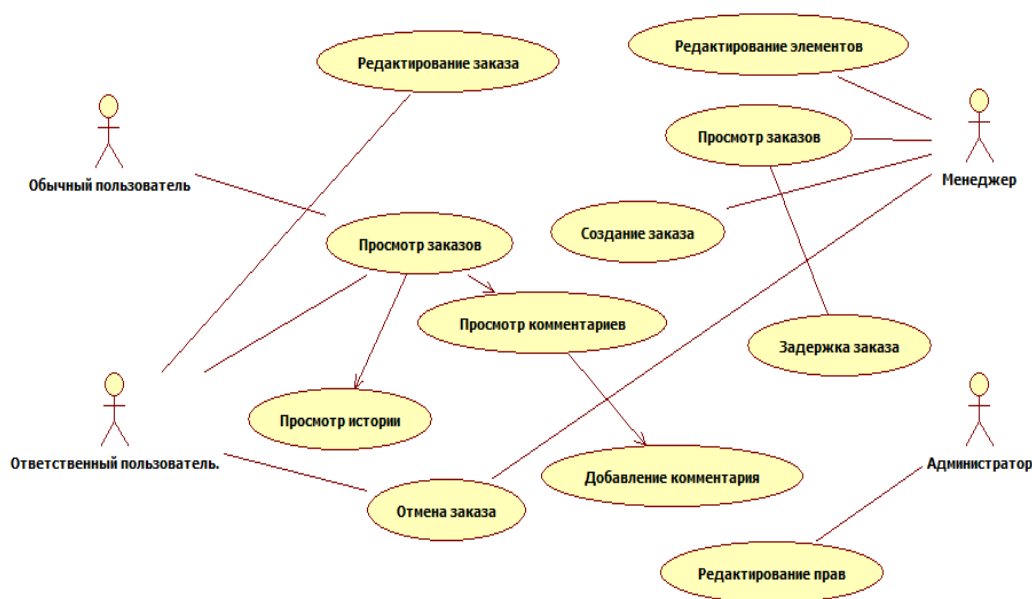


Рисунок 1 – Диаграмма прецедентов

Сервис реализован в стиле single page application (одностраничное приложение) [2]: для каждой из трёх ролей пользователей (обычный пользователь, менеджер и администратор) создано по странице, в рамках которой реализован весь функционал, требуемый для роли. Все действия, выполняемые в пределах функционала для роли, выполняются без перезагрузки страницы, обмен данными с сервером выполняется по технологии Ajax.

Серверная часть приложения реализована на языке Java с использованием фреймворков Spring и Hibernate. В приложении использованы такие техники, как Java Beans (классы, написанные по определённым правилам и используемые для объединения нескольких объектов в один для удобной передачи данных), контроллеры (классы, используемые для обеспечения работы веб-сервиса по принципу MVC), DAO (Data Access Object – объекты, предоставляющие абстрактный интерфейс к типам базы данных и механизмам хранения), DTO (Data Transfer Object – шаблон проектирования, используемый для передачи данных между подсистемами приложения и не содержащий какого-либо поведения).

Клиентская часть приложения написана на языке JavaScript с использованием библиотек jQuery, Underscore, Backbone.js. Благодаря их широким функциональным возможностям работы с данными и интерфейсом значительная часть функционала перенесена на сторону клиента и не использует серверные ресурсы.

В обобщённом виде схема обмена данными между клиентской и серверной частями выглядит следующим образом:

1 Клиент при помощи фреймворка Backbone.js выполняет REST-запрос на сервер. В запросе содержится действие, которое необходимо выполнить, и при необходимости – определённые данные.

2 Сервер при помощи Spring-контроллеров обрабатывает запрос, выполняет требуемые действия с данными, после чего при помощи Data Access Object подготавливает ответ и отправляет его на клиент.

3 Клиент принимает ответ от сервера. В зависимости от его содержания запускаются определённые действия на страницах приложения. Так, в случае необходимости отображения каких-либо данных они преобразуются в Backbone-коллекции предварительно описанных моделей, после чего с использованием Backbone-видов и Underscore-шаблонов отображаются на странице в требуемом виде.

Для установки сервиса пригоден любой хостинг, поддерживающий выполнение Java-приложений. Клиентская часть работает во всех современных браузерах – Mozilla Firefox, Opera, Google Chrome, Apple Safari, Microsoft Internet Explorer (версии 9 и старше), а также во всех других браузерах, основанных на движке WebKit.

В результате реализации проекта разработана и создана объектная модель, описывающая процессы и действия, проходящие при создании и обработке заказов сотрудников. Модель используется при разработке приложения и при необходимости может быть легко расширена и адаптирована к условиям. В ходе работы над проектом были применены возможности языков программирования Java и JavaScript (а также сторонних фреймворков и библиотек для этих языков), методы взаимодействия клиентской и серверной частей и обмена данными между ними.

М.М. Приходько (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **В.С. Давыдов**, канд. тех. наук, доцент

СЕРВИСЫ ДЛЯ ПОДБОРА КЛЮЧЕВЫХ СЛОВ ПРИ ПОИСКОВОЙ ОПТИМИЗАЦИИ

Основная цель обращения к подобным сервисам – узнать, сколько человек в месяц набрали определенную ключевую фразу. Например, необходимо добавить запрос «купить клубнику» и не известно, принесет ли он трафик. Обратившись к специальному сервису, можно узнать, сколько человек в месяц набирали запрос «купить клубнику».

Но прежде чем обратиться к сервису, необходимо выяснить, что такое частотность, и каких видов она бывает.

Частотой запроса называют количество поисковых запросов применительно к конкретным фразам или словам, заданным пользователями за определенное время – чаще всего за месяц.

Различают 3 вида частотности:

- базовая;
- точная;
- «сверхточная» (учитывает окончания).

Базовая частотность. Если ввести ключевую фразу в окно поиска, то будет получена базовая частотность. В колонке «показов в месяц» (находится на против каждого ключевого слова или фразы) находится цифра. Эта цифра показывает, сколько раз пользователи набрали в строке Яндекса указанную ключевую фразу (как отдельно, так и в составе других ключевых фраз). Если у фразы есть словоформа – она тоже учитывается. То есть «преподаватель» и «преподаватели» для Яндекса одно слово (в базовой частотности).

Точная частотность. Точная частотность показывает, сколько раз пользователи набрали в строке Яндекса данную ключевую фразу и её словоформы. Ключевую фразу нужно вводить в двойных кавычках. («купить клубнику»)

Сверхточная частотность. Данная частотность показывает, сколько раз пользователи набрали в строке Яндекса заданную ключевую фразу и не учитывает её словоформы. Ключевую фразу нужно вводить «Двойная кавычка, восклицательный знак, ключевая фраза, двойная кавычка» («!купить клубнику»).

Литература

1. Яндекс подбор слов [Электронный ресурс]. – 2014. – Режим доступа: <http://wordstat.yandex.ru>. – Дата доступа: 10.02.2014.

М.М. Приходько (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **В.С. Давыдов**, канд. тех. наук, доцент

АНАЛИЗ ВАЖНЕЙШИХ ЭЛЕМЕНТОВ САЙТА, ПРИВЛЕКАЮЩИХ ПОЛЬЗОВАТЕЛЕЙ И ПОИСКОВЫЕ СИСТЕМЫ

На каждом сайте есть факторы, которые одновременно оказывают влияние на продажи и продвижение в поисковых системах. Каждый из перечисленных пунктов имеет самое непосредственное отношение к оптимизации сайта, хотя этого и не видно при поверхностном взгляде. Со всем недавно аксиомой была следующая логика: лучше SEO – больше продаж. Сейчас логика работает и в обратном направлении: больше продаж – лучше SEO. Ниже будут перечислены некоторые факторы с описанием, почему они важны.

Отзывы. Без знакомства с отзывами сейчас не совершается ни одна более-менее серьезная покупка: от фена до квартиры. Обычно отзывы читают в тех случаях, когда покупка носит единоразовый, а не повторяющийся характер. Чаще встретишь отзывы о турах или шубах, чем о молоке или бензине. Отзывы покупателей могут относиться как к отдельным товарам и услугам, так и к компании в целом. Подавляющее большинство компаний найдет повод разместить отзывы на своем сайте.

Цены. Очевидный вывод, что цена обязательно должна присутствовать на продающих сайтах. Однако некоторые компании совершают ошибку, скрывая цены с помощью скриптов («Чтобы узнать цену, кликните по кнопке») или выдавая их только по запросу.

Существует реальный страх показать цену. Он не имеет никакого смысла, потому, что, выдав цену после продолжительного перехода по ссылкам, она не будет выглядеть более привлекательно. Есть только два способа ценообразования: конкурентоспособный и неконкурентоспособный. Цена может быть высокой и конкурентоспособной, если за ней стоит реальная ценность. Прятать цену – значит намекнуть клиенту, что она высокая и неконкурентоспособная, потому что в остальных случаях прятать нечего. На сайте интернет-магазина или сервиса информация о цене должна быть на виду.

Портфолио и клиенты. Другой способ повысить доверие к сайту – это разместить на нем образцы выполненных проектов и информацию о клиентах. Данный фактор также повышает уровень доверия и продлевает время пребывания на сайте. Все проекты и всех клиентов перечислить невозможно. Поэтому рекомендуется привести статистику: количество выполненных работ по направлениям, общее количество клиентов (если возможно, то и по отраслям).

Акции и специальные предложения. Желательно выделять акции и спецпредложения в отдельный раздел сайта. И очень не рекомендуется заниматься «креативом», придумывая для него оригинальное название. «Это интересно» – не самое удачное название для страницы с акциями. За таким заголовком может скрываться что угодно. Посетители сайта хотят простоты и ясности, а поисковые системы хотят видеть раздел «Акции». Для целей продаж желательно выводить главные специальные предложения на домашнюю страницу в виде баннера.

Раздел «О компании» – один из главных разделов любого коммерческого сайта. Главная ошибка большинства компаний состоит не в отсутствии раздела как такового, а в поверхностной и неубедительной информации, которая в нем содержится. Данный раздел должен содержать следующую информацию:

- 1 история компании;
- 2 специализация;
- 3 особенности и преимущества, чтобы подчеркнуть отличия от конкурентов;
- 4 данные о руководстве и ключевых экспертах, что особенно важно для компаний, оказывающих услуги;
- 5 новости;
- 6 контактные данные пресс-службы;
- 7 фотографии руководства;
- 8 фотографии с мероприятий;
- 9 промо-материалы: буклеты, ролики и фотосессии.

Пункты 5, 8 и 9 позволяют регулярно обновлять информацию в разделе, что благотворно влияет на продвижение сайта.

Форма обратной связи или кнопка «Заказать звонок». Отдельные виды бизнеса предполагают, что клиент долго принимает решение, изучая информацию и консультируясь с продавцами. Классический пример такого бизнеса – установка пластиковых окон. В этом случае клиенту приходится звонить в несколько компаний, что довольно утомительно. Форма обратной связи или кнопка «Заказать звонок» снимают с него часть хлопот. Кроме того, компания получает контактные данные потенциального клиента для последующей связи и повторных продаж.

Онлайн-консультант – это программа, которая позволяет консультировать клиента без звонков в офис и писем менеджерам. Моментальное консультирование дает возможность сразу построить контакт с потенциальным клиентом и улучшает поведенческий фактор. Сервис важен для каждого бизнеса, для которого характерно длительное принятие решения о покупке.

Новости. Выше говорилось о полезности новостей в разделе

«О компании». Однако совершенно необязательно размещать их именно там. Можно создать отдельный раздел с новостями или публиковать их в блоге компании. Регулярное обновление контента полезно для продвижения сайта. Кроме того, с помощью новостей можно анонсировать акции и новые поступления, рассылая их по почтовым ящикам клиентов. Как получить e-mail адреса клиентов? Например, с помощью формы обратной связи. На сайте все взаимосвязано. В разделе «О компании» можно публиковать деловые новости для партнеров, а информацию, предназначенную для конечного потребителя, размещать в других разделах.

Юридические данные компании. Элемент, полезный для интернет-магазинов. Повышает доверие к компании, показывая, что сайт зарегистрирован на реальную фирму и работает с соблюдением законодательства. Небольшим интернет-магазинам, которые едва вышли на рынок, настоятельно рекомендуется внедрить данный элемент.

Фотографии и видео. Интернет превращается в визуальную среду, что означает меньше текстовой и больше визуальной информации. Не рекомендуется отказываться от текстов, но желательно разбавлять их фотографиями и видео. Они могут быть размещены в самостоятельных разделах, а также иллюстрировать статьи и новости.

Р.О. Розинко (УО «ГГУ им. Ф. Скорины», Гомель)

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

АВТОМАТИЗАЦИЯ РАСЧЕТА ЗАРАБОТНОЙ ПЛАТЫ СОТРУДНИКОВ КЖУП «ГОМЕЛЬСКИЙ РАЙЖИЛКОМХОЗ»

В нынешнее время работа коммунальных жилищных предприятий осложнена достаточно большим количеством проблем, которые можно автоматизировать при помощи современных компьютеров и техники, а так же с помощью новых разработок в сфере языков программирования высокого уровня. На составление графиков расписания и просчет заработной платы затрачиваются большие временные и трудовые затраты, которые можно использовать в других направлениях, а данную работу производить на ЭВМ при помощи алгоритмов. Однако даже сейчас данная работа выполняется вручную.

Для реализации создания графиков смен необходим достаточно простой алгоритм, который бы на основании данных о сотрудниках, их количестве, а так же данных о календарном месяце автоматически просчитывал график и предоставлял документ для печати. Аналогично этому требуется реализовать алгоритм, который бы на основании графика смен просчитывал заработную плату для сотрудников. Данное

программное обеспечение значительно упростит и ускорит работу мастеров и бухгалтеров на предприятии.

Не смотря на то, что в современное время существует огромный выбор аппаратных средств (мощные ПК, которые могут использоваться как сервера баз данных по учету всего необходимого в работе КЖУП), и программных сред (большое количество разнообразных языков программирования), тем не менее полноценного функционального программного обеспечения для составления графиков работы сотрудников и для вычисления заработной платы в КЖУП введено не было, хотя это значительно упростило бы часть повседневной работы.

В результате выполнения дипломного проекта было написано ПО по автоматизации составления графиков смен сотрудников с дополнительным просчетом заработной платы. Программное обеспечение соответствует задаче дипломного проекта и написано на языке программирования PHP с применением базы данных MySQL. Доступ осуществляется через простой и понятный веб-интерфейс посредством любого браузера.

Для реализации программного обеспечения по автоматизации был выбран язык программирования PHP с использованием базы данных MySQL. Для работы скриптов установлен веб-сервер Apache вместе с модулем PHP последней версии. Так же установлен сервер БД MySQL.

Так как система построена на модульной структуре, то дописывание системы для дополнительных новых задач не составит никакого труда, а так же модульная система позволяет разграничить ПО на части для обновления существующих функциональных возможностей в будущем. Такая структура в последнее время становится все более популярной и повсеместно-внедряемой из-за своей простоты разработки, надежности работы, а так же простой возможности для обновления либо замены целых частей системы.

Интерфейс ПО понятный и легкий, с ним легко освоится даже человек с небольшим опытом работы за компьютером. Доступ к программному обеспечению осуществляется через любой браузер, который установлен у пользователя, что так же удобно ввиду отсутствия дополнительных программ на стороне клиента, с которыми нередко возникают проблемы. Каждая функция подписана, некоторые снабжены информативным значком, что положительно сказывается на простоте использования системы.

Исходя из всего перечисленного можно сделать вывод, что разработанное программное обеспечение по автоматизации расчета заработной платы сотрудников КЖУП действительно является системой, которая способна грамотно составить график смен, просчитать заработную плату с учетом всех отчислений и, главное, способна значительно сократить требуемые временные и трудовые затраты на составление этих данных вручную. Данное ПО может быть применено не только на объектах КЖУП, но и на схожих предприятиях с аналогичными графиками смен.

П.А. Руденя (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. Е.Е. Пугачева, ассистент

ИСПОЛЬЗОВАНИЕ АРХИТЕКТУРЫ «МОДЕЛЬ-ПРЕДСТАВЛЕНИЕ» ПРИ РАЗРАБОТКЕ DESKTOP-ПРИЛОЖЕНИЙ

Архитектура модель-представление-контроллер (Model-View-Controller, MVC) является шаблоном проектирования. MVC состоит из трех типов объектов. Модель – объект приложения, представление – его экранное представление и контроллер – определяет реакцию пользовательского интерфейса на пользовательский ввод. До MVC при разработке пользовательского интерфейса эти объекты смешивались вместе. MVC разделяет их, для увеличения гибкости и возможности повторного использования.

Если объединить объекты представления и контроллера, то в результате получится архитектура модель/представление. Это все еще отделяет способ хранения данных от способа их представления пользователю, но обеспечивает простую структуру, основанную на тех же принципах (рисунок 1).

Данное разделение дает возможность показать пользователю одни и те же данных в различных представлениях и реализовать новые типы представлений без изменения базовой структуры данных. Чтобы обеспечить гибкость управления пользовательским вводом, qt предоставляет концепцию делегата(delegate). Преимущество наличия делегата в этой структуре состоит в том, что это дает возможность для настройки представления и редактирования элементов данных.

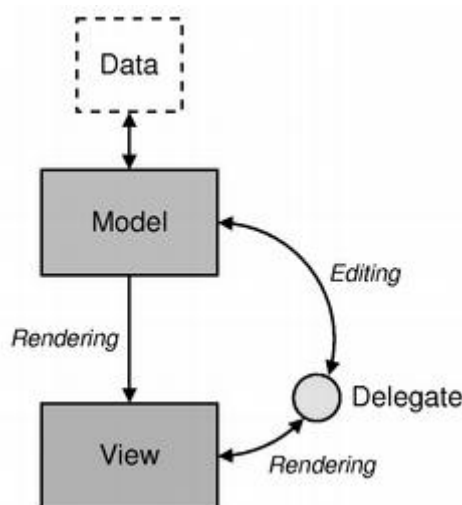


Рисунок 1 – Архитектура модель-представление

Применение архитектуры «модель-представление» дает такие преимущества как:

– возможность показать данные в нескольких представлениях без дублирования. При работе с элементарно-базированным подходом при добавлении новых элементов происходит дублирование данных. В подходе «модель-представление» связываются напрямую с моделью данных, поэтому дублирование не происходит;

– возможность внесения изменений с минимумом временных затрат. При изменении способа сохранения данных не придется вносить больших изменений в код программы, так как связь происходит с помощью интерфейса;

– удобство программного кода. Поскольку осуществляется разделение на данные и представление, то, если появится необходимость что-то дополнить или исправить, то изменения коснутся лишь одной из частей. Остальные части приложения останутся нетронутыми;

– упрощение интеграции баз данных.

Литература

1. Шлее, М. Профессиональное программирование на C++. Qt 4.8 / М. Шлее. – М.: ООО «БХВ-Петербург», 2011. – 874 с.

2. Саммерфилд, М. Qt профессиональное программирование / М. Саммерфилд. – М.: Символ-плюс, 2011. – 765 с.

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

Науч. рук. **А.В. Воруев**, канд. тех. наук, доцент

МЕХАНИЗМЫ ОБЕСПЕЧЕНИЯ СОВМЕСТИМОСТИ ПРОТОКОЛОВ IPv4 И IPv6

Уже на этапе разработки новой версии протокола IP было ясно, что переход от IPv4 к IPv6 не будет быстрым. Стало очевидным, что две версии протокола будут долгое время сосуществовать. Таким образом, вопрос совместимости IPv6 с IPv4 был и остаётся актуальным.

В общем случае, выделяют две схемы взаимодействия сетей разных протоколов:

1 Работа IPv6 сетей, построенных поверх существующих IPv4 сетей. В этом случае для взаимодействия хостов IPv6 используется среда передачи данных существующей сети IPv4.

2 Взаимодействие непосредственно хостов IPv6 и IPv4.

На сегодняшний день существует четыре основных механизма обеспечения как прямой, так и обратной совместимости протоколов:

- двойной стек;
- шлюз прикладного уровня (ALG – application level gateway);
- туннелирование;
- транслятор IP/ICMP.

Ниже представлено описание каждого из этих механизмов.

Двойной стек является наиболее простым механизмом обеспечения совместимости между IPv4 и IPv6.

На каждом узле сети, который работает с IPv6 и которому требуется взаимодействие с IPv4-хостами, устанавливается стек протокола IPv4, и ему выделяется IPv4-адрес. После этого этот узел может взаимодействовать как с IPv4 хостами, так и с IPv6 хостами. Главным преимуществом принципа двойного стека является его относительная простота. Однако у метода множество недостатков.

Во-первых, для того чтобы механизм нормально работал, необходимо, чтобы большинство промежуточных маршрутизаторов работали как с протоколом IPv4, так и с протоколом IPv6. Во-вторых, для использования двойного стека необходимо установить и настроить на каждом узле специальное программное обеспечение. А это требует дополнительных затрат как времени специалистов, так и денег на приобретение или разработку ПО. В-третьих, применение механизма двойного стека повышает использование системных ресурсов узлов Сети и может замедлять их работу. Две последние проблемы могут решить производители сетевого оборудования и программного обеспечения, добавив поддержку IPv6 в свои продукты.

Application Level Gateway (ALG) – шлюз прикладного уровня. Данный механизм предполагает, что для каждого сетевого приложения создается специальная утилита, которая преобразовывает весь входящий трафик этого приложения из IPv4 в IPv6, а весь исходящий – из IPv6 в IPv4. Т.е., сам узел работает по новой версии протокола, а весь трафик передается по старой. Таким образом, использование шлюзов обеспечивает как взаимодействие узлов, работающих с разными протоколами, так и IPv6-сетей между собой посредством существующей среды передачи данных IPv4. Преимуществами использования данного метода являются простота установки и настройки, а во-вторых – универсальность механизма.

Однако стоит отметить, что использование механизма шлюзов прикладного уровня является достаточно неэффективным. Это обусловлено тем, что необходима установка дополнительного программного обеспечения для каждого приложения. Что влечёт, в свою очередь, разработку таких программ и последующую установку их пользователем. С учётом того, что сегодня пользователями устанавливаются десятки программ,

которые используют Интернет, задача становится очень трудоёмкой и затратной.

Механизм туннелирования используется для создания IPv6 туннелей сквозь существующие IPv4 сети (например, Интернет), не поддерживающие протокол IPv6. Он не может применяться для связи IPv6-узлов с IPv4-хостами. Такие туннели создаются вручную либо автоматически различными способами и объединяют отдельные IPv6 сети между собой. Пакеты IPv6, входя в такой туннель, инкапсулируются в пакеты IPv4 и пересылаются по IPv4 сети на другой конец туннеля. Там они деинкапсулируются и обрабатываются далее маршрутизаторами как обычные IPv6 пакеты.

Главным преимуществом этого механизма является отсутствие необходимости приобретать и устанавливать дополнительное программное обеспечение на каждом узле. Так, для сети, построенной на основе протокола IPv6, достаточно создать несколько "туннелей", связывающих ее с другими такими же сетями.

Однако данное решение проблемы совместимости является частичным, т.к. оно обеспечивает создание наложенных IPv6 сетей поверх существующей сетевой инфраструктуры. Оно не обеспечивает взаимодействия IPv4 хостов с IPv6 хостами.

На сегодняшний день механизм туннелирования является наиболее распространённым как среди пользователей, так и среди разработчиков. Одной из причин этому послужило то, что на основе этого механизма была построена экспериментальная сеть `bone` – полигон для тестирования IPv6.

Транслятор IP/ICMP является самым современным механизмом и обеспечивает как совместимость между IPv4 и IPv6-узлами, так и связь между собой IPv6-сетей посредством среды передачи данных IPv4.

В нем используются специальные типы адресов, которые есть в новой версии протокола. Для реализации этого механизма необходимо специальное программное обеспечение – транслятор, который устанавливается на границе IPv6-сети. Именно транслятор принимает трафик в формате IPv4, преобразовывает его в соответствии со стандартом IPv6 и отправляет узлам сети, которым уже присвоен IPv4-транслированный адрес. Исходящий трафик, соответственно, преобразовывается согласно протоколу IPv4 и затем отправляется адресатам.

Транслятор просто устанавливается, настраивается и не вызывает трудностей в использовании. Над используемым в сети программным обеспечением не нужно производить никаких преобразований. Единственной задачей системного администратора становится присвоение всем хостам, входящим в сеть, IPv4-транслированных адресов.

Стоит отметить, что при всех своих плюсах, транслятор имеет и минусы, самым существенным из которых является его «односторонность». Транслятор может использоваться только для связи сетей IPv6 через пространство IPv4, но не наоборот. Таким образом, транслятор наиболее актуален только на начальном этапе перехода на новый протокол, когда пространство IPv4 намного превосходит пространство IPv6. В будущем, при постепенном разрастании последнего, обратная задача – обеспечение связи IPv4-сетей через пространство IPv6 – будет невозможна при помощи транслятора.

Литература

1. Механизмы совместимости IPv4 и IPv6 / <http://hostinfo.ru> – справочная информация и практические советы, 2013. – В режиме доступа: <http://hostinfo.ru/articles/386>. – Дата доступа: 10.2.2014.

2. Шнитман, В.З. Обеспечение совместимости протоколов IPv4 и IPv6: бесконтекстный IP/ICMP транслятор в среде Linux / В.З. Шнитман, А.А. Ломака. – Институт Системного Программирования Российской Академии Наук, 1999. – В режиме доступа: <http://ipv6.ispras.ru/pubs.htm>. – Дата доступа: 10.2.2014.

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

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

РАЗРАБОТКА ПРОЕКТА МОДЕРНИЗАЦИИ ЛВС ДЛЯ ГОМЕЛЬСКОГО ИНТЕРНЕТ-ПРОВАЙДЕРА «ГАРАНТ»

Известно, что каждый Интернет-провайдер, либо другая любая компания желает занять возможную большую нишу, как для информационного охвата, так и для собственной коммерческой выгоды. Расширение информационного охвата сосредоточивает возможности по управлению в одних руках. Так провайдер, охватывая все новые рынки предоставления мультимедийных услуг, диктуя правила в этом регионе при условии заведомо большего потенциала, вытесняет других более мелких и менее развитых региональных провайдеров, монополизируя рынок. Соответственно, это ведет к увеличению прибыли компании монополиста, и ОАО «Гомельский ОТТЦ «Гарант» не стал в этом исключением. Поставленная цель разработки проекта по расширению сети в новый регион охвата в г. Речице и объединения локальных вычислительных сетей в одну региональную, позволяет гомельскому провайдеру решить вопрос информационного охвата и монополизировать рынок мультимедийных услуг в регионе Гомельской области.

Известно, что при вводе любой ЛВС в эксплуатацию она замкнута на определенный план действий, который условно можно назвать циклом. Цикл – это жизненный путь ЛВС. Жизненный цикл можно разделить на шесть этапов: разработка проекта, монтаж, тестирование, обслуживание, модернизация, демонтаж. Каждый из этапов подразумевает выполнение определенных действий, входящих в общий план. Весь план может быть выполнен только при строгом выполнении всех пунктов следующих друг за другом этапов.

При разработке проекта ЛВС для филиала ОАО «Гомельский ОТТЦ «Гарант» в г. Речица использовался нестандартный подход к проектированию локальных вычислительных сетей для предприятий данной структуры. Нестандартность заключалась в том, что потребовалось не просто спроектировать локальную вычислительную сеть филиала компании, но и объединить ее с опорной сетью главного офиса, находящегося в другом городе. Ответственным шагом было выработать оптимальное решение по прокладке волоконно-оптической линии связи для объединения двух ЛВС.

В ходе начальных исследований был проведен полный анализ сети главного офиса компании. Выявлены возможности синхронизации и сопряжения с будущей локальной сетью филиала. Определены принципы и алгоритмы проектирования локальных вычислительных сетей филиалов на основе анализа структуры компании, финансовых возможностей, доступных сетевых технологий, существующего сетевого оборудования, метода размещения сетевого оборудования и модели функционирования сети в имитационном графическом редакторе Cisco Packet Tracer (рисунок 1).

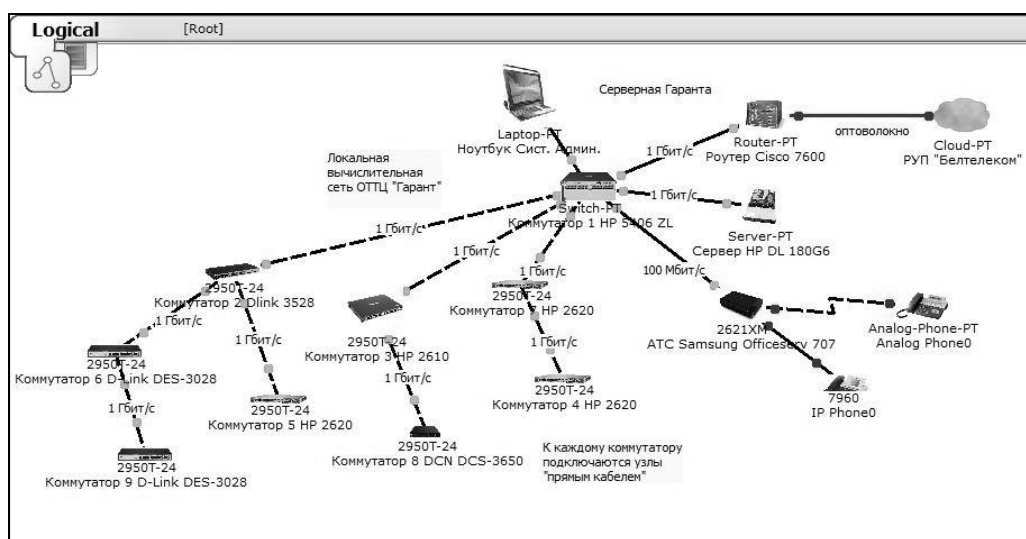


Рисунок 1 – Модель ЛВС в графическом редакторе Cisco Packet Tracer

В результате проведенного анализа были выделены следующие этапы проектирования локальной вычислительной сети филиала компании:

- мониторинг рынка современного активного и пассивного сетевого оборудования;
- анализ сетевого комплекса решений по прокладке магистрали ЛВС Гомель – Речица;
- проектное обследование здания филиала и сценарий размещения сетевого оборудования;
- анализ структуры управления филиалом и определение количества подключений абонентов к ЛВС;
- анализ оборудования для монтажных работ при прокладке коммуникаций ЛВС филиала;
- изучение требуемых санитарно-эпидемиологических норм, правил техники безопасности, ГОСТов, стандартов и других нормативно-правовых актов, соблюдаемых при монтаже локальных вычислительных сетей;
- монтаж структурированной кабельной системы;
- настройка конфигурации оборудования для стабильной работы всех компонентов ЛВС;
- настройка политик управления доступом к сети филиала компании.

В итоге разработан проект, который условно разделен на две части. Первая часть – проект прокладки оптоволоконной линии связи от главной станции в г. Гомеле до принимающей в г. Речице с пропускной способностью 1 и 10Гб/с на одно волокно и использованием сетевых технологий вставок Gigabit-LX-LC Mini-GBIC, осуществляющих преобразование электрического сигнала 100/1000BaseT в оптический и обратно. Вторая часть – проект создания самой ЛВС филиала. Разработанный проект успешно внедрен в филиале компании ОАО «Гомельский ОТТЦ «Гарант» в г. Речица.

А.А. Самусев (БТЭУ ПК, Гомель)

Науч. рук. **С.В. Карпенко**, канд. эконом. наук, доцент

РАЗРАБОТКА САЙТА ДЛЯ ИНТЕРНЕТ-ТОРГОВЛИ АВТОЗАПЧАСТЯМИ

За прошлый год товарооборот интернет-магазинов автозапчастей составил 231,6 млрд. рублей, тогда как в 2011-м – 69,9 млрд. рублей.

Как отметила зам. министра торговли Республики Беларусь Наркевич И.В. [1], плюсом интернет-торговли является более низкая цена

товаров, чем в розничных магазинах за счет экономии средств на аренде торговых площадей и зарплате продавцов и возможности совместной аренды складских помещений. Наркевич считает, что «данная сфера деятельности будет развиваться активными темпами».

На начало апреля 2012 г. в Беларуси зарегистрировано более 370 интернет-магазинов автозапчастей. Специалист напомнила, что в соответствии с законодательством интернет-магазины, реализующие товары в Беларуси, должны быть зарегистрированы в Торговом реестре, который размещен на сайте Минторга. Там же есть вся информация о нормативных актах, регулирующих деятельность интернет-магазинов, и бланк соответствующего заявления на регистрацию. Данный вид деятельности не лицензируется, а регистрация производится в течение одного дня с момента подачи заявления.

Потребители в целях защиты своих прав сами должны контролировать покупку через интернет, требовать документы об оплате и договоры о сервисном обслуживании, а также читать всю информацию о продавце.

Потенциальные покупатели автомобиля перед определением марки и модели желаемого автомобиля, почти всегда задаются вопросом доступности запасных частей, как по наличию в авто магазинах, так и по цене. Определимся с маркой автомобиля, возьмём к примеру Форд (Ford). Далее мы определимся, какие модели Форд (Ford) обычно покупают белорусы. Самой популярной моделью Форд (Ford) в Республике Беларусь является Фокус (Focus)[2], после него идут Transit и Mondeo, далее Orion и KA, C-Max и Galaxy, все больше появляются Explorer, все меньше остается Cougar.

Хочется отметить надежность автомобилей Ford, но какой бы ни был сверхнадежный автомобиль, автозапчасти все равно рано или поздно понадобятся. Чаще всего меняют в автомобиле «расходники». Выполним сравнительный анализ цен на качественные «расходники».

Для примера выберем среднюю по популярности модель: Ford-Mondeo v161.8 л. 115 л.с. Что будем менять: ремень ГРМ с роликом, фильтр воздушный, топливный и масляный.

Проведем сравнительный анализ цен и условий по интернет магазинам автозапчастей с электронными каталогами. Запчасти выбирались самые дешевые (цена зависит от срока доставки) в бел. руб. на 29.12.2013.

Посмотрим и проанализируем отзывы о данных интернет-магазинах.

Exist.by. Как можно увидеть по таблице, цены здесь самые приемлемые, а самое привлекательное – это очень большой выбор аналогов по различной цене. Цена тем ниже, чем больше срок доставки (за срочность нужно платить). Несомненным преимуществом является

иллюстрированный каталог, понятный обычному пользователю, а также развитая сеть офисов: один офис в Гомеле и офисы в областных городах. Большим недостатком является необходимость предоплаты (50%), приходится ездить два раза: сначала для внесения предоплаты, а потом для того чтобы забрать запчасти. Этот недостаток снимается, если положить на свой счет в системе некоторую сумму денег и пополнять её, когда забираете запчасти. Еще одним недостатком являются очереди: в после рабочее время скорее всего придется постоять в очереди, поэтому лучше приезжать днем.

Таблица 1– Сравнительный анализ цен на автозапчасти в интернет магазинах Гомеля в бел. руб. на 29.12.2013

	Цена, бел.руб. (на 29.12.2013)				
	Exist.by	Atr.by	Avtocat.com	Braz.by	Interparts.by
<i>Ремень ГРМ Contitech СТ728</i>	121000	143000	142000	127700	154200
<i>Ролик натяжителя Stellox 03-40021-SX</i>	101000	124000	96000	92400	137500
<i>Фильтрвоздушный Bosch 1 457 433 609</i>	97600	106000	93000	98400	102200
Фильтр топливный Champion L218/606	70400	85000	70500	76200	143300
Фильтр масляный Champion C150/606	40800	54000	39200	44200	53900

Atr.by. Ассортимент достаточно велик, всегда есть выбор из аналогов. Цена (по опыту) иногда даже меньше чем на Exist.by. Неоспоримыми преимуществами являются наличие склада (возможность забрать запчасти в день заказа) и отсутствие предоплаты.

Avtocat.com. Сайт предоставляет список магазинов с контактными данными и ценами, у которых есть необходимая запчасть, а также выдает список аналогов.

Braz.by. Недавно появился новый ресурс. Поиск на нем всегда адекватный, например, вместо аналогов на фильтр MEAT & DORIA 18043 выдает список пыльников, а шины сортируются по типоразмеру, а не по радиусу. Преимущества: доставка в тот же день, при заказе на сумму от 90 тыс. бел. Руб. доставка бесплатная, экспресс-доставка, есть накопительные скидки.

Interparts.by. Сайт по своей структуре похож на Exist, схема поиска, выбора и покупки запчастей аналогичная. Цена на некоторые позиции заметно ниже, чем у Exist.

Таким образом, данная область интернет-торговли является актуальной и весьма прибыльной на сегодняшний день. Анализ позволяет сделать вывод, что лучшим интернет-магазином в г. Гомеле является магазин Exist.by: здесь хорошо налажена поставка запчастей в кратчайшие сроки, имеется квалифицированный персонал, который поможет с выбором автозапчасти за приемлемые деньги. В офисе компании, куда поставляются заказы, есть клиентский терминал (клиентский ПК) для автономной работы покупателя на сайте.

Представленная мною модель отражает налаженную структуру интернет сайта, готовую заниматься интернет торговлей (рисунок 1).

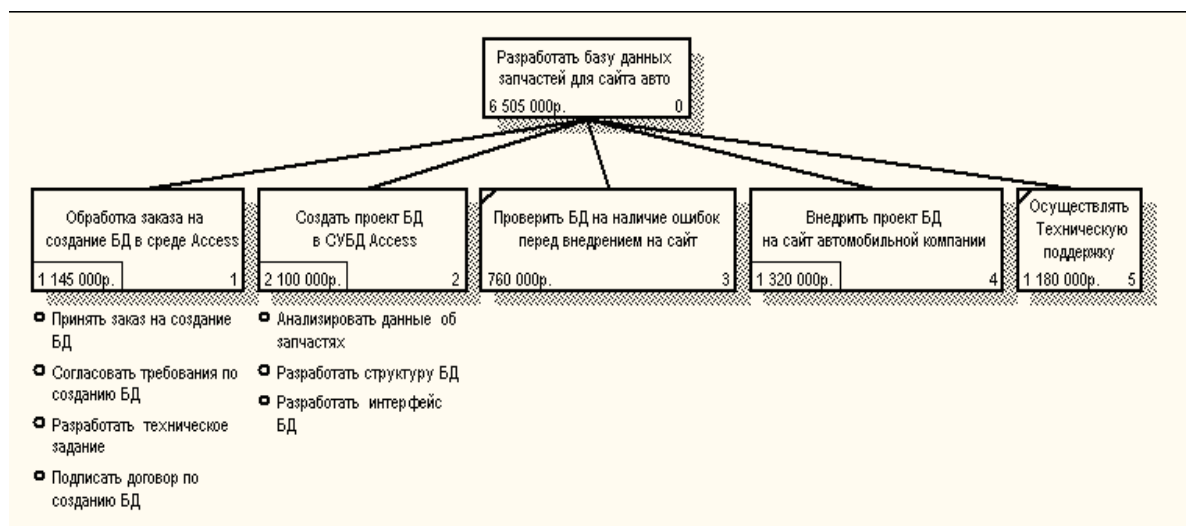


Рисунок 1– Дерево сайта интернет магазина

В качестве основы для разработанного мной интернет сайта взята структура сайта Exist написанный HTML-кодом, размещенном на сервере Server: Microsoft-IIS/6.0. Такой сервер может обслуживать сразу несколько сайтов.

А.В. Сапанович (УО «ГГТУ им. П.О. Сухого», Гомель)
 Науч. рук. **Е.Г. Стародубцев**, канд. физ.-мат. наук, доцент

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ МОБИЛЬНЫХ ПЛАТЕЖНЫХ ПРИЛОЖЕНИЙ

На текущий момент электронная коммерция стала неотъемлемой частью мировой финансовой системы. Среди наиболее трудоемких операций при тестировании данных систем выделяется тестирование

мобильных приложений. Если учитывать, что одновременно сопровождается несколько мобильных платформ, то в этом направлении в первую очередь необходима автоматизация. На текущий момент среди уже готовых решений есть два направления: утилиты, интегрированные в среды разработки, и удаленные сервисы тестирования, например, TestCloud, производства компании Xamarin [1]. Вариант с встроенным ПО является самым худшим, потому что разные компании используют различные среды разработки, а значит, ограничивается возможность распространения системы тестирования. Использование удаленных сервисов тоже вызывает определенные неудобства: тестовый сценарий сервис генерирует сам, либо его нужно заранее описать на понятном сервису языке, что влечет за собой дополнительные затраты по времени [2]. Также подобные сервисы не проводят все тесты на реальных устройствах, а часть ошибок проявляется именно на них. Самостоятельная разработка системы тестирования позволяет избежать описанных недостатков: написание сценария можно вести на привычном языке программирования, нет необходимости в жесткой связи со средой разработки, а также можно гарантированно использовать набор реальных устройств.

Для пробного проекта наиболее удобной является платформа Android, поскольку она является достаточно открытой, и средства разработки для нее в основном являются бесплатными.

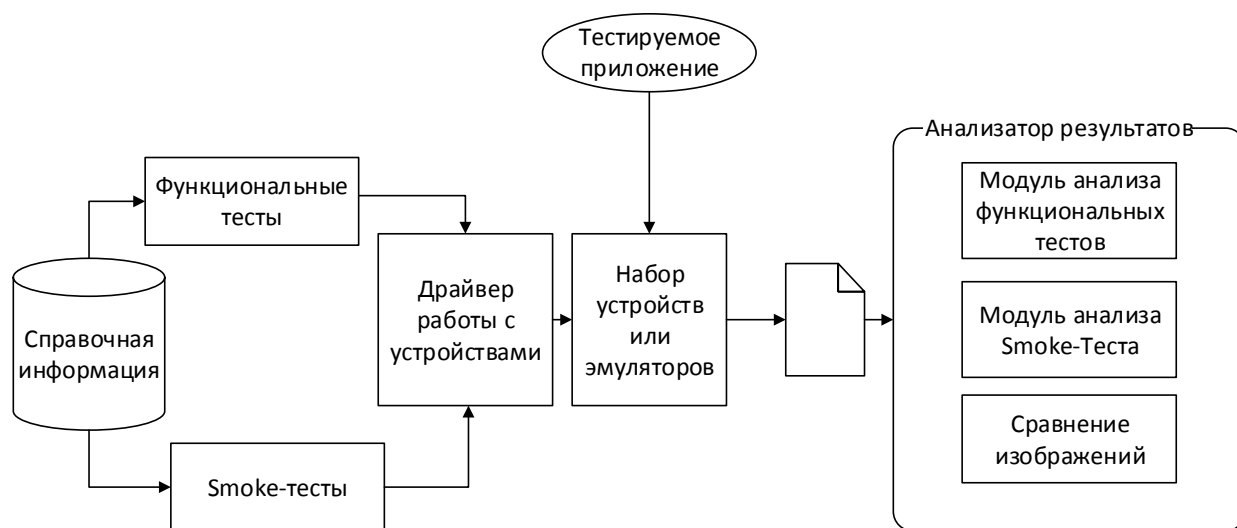


Рисунок 1 – Структурная схема системы тестирования

Тестовая структура была построена на основе фреймворка Robotium. Средство тестирования представляет собой структуру, представленную на рисунке 1. Вся справочная информация хранится в базе данных и конфигурационных файлах, поскольку компоненты могут быть

разнесены по различным компьютерам и обмениваться данными по сети. Функциональные и smoke-тесты основаны на различной логике и разнесены в два приложения. При запуске каждого из сценариев приложение собирается и при помощи драйверов загружается в тестовое устройство или эмулятор – на схеме они обозначены одним условным блоком, потому что в различных ситуациях используются различные наборы драйверов: androiddebugbridge, драйвера конкретных устройств. Также возможно использование промежуточного интерфейса Spoon, который позволяет запускать один и тот же сценарий одновременно на нескольких устройствах [3].

Функциональные тесты основаны на поиске текстов и элементов, содержащих эти тексты: например, ищутся все поля на итоговом чеке, содержащие данные о суммах. Сами тесты реализованы с учетом особенностей тестируемых приложений. Так, например, тесты «умеют» генерировать номера банковских карт, запрашивать у компонент внешних систем контрольные суммы по необходимым протоколам. Существующая реализация тестовых сценариев позволяет учитывать даже мелкие особенности такие, как частота появления определенных сообщений: например, всевозможных напоминаний. Использование удаленных сервисов для тестирования не дает такого полного охвата. Также учитывается и другая особенность проверяемых приложений: динамическая генерация идентификаторов меню. Автотесты полностью охватывают элементы меню, при этом отсутствует привязка к какому-то конкретному устройству. В процессе выполнения после каждого пункта сценария система перезапускает тестируемое приложение, с целью сброса кэша и т.д. Это делается для того, чтобы обеспечить возможность проведения тестирования на старых устройствах с малым объемом памяти.

Каждая ветка генерирует свой лог-файл, который перезаписывается при каждом запуске тестов. При формировании файлов используются различные подходы, основанные на логике тестов. Smoke-тест построен так, что если какая-либо операция не выполнялась, то зависящие от нее операции утилита даже не будет пытаться выполнить. В итоге в отчет будет добавлен перечень всех операций, которые не выполнялись с указанием причины. В основных функциональных тестах такая логика изначально не была заложена, поэтому в отчет пишутся служебные сообщения, в которых указывается, какой этап начат и закончен при помощи специальных тегов, а также сообщения с ошибкой при обработке исключений. Утилита-анализатор отчетов ищет для каждого этапа начало и конец и, если тег конца не найден, выводит сообщение об ошибке и отмечает этой ошибкой данный этап.

Литература

1. Xamarintestcloud [Электронный ресурс] / Xamarin. – 2013. – Режим доступа: <http://xamarin.com/test-cloud>. – Дата доступа: 22.12.2013.
2. Тестирование мобильных приложений в облаке [Электронный ресурс] / Открытые системы. – 2013. – Режим доступа: <http://www.osp.ru/os/2013/04/13035541/>. – Дата доступа: 02.01.2014.
3. Distributing instrumentation tests to all your Androids [Электронный ресурс] / Spoon. – 2013. – Режим доступа: <http://square.github.io/spoon/>. – Дата доступа: 22.12.2013.

Д.Л. Семенякин (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **А.И. Кучеров**, ст. преподаватель

ВЫБОР ВИДЕОКАМЕР ДЛЯ ВИДЕОНАБЛЮДЕНИЯ

Правильный подбор и размещение видеокамер – одна из самых важных задач при проектировании системы видеонаблюдения на любом объекте. И успех в решении этой задачи напрямую зависит от понимания характеристик, возможностей и вариантов применения существующих на данный момент видеокамер. Именно этому и посвящена данная статья.

На рисунке 1 ниже отображена обобщенная функциональная схема видеокамеры:

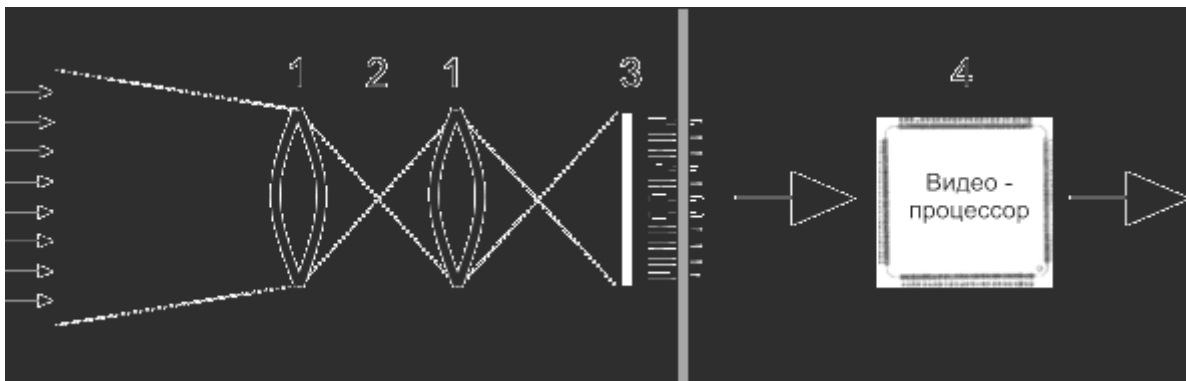


Рисунок 1 – Устройство видеокамеры

Как видим, оптическая часть видеокамеры функционально очень похожа на человеческий глаз: изображение проходит через систему линз (1), диафрагму (2) и проецируется на фотоприемную ПЗС-матрицу (3). Далее, в ПЗС-матрице (ПЗС – Прибор с Зарядовой Связью) изображение преобразуется в электрический сигнал, который поступает уже в электронную часть (4) видеокамеры и там подвергается обработке. Электроника видеокамеры может, в свою очередь, управлять

диафрагмой для регулировки яркости изображения. У большинства современных видеокамер (для упрощения конструкции и удешевления) регулировка яркости реализована не через механическое управление диафрагмой (АРД – автоматическая регулировка диафрагмы), а с помощью электронной обработки сигнала, поступающего с ПЗС-матрицы – так называемый «электронный затвор». Кроме того, существуют видеокамеры, которые могут и управлять системой линз для возможности оптического увеличения изображения («Zoom»), и поворачиваться в нужном направлении (поворотный механизм) по команде оператора, который в данный момент производит оперативное наблюдение (правда, такие видеокамеры очень дорогие и при этом такие возможности в большинстве случаев не нужны, особенно если нет этого самого оператора, а изображения со всех камер записываются видеорегистратором).

Теперь более подробно остановимся на каждой из характеристик современных видеокамер, применяющихся для видеонаблюдения:

Видеокамеры бывают наружного исполнения («уличные»), и для отапливаемых помещений («внутренние»). Основные требования к уличным камерам – герметичный корпус, возможность работы при большом диапазоне температур (от -30°C до $+60^{\circ}\text{C}$) и наличие козырька для защиты стекла от воды и боковых солнечных лучей. В большинстве уличных видеокамер встроен нагревательный элемент с термодатчиком – для работы при низких температурах, хотя, есть экземпляры, которым хватает для работы и собственного тепла, выделяемого электроникой (но у этих видеокамер либо нижний диапазон температур температур только от минус 10°C , либо у них могут быть проблемы в жаркое время года из-за перегрева). Есть еще один вариант наружной установки видеокамер – термокожух с подогревом в него можно установить, например, модульную видеокамеру, или более серьезную камеру с хорошим объективом с автодиафрагмой.

Различают корпусные (рисунок 2) с объективом, корпусные без объектива и бескорпусные (модульные) (рисунок 3).

Корпусные с объективом – это компактные камеры, поставляемые с фиксированным объективом, который в половине случаев заменить на другой не получится (в некоторых моделях возможность замены объектива предусмотрена изначально, в некоторых – только при полной разборке корпуса камеры, а в некоторых замена объектива в принципе невозможна), кроме того, у этих камер обычно вообще нет диафрагмы (за исключением единичных моделей), а регулировка яркости производится с помощью электронной обработки видеосигнала.

Корпусные без объектива – видеокамеры, у которых объектив сменный и поставляется отдельно от камеры, при этом есть возможность

использовать объективы с любыми возможностями (автодиафрагма, ручная диафрагма, регулировка фокусного расстояния – «ручной зум», и т.д.). У этих камер, в зависимости от установленного объектива, наиболее широкие возможности по подстройке угла обзора и диафрагмы непосредственно на месте установки, возможность использования автодиафрагмы при уличной установке (в термокожухе). Недостатки этих камер – большие размеры и высокая цена.



Рисунок 2 – Корпусные с объективом



Рисунок 3 – Корпусные без объектива

Бескорпусные (модульные) – в общем, это то же самое, что и корпусные с объективом, но только без корпуса: фиксированный объектив без диафрагмы (с возможностью замены), электронная регулировка яркости (правда, есть отдельные единичные модели с диафрагмой на объективе и ручным зумом).

Эти камеры чаще всего используются для установки в термокожухе (для уличной установки), скрытой установки, или в любых других местах, где требуются небольшие размеры при низкой стоимости.

Для установки на улице или сырых неотопливаемых помещениях (склады, ангары, и т.д.) годятся либо видеокамеры в герметичных корпусах с подогревом, либо установленные в термокожух с подогревом. Высоту установки желательно выбирать выше 3 м от земли (или пола) – чтобы усложнить возможность кражи видеокамеры. Но, с другой стороны, очень высоко устанавливать тоже не нужно, иначе объекты наблюдения будут маленькими и трудноразличимыми (хотя, это решается с помощью правильного подбора объектива), к тому же людей в этом случае мы будем видеть сверху, а не сбоку, а по макушке узнать человека намного сложнее, чем по лицу. При выборе места установки видеокамеры нужно учитывать и расположение сторон света – прямой солнечный свет на восходе или закате не должен попадать прямо в объектив для исключения «засветки». Иначе несколько часов в день

изображение будет «залито белым» и разглядеть на нём что-либо будет очень сложно. Еще один важный момент – ночное уличное освещение. Если его нет совсем или оно очень слабое, то самым простым и правильным решением будет поставить дополнительные фонари или прожектора, но если это трудно сделать, то придется устанавливать видеокамеры с высокой чувствительностью (малым количеством люксов) и объективами с автоматической диафрагмой. Видеокамеры желательно устанавливать подальше от фонарей и прожекторов (минимум 1,5–2 м). Варианты с установкой видеокамер со встроенной инфракрасной подсветкой желательно тоже исключить – как уже упоминалось выше, фонари и ИК-подсветка привлекают к себе ночных насекомых, которые будут летать очень близко возле объектива и сбивать с толку детектор движения. В результате, система видеонаблюдения будет записывать много бесполезной информации (движение самих насекомых, пауков, качание паутины на ветру (а ночью, подсвеченная ИК-подсветкой, паутина видна в виде ярких белых линий на изображении)). Мало того, что нужно будет время от времени очищать видеокамеру от паутины и пауков, так еще и при просмотре записей придется потратить уйму драгоценного времени, наблюдая за полетами насекомых и качанием паутины. Кроме того, из-за этой ненужной информации уменьшится полезная ёмкость диска видеорегистратора (на одном из объектов вместо 26–30 дней которые могла записать система видеонаблюдения зимой, летом на этом же диске помещалось только 8–12 дней, а виной всему – две уличные видеокамеры с ИК-подсветкой). При установке на стенах зданий, желательно учитывать слив воды с крыш в дождливую погоду и расположение водостоков. Если во время дождя потоки воды с крыши будут попадать на саму камеру или вблизи камеры в поле зрения объектива, то это тоже будет мешать нормальной работе детектора движения системы видеонаблюдения, и камеру придется переставить в другое место.

В заключение хочется отметить, что правильный выбор места установки и модели видеокамеры – это только первый этап при проектировании системы видеонаблюдения в целом. С выхода видеокамеры видеосигнал будет передаваться дальше по кабельным линиям до видеорегистратора или монитора. И следующий шаг – расчет кабельной системы и выбор видеорегистратора. От того, насколько грамотно это будет сделано зависит и качество полученного изображения и эффективность всей системы.

Литература

1. Лыткик, А.А. IP-видеонаблюдение: наглядное пособие / А.А. Лыткин – Москва: Издательство Security Focus, 2011. – 200 с.
2. Торстен, А. Видеоаналитика: Мифы и реальность / Торстен Анштедт. – Москва: Издательство Security Focus, 2012. – 176 с.

Д.А. Сердюков (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **В.С. Давыдов**, канд. тех. наук, доцент

ВЫСОКОЧАСТОТНЫЕ ЗАПРОСЫ

Среди самых дорогостоящих, но в то же время наиболее эффективных мероприятий в комплексе продвижения сайта, является метод раскрутки web-ресурса с помощью высокочастотных (или сокращенно ВЧ) запросов. К ним относят самые популярные запросы, вводимые пользователями в строки поиска, и наиболее часто – обычно это короткие, но в то же время весьма пространные понятия.

Условная градация запросов под высокочастотными ключами подразумевает, прежде всего, те слова, частота запросов которых достигает нескольких десятков тысяч в день. И, разумеется, странички web-ресурса, выведенные в топ выдачи поиска по конкретному высокочастотному запросу, способны обеспечивать массовую посещаемость сайта, превосходящую порой сотни тысяч пользователей в месяц. Нельзя точно назвать то число, которое бы могло считаться наиболее полным признаком высокочастотного запроса, но при этом можно с уверенностью считать таковыми те слова, частота запросов которых лидирует с большим отрывом от остальных.

В отличие от низко- и среднечастотных, высокочастотные запросы представляют собой более обобщенные вариации поиска. В результате этого многие пользователи, оказавшиеся на отдельно взятом ресурсе, куда привел их высокочастотный запрос, являются там «случайными гостями», поскольку в большинстве случаев информация на обнаруженном сайте не всегда соответствует тому, что они искали. Говоря иначе, высокочастотные запросы способны обеспечить высокую посещаемость отдельного web-сайта, но в то же время – они не могут гарантировать посещение ресурса представителями именно целевой аудитории. Пример высокочастотного запроса – «Туры в Египет». Вроде бы все люди, вводящие это словосочетание в поисковик, хотят одного и того же – отдохнуть в Египте, а для этого найти турагентство, которое продаст им путевку. На деле же оказывается, что кто-то хочет полететь исключительно в Хургаду, а кто-то – не иначе как в Шарм-эль-Шейх; кому-то удобнее вылетать в Египет из Москвы, а кому-то – из аэропорта где-то в Сибири. Кто-то ищет тур на будущее, кому-то нужна горящая путевка. Вот и выясняется, что на сайте, на который привет посетителя его запрос, нужной информации вовсе нет.

SEO-специалисты со стажем утверждают, что использование метода продвижения по высокочастотным запросам способно явиться отличным стартом для начала масштабных акций и кампаний, рассчитанных

на широкий круг пользователей. Причем многие из этих пользователей в дальнейшем могут представлять из себя постоянных клиентов продвигаемого проекта.

Выражаясь образно, высокочастотные запросы в плане конкуренции представляют собой поле битвы, где ведут ожесточенную борьбу множество крупных и сильных ресурсов. Здесь уместно сказать о том, что продвижение молодого сайта по высокочастотным запросам неспособно привести ни к чему, кроме лишних затрат времени и денежных средств, ибо конкуренция в данной сфере чрезвычайно высока.

Но при этом, если использование данного метода продвижения происходит в сочетании с другими, например, с методами раскрутки при помощи среднечастотных и низкочастотных запросов, то эффективность прикладываемых усилий многократно повышается. Высокочастотные запросы обеспечивают посещаемость ресурса, а, в свою очередь, средне- и низкочастотные запросы привлекают внимание не всех посетителей подряд, а именно целевой аудитории. Однако осуществление подобных мер по силам только опытным специалистам, поскольку в каждом отдельно взятом методе существуют свои специфические нюансы.

К примеру, неосведомленному человеку данная задача покажется вполне осуществимой. Но такое поверхностное представление о сложностях раскрутки ресурса часто ведет к противоположному эффекту: алгоритмы работы современных поисковых систем столь совершенны, что перенасыщенный одними и теми же словами текст они автоматически причисляют к бесполезным для пользователей, и сайт, содержащий подобную информацию, моментально теряет свои позиции в рейтинге выдачи результатов поиска. А в особо «тяжелых случаях» - и вовсе отправляется в бан, откуда выволить его очень сложно.

Продвижение высокочастотными запросами – это задача, обычно стоящая перед крупными и довольно долго существующими в сети сайтами. Из года в год они сражаются между собой за лидирующие позиции в топе, который выводится пользователю по высокочастотному запросу.

Новичкам же предпочтительнее всего начинать заявлять о себе менее сложными методами. Среди них как раз и выделяют медленное, но уверенное и стабильное продвижение веб-ресурса при помощи низкочастотных и среднечастотных запросов. А уже дальше, доверившись опытным seo-специалистам, владелец сайта может позволить своему детищу постепенно вникать в борьбу с маститыми и известными интернет-проектами

Т.С. Скринникова (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **В.Н. Кулинченко**, ст. преподаватель

МОДУЛЬ РАСПОЗНАВАНИЯ ЛИЦ

Модуль распознавания лиц предназначен для автоматической идентификации личности по видеоизображению. Он производит распознавание лиц, захваченных детектором лиц, сравнивая их с заранее созданной базой эталонных изображений.

Модуль захвата и распознавания лиц обеспечивает высокий процент распознавания и может использоваться совместно со СКУД для увеличения уровня контроля доступа на объектах с повышенными требованиями к обеспечению безопасности, например, в банках или на режимных предприятиях. Другим важным применением модуля является автоматизация фейс-контроля в казино, гостиницах, ресторанах и на других подобных объектах.

Область применения – это интеллектуальный видеоконтроль железнодорожных и автовокзалов, аэропортов, метрополитена, мест проведения спортивных и зрелищных мероприятий.

Использование модуля распознавания лиц для видеоконтроля обеспечивает в реальном времени регистрацию всех людей, прошедших через рубеж контроля, предоставление данных о присутствии разыскиваемых персон.

Правоохранительные органы могут эффективно использовать систему для идентификации террористов, преступников, нарушителей при интеграции с собственными базами данных.

Контроль и ограничение доступа на особо охраняемые объекты с пропускным режимом (министерства, научные институты и лаборатории, медицинские учреждения), промышленные предприятия, школы, банки, офисы.

Модуль распознавания лиц является эффективным инструментом для предотвращения проникновения "нежелательных людей" на охраняемую территорию за счет идентификации допущенных лиц.

Принцип действия

I этап. Поиск лица в видеопотоке

Поиск лиц в видеопотоке для их дальнейшего распознавания производится путем выделения симметрий в каждом видеокadre. Для этого применяется определенный набор симметричных сверток в заданном диапазоне масштабов изображения, после чего видеоизображение обрабатывается нейросетью.

Этот алгоритм обеспечивает такие преимущества данной технологии как устойчивость к шуму и неравномерной засветке лица.

II этап. Трекинг (отслеживание) лица между кадрами видеопотока

Лицо человека, единожды попав в поле зрения видеокамеры, с использованием алгоритма предсказания вектора движения и корреляционных алгоритмов будет автоматически отслеживаться от кадра к кадру. Все изображения будут сохраняться во временном буфере. В результате будет выбран кадр с оптимальным ракурсом лица и качеством изображения.

III этап. Выделение основных признаков лица

На этом этапе с использованием алгоритма анализа контуров производится выделение на видеоизображении лица его основных признаков: глаз, носа, рта и т.д.

Преимуществом этого метода является надежная работа при нефронтальном позиционировании лица.

IV этап. Нормализация изображения лица

После выделения основных признаков лица, его изображение приводится к стандартному виду: для надежного распознавания изображение лица должно иметь определенные размеры, необходимо выдержать расстояние между глазами, положение лица относительно центра. Для этого изображение масштабируется, разворачивается, в некоторых случаях также определяется положение лица (фас, положение в три четверти или точные 3D координаты), автоматически нормализуется яркость и контрастность.

V этап. Преобразование индивидуальных признаков лица

Непосредственное сравнение полученного изображения лица с изображениями из базы данных – ресурсоемко и неэффективно. Решение этой проблемы преобразование его характерных признаков в форму, максимально подходящую для быстрого сравнения – в вектор малой размерности.

VI этап. Сравнение полученного векторного представления с базой данных

Далее возможны две схемы работы модуля распознавания лиц: идентификация и верификация.

В режиме идентификации захваченное лицо сравнивается со всеми изображениями лиц, хранящимися в базе данных. Таким образом, выясняется, в частности, наличие человека в базе данных нежелательных посетителей или VIP-клиентов какого-либо заведения.

В режиме верификации лицо человека, воспользовавшегося карточкой-пропуском или каким-либо другим идентификатором личности для прохода через турникет или дверь с электронным замком, сравнивается с фотографией владельца пропуска, хранящейся в базе данных. Таким образом, можно выяснить, является ли человек, пытающийся получить доступ, тем, за кого он себя выдает.

Функции:

- идентификация – сравнение лиц людей с изображениями из собственной или внешней базы данных;
- распознавание лиц в режиме реального времени;
- отображение на экране распознанных лиц и сопутствующих данных: номера камеры, даты и времени распознавания, эталонной фотографии из базы данных, ФИО, степени сходства и комментария, введенного при добавлении лица в базу;
- печать фотографии и персональных данных распознанных личностей на принтере, сохранение изображений распознанных лиц в формате bmp или jpeg;
- просмотр видеофрагмента, соответствующего моменту распознавания лица;
- поиск и отображение на экране всех распознанных за определенный временной интервал лиц. Поиск ведется по ФИО и номеру камеры с фильтрацией по степени сходства;
- поиск личности в базе данных по фотографии;
- добавление в базу данных лиц новых записей, содержащих цифровую фотографию, персональные данные человека и комментарий;
- проверка фотографий в базе данных на соответствие биометрическим стандартам для систем автоматической идентификации личности. Возможна проверка как фотографии одного человека, так и всех имеющихся фотографий. Результатом проверки является отчет, который сохраняется и впоследствии отображается при просмотре фотографии;
- отображение на экране статистики распознавания: общее количество кадров, захваченных детектором лиц, количество кадров, на которых алгоритмом распознавания лиц обнаружено лицо, количество кадров, на которых обнаружены глаза, и количество распознанных лиц.

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

Науч. рук. **Е.Е. Пугачева**, ассистент

РАЗРАБОТКА БАЗЫ ДАННЫХ ДЛЯ ПРИЛОЖЕНИЯ ПО АВТОМАТИЗАЦИИ И УЧЕТУ ОБОРУДОВАНИЯ НА ПРЕДПРИЯТИИ

В разработке любого приложения, предназначенного для хранения информации, большую роль играет использование базы данных в качестве хранилища для информации. Такой подход позволяет впоследствии уменьшить размер самого хранилища, за счет нормализации базы данных и приведения её к третьей нормальной форме.

В спроектированной базе данных, за счет нормализации уменьшается избыточность хранения информации – хранимая информация содержит минимум дублирующих данных, она при этом хорошо структурирована. Базы данных используются повсеместно для хранения данных различных приложений. Задача программиста – правильно провести нормализацию базы данных и привести ее к третьей нормальной форме.

База данных представляет собой структурированную совокупность данных. Для записи, выборки и обработки данных, хранящихся в компьютерной базе данных, необходима система управления базой данных, каковой и является ПО MySQL. Поскольку компьютеры замечательно справляются с обработкой больших объемов данных, управление базами данных играет центральную роль в вычислениях. Реализовано такое управление может быть по-разному – как в виде отдельных утилит, так и в виде кода, входящего в состав других приложений.

MySQL – это система управления реляционными базами данных, в которых данные хранятся не все скопом, а в отдельных таблицах, благодаря чему достигается выигрыш в скорости и гибкости. Таблицы связываются между собой при помощи отношений, благодаря чему обеспечивается возможность объединять при выполнении запроса данные из нескольких таблиц. SQL как часть системы MySQL можно охарактеризовать как язык структурированных запросов плюс наиболее распространенный стандартный язык, используемый для доступа к базам данных.

MySQL – это ПО с открытым кодом. Применять его и модифицировать может любой желающий. Такое ПО можно получать по Internet и использовать бесплатно. При этом каждый пользователь может изучить исходный код и изменить его в соответствии со своими потребностями.

Для разрабатываемого приложения была спроектирована база данных, которая будет хранить информацию о пользователях, которые будут иметь доступ к спроектированному приложению, информацию о правах доступа пользователей, об оборудовании, установленном на объектах предприятия, о городах, улицах, домах, где предприятие имеет обслуживаемые объекты.

Для доступа из РНР к данным, хранящимся в базе, был использован непроцедурный язык SQL (Structured Query Language – структурированный язык запросов) ориентирован на операции с данными, представленными в виде логически взаимосвязанных совокупностей таблиц. Особенность предложений этого языка состоит в том, что они ориентированы в большей степени на конечный результат обработки данных, чем на процедуру этой обработки. SQL сам определяет, где находятся данные, какие индексы и даже наиболее эффективные последовательности операций следует использовать для

их получения: не надо указывать эти детали в запросе к базе данных.

Потому как вся информация, хранящаяся приложением, будет структурирована и объем самой базы данных будет относительно невелик, по сравнению к, например, записи информации в файл, то доступ к данным будет максимально быстрым и приложение будет способно максимально быстро обрабатывать запросы пользователей.

Литература

1. Грабер, М. Введение в SQL / М. Грабер. – Лори, 1992. – 382 с.
2. Краева, Т.А. Методология и организация учета в условиях автоматизации / Т.А. Краева. – М. Финансы и статистика, 1992. – 380 с.
3. Когаловский, М.Р. Технология баз данных на персональных ЭВМ / М.Р. Когаловский. – М.: Финансы и статистика, 1992. – 224 с.
4. Кузнецов, С.Д. Основы современных баз данных / С.Д. Кузнецов. – М.: «Экон», 2003. – 412 с.

К. И. Слесаренко (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

ПРИЁМЫ И МЕТОДЫ ПОРОЖДАЮЩЕГО ПРОГРАММИРОВАНИЯ

Порождающее программирование – технология разработки программного обеспечения, основанная на моделировании семейства программных систем, используя которые можно по конкретным техническим требованиям автоматически получить специализированный и оптимизированный промежуточный или конечный программный продукт из элементарных, многократно используемых компонентов реализации с помощью базы знаний о конфигурациях. Порождающее программирование – автоматизированное производство программных продуктов из отдельных компонентов, как это происходит при производстве промышленных изделий, автомобилей. Ключевыми моментами здесь являются *унификация, автоматизация и гибкость*. Современные сборочные линии позволяют создавать автомобили с различными конфигурациями. Например, для Mercedes-Benz класса E имеется 8000 вариантов водительских мест и 10 000 вариантов сидений.

В отличие от методологии компонентного программирования в порождающем программировании происходит автоматическая сборка программного продукта. Компонентно-ориентированное программирование можно уподобить сборке автомобиля из конструктора, в то время

как в порождающем программировании задаются требования к синтезируемой системе: клиент заказывает у дилера желаемую конфигурацию модели. Для реализации такой технологии необходимы:

- компоненты реализации, с учетом общей архитектуры данной линейки продуктов;
- модель знаний о трансляции абстрактных требований в конкретные связи компонентов;
- реализация этих знаний в генераторах.

Рассмотрен метод DEMRAL порождающего программирования. Это «метод разработки алгоритмических библиотек многократного применения на основе инженерии предметной области» – метод инженерии предметной области, специализированный для разработки порождающих алгоритмических библиотек. К категории алгоритмических относятся библиотеки численного анализа, контейнеров, обработки изображений, распознавания изображений речи, графовых вычислений и т. д. Реализована библиотека перестановок.

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

Науч. рук. **Н.Б. Осипенко**, канд. физ.-мат. наук, доцент

ИСПОЛЬЗОВАНИЕ ЭЛЕКТРОННОГО ПОСОБИЯ НА ТЕМУ «КОРРЕЛЯЦИОННО-РЕГРЕССИОННЫЙ АНАЛИЗ В РАЗНЫХ ПРОГРАММНЫХ СИСТЕМАХ» В ФИЗИКЕ

Обработка статистических данных уже давно применяется в самых разнообразных видах человеческой деятельности. Вообще говоря, трудно назвать ту сферу, в которой она бы не использовалась.

Например, статистическая физика – раздел физики, одна из задач которой – выразить свойства макроскопических тел, т.е. систем, состоящих из очень большого числа одинаковых частиц (молекул, атомов, электронов и т.д.), через свойства этих частиц и взаимодействие между ними. В свою очередь, изучением макроскопических тел занимаются и другие разделы физики – термодинамика, механика сплошных сред, электродинамика сплошных сред.

Однако при решении конкретных задач методами этих дисциплин в соответствующие уравнения всегда входят неизвестные параметры или функции, характеризующие данное тело. Так, для решения задач гидродинамики необходимо знать уравнение состояния жидкости или газа, т.е. зависимость плотности от температуры и давления, теплоёмкость жидкости, её коэффициент вязкости и т.п. Все эти зависимости и

параметры можно, разумеется, определять экспериментально, поэтому методы, о которых идёт речь, называются феноменологическими. Статистическая же физика позволяет во многих случаях вычислить все эти величины, если известны силы взаимодействия между молекулами.

Представим, что выполняя лабораторную работу по физике, например, изучая зависимость некоторой физической величины Y от физической величины X , получили следующие экспериментальные данные:

X	x_1	x_2	...	x_n
Y	y_1	y_2	...	y_n

Для того чтобы наглядно представить зависимость Y от X , нанесли экспериментальные точки на координатную плоскость и заметили, что экспериментальные точки не лежат на одной прямой. Причина такого результата – случайные погрешности эксперимента, вызванные действием большого числа факторов, учесть которые невозможно. Как же правильно построить график зависимости $Y=Y(X)$? Как найти функцию $Y=Y(X)$, которая бы наилучшим образом соответствовала экспериментальным точкам?

Использование возможностей современной вычислительной техники, оснащенной пакетами программ машинной обработки статистической информации на ЭВМ, делает практически осуществимым оперативное решение задач изучения взаимосвязи показателей методами корреляционно-регрессионного анализа. При машинной обработке исходной информации на ЭВМ, оснащенных пакетами стандартных программ ведения анализов, вычисление параметров, построение графиков зависимости $Y=Y(X)$ и т.п. является быстро выполняемой операцией.

Корреляционно-регрессионный анализ наиболее легко и удобно реализовывать в таких средах, как Excel, MathCAD и Statistica.

Excel – программный продукт с богатыми возможностями, который можно использовать на многих уровнях. Электронная таблица Excel, вероятно, самая простая в использовании программа. Многие ее возможности были специально разработаны для того, чтобы и начинающие, и профессиональные пользователи могли выполнять повседневные задачи просто и быстро. При выполнении определенных процедур программа ведет пользователя шаг за шагом, основные команды редактирования и форматирования просты и понятны [2].

MathCAD – универсальный математический пакет, предназначенный для выполнения инженерных и научных расчетов. Основное преимущество пакета – естественный математический язык, на котором формируются решаемые задачи. Он имеет удобный для описания аналитических моделей интерфейс, ориентированный на классическую форму

языка математики. Программа достаточно проста в освоении для того, чтобы ее применяли не только ученые, инженеры или студенты вуза, но и школьники старших классов.

Так же MathCAD имеет встроенный язык программирования. С одной стороны это вынужденная мера, поскольку разработчики не могут предвосхитить и обеспечить все потенциальные требования пользователей. Необходимость освоения языка является недостатком программы. С другой стороны, синтаксис языка довольно простой, его использование может существенно расширить круг задач, решаемых исследователями, при этом вспомогательные инструменты, в частности, построение графиков всегда у него под рукой [3].

Statistica – наиболее распространенная универсальная статистическая система фирмы StatSoft Inc., созданная в начале 90-х годов для среды Windows, позволяющая проводить исчерпывающий, всесторонний анализ данных, представлять результаты анализа в виде таблиц и графиков, автоматически создавать отчеты о проделанной работе. Statistica предлагает широкий спектр линейных и нелинейных средств моделирования, поддерживает непрерывные и категориальные предсказания, взаимодействия, иерархические модели, возможность автоматического выбора моделей, а также другие методы [3].

На основе проведенного исследования особенностей решения задач анализа данных в средах Excel, MathCAD и Statistica разработано электронное пособие, которое состоит из трёх блоков: первый блок содержит теоретический материал о корреляционно-регрессионном анализе, второй – практическую часть с описанием возможностей проведения корреляционно-регрессионного анализа в различных программных средах, третий – образцы примеров и практические задания по лабораторным занятиям. Наряду с сопровождающей частью по решению задачи корреляционно-регрессионного анализа в средах Excel, MathCAD и Statistica, разрабатываются комментарии по выполнению этой задачи и в других программных средах. Пособие прошло апробацию при выполнении студентами заданий по лабораторным работам и показало свою хорошую работоспособность.

Литература

1. Рудикова, Л.В. MicrosoftExcel : учебное пособие / Л.В. Рудикова. – СПб. : БХВ-Петербург, 2005. – 368 с.
2. Алябьева, С.В. Mathcad для студентов : справочное издание / С.В. Алябьева [и др.] – Петрозаводский государственный университет, 2007. – 154с.
3. Боровиков, В.П. Популярное введение в STATISTICA : справочное издание / В.П. Боровиков. – М. : КомпьютерПресс, 1998. – 267с.

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Криптография – это наука об обеспечении безопасности данных. Она занимается поисками решений четырех важных проблем безопасности – конфиденциальности, аутентификации, целостности и контроля участников взаимодействия. Задача криптографии, т.е. тайная передача, возникает только для информации, которая нуждается в защите.

Шифрование – это преобразование данных в нечитабельную форму, используя ключи шифрования-расшифровки. Шифрование позволяет обеспечить конфиденциальность, сохраняя информацию в тайне от того, кому она не предназначена.

Традиционной задачей криптографии является проблема обеспечения конфиденциальности информации при передаче сообщений по контролируемому противником каналу связи. В простейшем случае эта задача описывается взаимодействием трех субъектов (сторон). Владелец информации, называемый обычно отправителем, осуществляет преобразование исходной (открытой) информации (сам процесс преобразования называется шифрованием) в форму передаваемых получателю по открытому каналу связи зашифрованных сообщений с целью ее защиты от противника (рисунок 1).



Рисунок 1 – Передача зашифрованной информации

Под противником понимается любой субъект, не имеющий права ознакомления с содержанием передаваемой информации. В качестве противника может выступать криптоаналитик, владеющий методами раскрытия шифров. Законный получатель информации осуществляет расшифрование полученных сообщений. Противник пытается овладеть защищаемой информацией (его действия обычно называют атаками).

Литература

1. Алферов, А.П. Основы криптографии: учеб. пособие. 3-е изд., испр. и доп. / А.П. Алферов [и др.] – М.: 2005. – 480 с.
2. Введение в криптографию / Под общ. ред. В.В. Яценко. – 3-е изд., доп. М.: 2000. – 288 с.

М.В. Сосновский (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **П.Л. Чечет**, канд. тех. наук, доцент

ОСНОВНЫЕ ПОНЯТИЯ В КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЕ ИНФОРМАЦИИ

При криптографической защите информации наиболее важные понятия скрываются под определениями *шифр*, *ключ* и *криптографический алгоритм*.

Под *шифром* обычно понимается семейство обратимых преобразований, каждое из которых определяется некоторым параметром, называемым *ключом*, а также порядком применения данного преобразования, называемым *режимом преобразования*.

Ключ – это важнейший компонент шифра, отвечающий за выбор преобразования, применяемого для зашифровывания конкретного сообщения. Обычно ключ представляет собой некоторую буквенную или числовую последовательность. Эта последовательность как бы «настраивает» алгоритм шифрования.

Каждое преобразование однозначно определяется ключом и описывается некоторым *криптографическим алгоритмом*. Один и тот же криптографический алгоритм может применяться для шифрования в различных режимах. Тем самым реализуются различные способы шифрования. Каждый режим шифрования имеет как свои преимущества, так и недостатки. Поэтому выбор режима зависит от конкретной ситуации. При расшифровывании используется криптографический алгоритм, который в общем случае может отличаться от алгоритма, применяемого для зашифровывания сообщения. Соответственно могут различать ключи зашифровывания и расшифровывания. Пару алгоритмов зашифровывания и расшифровывания обычно называют *шифрсистемой*, а реализующие их устройства – *шифртехникой*.

Алгоритмы шифрования с использованием ключей предполагают, что данные не сможет прочитать никто, кто не обладает ключом для их расшифровки. Они могут быть разделены на два класса, в зависимости от того, какая методология криптосистем напрямую поддерживается ими.

Симметричные алгоритмы. Для шифрования и расшифровки используются одни и те же алгоритмы. Один и тот же секретный ключ используется для шифрования и расшифровки.

Асимметричные алгоритмы. Используется два разных ключа – один известен всем, а другой держится в тайне. Обычно для шифрования и расшифровки используется оба этих ключа. Но данные, зашифрованные одним ключом, можно расшифровать только с помощью другого ключа.

Криптография обеспечивает подотчетность, прозрачность, точность и конфиденциальность. Она предотвращает попытки мошенничества в электронной коммерции и обеспечивает юридическую силу финансовых транзакций. Криптография помогает установить вашу личность, но и обеспечивает вам анонимность.

Литература

1. Алферов, А.П. Основы криптографии: Учебное пособие. 3-е изд., испр. и доп. / А.П. Алферов [и др.] – М.: 2005. – 480 с.
2. Введение в криптографию/ Под общ. ред. В.В. Яценко. – 3-е изд., доп. М.: 2000.– 288 с.

Т.Г. Стадник (УО «ГГУ им. Ф. Скорины», Гомель)

Науч. рук. **М.И. Жадан**, канд. физ.-мат. наук, доцент

ЗАДАЧИ СИНХРОНИЗАЦИИ В АСПЕКТНО-ОРИЕНТИРОВАННОМ ПРОГРАММИРОВАНИИ

Аспектно-ориентированное программирование (АОР) – новое направление в программировании, предложенное исследователями из исследовательского центра Xerox PARC. Аспектно-ориентированное программирование содержит различные методы и методики разбиения задач на ряд функциональных компонентов, а также аспектов, которые «пересекают» функциональные компоненты, и предусматривает их композицию в целях получения реализаций систем.

Являясь одним из фундаментальных принципов инженерии, разделение задач применяется в ходе анализа, проектирования и реализации. В большинстве нотаций анализа и проектирования, равно как и в языках программирования, присутствуют конструкции, позволяющие структурировать описания систем в виде иерархических композиций мелких модульных элементов. При этом, эти конструкции ориентированы на выявление и построение функциональных элементов, которые, как правило, выражаются в виде объектов, модулей, процедур и т. д. Цель композиции заключается в том, чтобы средствами конкретного языка обеспечить однозначное кодирование требований. Инструментарием разработки в АОР являются языки программирования Java, AspectJCool, Smalltalk.

В частности, синхронизация стека связана с четырьмя ограничениями. Оптимальный вариант – выразить эти ограничения четырьмя согласованными операторами. Что касается фиксации аспектов, то есть три варианта:

Кодировать поддержку аспектов в виде традиционной библиотеки (например, классов или процедур).

Составить для выражения аспекта специальный язык.

Составить для выражения аспекта расширение языка.

Решены задачи, связанные с синхронизацией стека. Приведена «спутанная» реализация синхронизированного стека средствами Java и реализация стека на Java и его агентом синхронизации Cool. Рассмотренный здесь координатор стека относится к категории поэкземплярных, т.е. каждому экземпляру Stack должен соответствовать отдельный экземпляр координатора. Заметим, что Cool также поддерживает поклассовые координаторы, которые могут применяться совместно с несколькими классами.

Н.В. Старовойтов (УО «ГГУ им. Ф. Скорины», Гомель)

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

РАЗРАБОТКА ПРОЕКТА УПРАВЛЕНИЯ ЗАКАЗАМИ ДЛЯ ООО «ЦЕНТР ТЕХНИЧЕСКИХ РЕШЕНИЙ»

Стремительное развитие информационных технологий приводит к их широкому внедрению во многие сферы жизнедеятельности общества. Одной из таких структур, в которой глобальные компьютерные сети предоставляют широкие возможности для их развития, является торговля и оказание услуг. Внедрение информационных технологий в сферу торговли началось еще в начале 90-х годов прошлого века. Именно тогда появились системы для автоматизации деятельности продавцов (SFA – Sales Force Automation).

С тех пор системы SFA усовершенствовались и появились CRM-системы, предназначенные для автоматизации стратегий взаимодействия с клиентами, в частности, для повышения уровня продаж, оптимизации маркетинга и улучшения обслуживания клиентов путём сохранения информации о клиентах и истории взаимоотношений с ними.

В результате проделанной работы, была разработана система для приема заказов и управления ими. Администратор системы имеет возможность добавлять новые заказы в систему, а также управлять их статусами (принят, в работе, выполнен, в архиве). При добавлении нового заказа имеется возможность, как выбрать клиента из списка (в том случае, если клиент обращался ранее и информация о нем сохранена в базе данных), так и добавить нового (если клиент обращается впервые).

Администратор может легко получить выборку интересующих его заказов за определенный период времени, а также другую статистическую информацию по имеющимся в БД заявкам. Выборка производится посредством фильтров: по статусу выполнения, по времени поступления, по клиенту, оформившему заказ.

В системе ведется учет клиентской базы организации. Реализована возможность просмотра списка клиентов, их контактных данных, а также дополнительной информации (комментарии, индивидуальные скидки). В разработанном программном комплексе предусмотрена возможность получить информацию по заказам определенного клиента. В программном комплексе существует возможность оперативного уведомления клиента. Например, можно отослать уведомление о завершении работы над заказом либо о необходимости перезвонить для уточнения деталей. Уведомления могут отсылаться на e-mail клиента и посредством SMS-сообщений.

Список услуг, оказываемых организацией, можно редактировать. Существует возможность добавления новой услуги, а также удаления неактуальных услуг.

Система интегрирована в веб-сайт организации, который также оказывает информационную поддержку предприятия. На сайте размещены информационные материалы о предоставляемых услугах, новостях и проводимых акциях компании, а также контактная информация предприятия.

Пользователь после авторизации на сайте, попадает в личный кабинет, где может отслеживать выполнение заказов, смотреть историю выполненных заказов, а также получить информацию по полагающейся ему скидке на услуги. О смене статуса заказа пользователь может быть уведомлен посредством e-mail или SMS-сообщения.

Система реализована на PHP. В качестве базы данных используется MySQL. Для создания пользовательского интерфейса использован набор инструментов Twitter Bootstrap – простой и легко настраиваемый HTML, CSS и Javascript фреймворк для более быстрой и удобной Web-разработки.

Основной целью внедрения системы ставится увеличение степени удовлетворённости клиентов организации за счёт анализа накопленной информации о клиентском поведении, регулирования тарифной политики, настройки инструментов маркетинга. Система с легкостью может быть использована для реализации других подобных проектов.

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

Науч. рук. **А.В. Воружев**, канд. тех. наук, доцент

ПРЕДСТАВЛЕНИЕ УЧЕБНЫХ МАТЕРИАЛОВ В ФОРМАТЕ SCORM

Потребность в дистанционной форме обучения растет с каждым годом. Вместе с тем на рынке программных продуктов растет и количество программных продуктов для дистанционного обучения – систем дистанционного обучения (СДО).

Дистанционные образовательные технологии включают комплекс образовательных услуг, предоставляемых с помощью специализированной информационной образовательной среды, базирующейся на средствах обмена учебной информацией посредством сети Интернет. Информационно-образовательная среда дистанционных образовательных технологий представляет собой системно-организованную совокупность средств передачи данных, информационных ресурсов, протоколов взаимодействия, аппаратно-программного и организационно-методического обеспечения, ориентированную на удовлетворение образовательных потребностей пользователей.

Задачу обеспечения интероперабельности учебных курсов, независимости их от конкретной СДО можно решить за счет использования универсальных форматов представления обучающего контента, таких, как SCORM (Shareable Content Object Reference Model).

SCORM – набор технических стандартов для СДО. Основой модели SCORM является модульное построение учебников и учебных пособий, близкое к концепции модульных учебников. Модули учебного материала в SCORM называются разделяемыми объектами контента (SCO). SCO – автономная единица учебного материала, имеющая метаданные и содержательную часть. Совокупность модулей определенной предметной области называется в SCORM библиотекой знаний. Модули могут в различных сочетаниях объединяться друг с другом в составе учебников и учебных пособий.

Таким образом, использование SCORM обеспечивает возможность многократного использования учебных модулей, легкого сопровождения и адаптации курсов, ассемблирования контента отдельных модулей в учебные пособия в соответствии с индивидуальными запросами пользователей, способствует решению задачи поддержки интероперабельности учебных курсов.

В процессе работы над SCORM были сформулированы несколько требований ко всем системам, которые будут разрабатываться в соответствии с данным стандартом:

- доступность: способность определять местонахождение и получить доступ к учебным компонентам из точки удаленного доступа и предоставить их многим другим точкам удаленного доступа;
- адаптируемость: способность адаптировать учебную программу согласно индивидуальным потребностям и потребностям организаций;
- эффективность: способность увеличивать эффективность и производительность, сокращая время и затраты на доставку инструкции;
- долговечность: способность соответствовать новым технологиям без дополнительной и дорогостоящей доработки;
- интероперабельность: способность использовать учебные материалы вне зависимости от платформы, на которой они созданы;
- возможность многократного использования: способность использовать материалы в разных приложениях и контекстах.

Электронные курсы в формате SCORM предназначены для дистанционного обучения через Интернет с помощью специального программного обеспечения – LMS (Learning Management System – Системы управления учебным процессом). Другими словами, согласно спецификации стандарта SCORM, LMS определяет какую информацию и куда надо предоставить, и отслеживает работу пользователя с материалом.

Электронный курс – это структурированный материал по той или иной теме, предназначенный для решения заранее определенных задач обучения. Учебные материалы, выполненные в виде интерактивных страниц, позволяют сделать процесс обучения максимально эффективным и увлекательным – наряду с получением теоретических знаний. Наиболее часто и успешно электронные курсы используются для задач дистанционного обучения, повышения квалификации и развития компетенции сотрудников компаний.

При использовании дистанционных технологий появляется необходимость более формализованного подхода к построению учебного плана и системы изучаемого курса. Это связано и с более четкими требованиями к описаниям курса, системе оценки его усвоения, асинхронностью учебного процесса и с особенностями взаимодействия преподавателя и студентов.

Методика разработки электронного курса состоит из следующих этапов:

1 Отбор из избыточного содержания предметной области дисциплины (модуля) материалов для обеспечения формирования запланированных результатов обучения.

2 Структурирование отобранного содержания в соответствии с вышеизложенными принципами.

3 Разработка плана испытаний для контроля результатов обучения.

4 Разработка оценочных средств для контроля результатов обучения.

В центре процесса обучения находится самостоятельная деятельность обучающегося. Самостоятельное приобретение знаний не должно носить пассивный характер, наоборот, обучаемый должен быть вовлечен в активную познавательную деятельность. Испытания для контроля результатов обучения в электронном курсе должны проводиться после завершения каждого раздела электронного курса.

Литература

1. Актуальность создания механизмов поддержки интероперабельности учебных курсов [Электронный ресурс]. – В режиме доступа: <http://tm.ifmo.ru>. – Дата доступа: 15.03.14.

2. Обучение с использованием дистанционных технологий [Электронный ресурс]. – В режиме доступа: http://www.sde.ru/?page_id=1506. – Дата доступа: 15.03.14.

3. Стандарт SCORM и его применение [Электронный ресурс]. – В режиме доступа: <http://ccsr.ifmo.ru/scorm/>. – Дата доступа: 16.03.2014.

Т.Ю. Тищенко (УО «ГГУ им. Ф. Скорины», Гомель)

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

ОБЗОР ВОЗМОЖНОСТЕЙ КОМПЛЕКСА 1С: ПРЕДПРИЯТИЯ С ДРУГИМИ СИСТЕМАМИ

«1С: Предприятие» является мощной системой предназначенной для комплексной автоматизации бухгалтерского учета во всех мыслимых масштабах. Компонентность этой системы позволяет в каждом индивидуальном случае подобрать оптимальное решение, учитывающее специфику организации. Набор компонент может определяться индивидуально и в последствии пополняться.

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.

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.
- Любая компонента «1С: Предприятия» может использоваться как автономно, так и совместно с другими компонентами.
- При совместном использовании компоненты «1С: Предприятия» не только работают с общими данным в одной информационной базе, но и полностью объединяют свою функциональность. В этом случае пользователь работает, фактически, с единой системой, комплексно отражающей хозяйственную деятельность предприятия с использованием возможностей установленных компонент.

Для пользователей системы программ «1С: Предприятия» фирма «1С» предлагает набор решений по подключению различного оборудования, применяемого в торговле и складском деле: контрольно-кассовых машин, POS-терминалов, сканеров и принтеров штрих-кодов, дисплеев покупателя, электронных весов, терминалов сбора данных. Эти решения основаны на технологии внешних компонент и режимов экспорта-импорта информации.

Программные продукты системы «1С: Предприятие» способны «расти» вместе с организацией, в которой они работают. Спектр программных продуктов системы «1С: Предприятие» включает однопользовательские и сетевые версии; версии, работающие в архитектуре «клиент-сервер», а также дополнительную компоненту для организации обмена информацией между территориально удаленными филиалами организации. Все версии полностью функционально совместимы, что позволяет легко перейти от использования программы на одном рабочем месте к многопользовательским и распределенным вариантам использования системы.

Ниже приведены технологии, которые могут использоваться платформой «1С:Предприятие» при решении задач интеграции:

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

Внешнее соединение. Основная задача, решаемая с помощью внешнего соединения – обеспечение надежного и быстрого программного доступа к данным системы «1С:Предприятие» из внешних приложений.

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.

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.

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».

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.

Литература

1. Габец, А.П. Реализация прикладных задач в системе «1С: Предприятие 8.2» / А.П. Габец, Ю.М. Козырев, Д. В., Кухлевский, Д. С., Хрусталева, – Мн.: 1С-Паблишинг, 2010. – 720 с.
2. Постовалова, А. М. «1С:Бухгалтерия 8.2»/ А.М. Постовалова, С. Ф. Постовалов, – Санкт – Петербург: изд. Питер, 2011. – 256 с.
3. Селищев, Н. В. Администрирование системы «1С: Предприятие 8.2» / Н.В. Селищев, – Санкт – Петербург: изд. Питер, 2012. – 400 с.
4. Радченко, М. Г. «1С: Предприятие 8.2». Практическое пособие разработчика. Примеры и типовые приемы./ М.Г. Радченко, Е.Ю. Хрусталева, – Мн.: 1С-Паблишинг, 2009. – 876 с.

Ю.В. Устимчук (УО «ГГУ им. Ф. Скорины», Гомель)
Науч. рук. **Е.М. Березовская**, канд. физ.-мат. наук, доцент

ПАРАМЕТРИЗОВАННЫЕ КОМПОНЕНТЫ В РОДОВОМ ПРОГРАММИРОВАНИИ

Суть родового программирования, которое обязано своему распространению стандартной библиотеке шаблонов C++, состоящей из структур-контейнеров и алгоритмов, в частности, поиска и сортировки, заключается в том, что оно позволяет представлять предметные области в виде коллекций абстрактных компонентов очень общего характера, из многочисленных сочетаний которых производятся крайне эффективные конкретные программы. Поддержка родового программирования со стороны объектно-ориентированных средств моделирования и программирования довольно скудна. Большинство инструментов просмотра, существующих в рамках интегрированных объектно-ориентированных сред разработки, основываются на иерархиях наследования и с точки зрения организации родовых библиотек оказываются практически бесполезными.

В работе рассматривались принципы и методики родового программирования: параметризация типов, понятие параметризованных компонентов, принципы родового анализа и проектирования. Родовые параметры помогают проводить частичную параметризацию алгоритмов – например, параметризацию стратегии сравнения в подпрограммах сортировки. Для того чтобы параметризовать стратегию сравнения, можем инкапсулировать ее в форме объекта и передать подпрограмме сортировки

в качестве дополнительного параметра – *объекта функции*. Возможности объекта функции на C++ обычно реализуются средствами операции «скобки», которая позволяет вызывать его аналогично функции. Объекты функций оказываются особенно полезными в тех случаях, когда стоит задача накопить или сохранить состояние между вызовами функций.

Метод разработки любых библиотек параметризованных компонентов представляет собой метод *параметризованного программирования*. Одним из образцов параметризованного программирования является стандартная библиотека шаблонов C++.

Рассмотренные концепции были проиллюстрированы на примерах пузырьковой сортировки, проводящейся в параметризованном порядке с использованием разных стратегий на языке C++.

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

Науч. рук. **В.Д. Левчук**, канд. тех. наук, доцент

АВТОМАТИЗАЦИЯ УЧЕТА МАТЕРИАЛОВ НА СКЛАДЕ ДЛЯ ВИТЕБСКОГО ФИЛИАЛА ПРЕДПРИЯТИЯ «БЕЛВТОРМЕТ»

Одним из важных факторов в автоматизации предприятия является автоматизация учета материалов на складе. На предприятии документы являются первичным носителем информации об остатках и оборотах материальных средств. Но очень сложно в процессе работы ответить, где находится товар в определенный момент времени, сколько его продано, на какую сумму, какую прибыль получило предприятие. Просмотр бумажных документов затрачивает значительное количество времени. Для этого организации необходим автоматизированный учет, что бы информация была пригодна для оперативного анализа.

Для автоматизации учета материалов на складе для витебского филиала предприятия «БелВторМет» используется программа 1С Предприятие 8.2. Она позволяет организовать эффективный бухгалтерский, кадровый, оперативный, торговый, складской и производственный учет, а также расчет заработной платы.

Складской учет товаров на складе необходим для контроля за поступлением, хранением и выбытием товаров и их оплатой, а также для контроля за поступлением, использованием. В процессе учета учитывается поступление, движение внутри склада и выбытие за пределы склада товаров в натуральном и стоимостном выражении, с использованием данных товарных приходных и расходных документов. Любое движение товара строго документируется.

Отпуск продукции осуществляется по накладным с указанием получателя, даты отгрузки, наименования (артикула), количества и стоимости. При выявлении бракованных товаров составляется акт на списание. Складские документы направляются в бухгалтерию, где производится их проверка и постановка на учет, либо списание с учета (для расходных документов).

Учетом хранения товаров занимаются материально ответственные лица (заведующий складом, кладовщик).

Для автоматизации используются следующие документы:

– поступление материальных ценностей – в это документе ведется учет о поступлении товара на предприятие, его количестве, приведены характеристики товара, а так же указываются ответственные лица;

– реализация материальных ценностей – учет по реализации товара предприятием, указывается количество реализованного товара, сумма реализации, с НДС и без НДС, а так же ответственное лицо за реализацию товара;

– перемещение материальных ценностей – перемещение материалов из склада в склад, указывается товар который необходимо переместить и склад с которого перемещается товар и на который необходимо его переместить;

– списание материальных ценностей – списание материалов со склада, указывается материал который списывает, его количество и ответственное лицо;

– комплектация ТМЦ – информация о комплектации материалов;

– возврат МЦ поставщику – возврат материалов поставщику.

Для того, чтобы была возможность контролировать материалы используются регистры. Данные в регистры добавляются только при проведении документов. Сведения из регистров используются для формирования отчетов.

Пользователь может получить информацию о состоянии складских запасов, заказов, продаж, взаиморасчетов в любых аналитических разрезах с требуемой детализацией. Например, можно получить данные о продажах по покупателям и позициям номенклатуры с разбивкой по месяцам. Пользователь может самостоятельно задавать уровень детализации, параметры группировки и критерии отбора данных в отчетах в соответствии со спецификой решаемых задач. Такие индивидуальные настройки могут быть сохранены для повторного использования. Так же отчетные данные можно получить в наглядной графической форме.

Для автоматизации были разработаны следующие отчеты: ведомость по материальным ценностям на складах, карточка складского учета номенклатуры, группа отчетов по готовой продукции, приход

материальных ценностей, расход материальных ценностей, остатки материальных ценностей, реализация материальных ценностей, списание материальных ценностей.

Литература

1. Радченко, М.Г. 1С: Предприятие 8.2. Практическое пособие разработчика / М.Г. Радченко, Е.Ю. Хрусталёва. – М.: ООО «1С – Паблишинг», 2009. – 872 с.

2. Тимофеев, Г.С. Конфигурирование и администрирование 1С: Предприятие / Г.С. Тимофеев, Д.С. Шумейко. – М: Феникс, 2003. – 320 с.

Д.Е. Храбров (УО «ГГТУ им. П.О. Сухого», Гомель)

Науч. рук. **И.А. Мурашко**, д-р. тех. наук, доцент

МЕТОДИКИ ЛОКАЛЬНОГО ПОЗИЦИОНИРОВАНИЯ В ПОМЕЩЕНИИ УНИВЕРСИТЕТА НА ОСНОВАНИИ СУЩЕСТВУЮЩЕЙ WiFi-СЕТИ

Предлагается методика позиционирования объекта внутри организаций на основе стандарта WiFi. Ведь GPS навигация не подходит для многоэтажных зданий, а значит необходимо использовать другую технологию. В то же время, большое количество организаций для предоставления доступа в интернет используют беспроводные WiFi-сети.

Административные здания как правило содержат большое количество WiFi точек доступа, как стационарных, так и нет. Мобильные устройства, такие как ноутбуки, планшеты, мобильные телефоны, постоянно перемещаются и не могут быть использованы для WiFi позиционирования ввиду того, что точные координаты такой точки в каждый момент времени не известны.

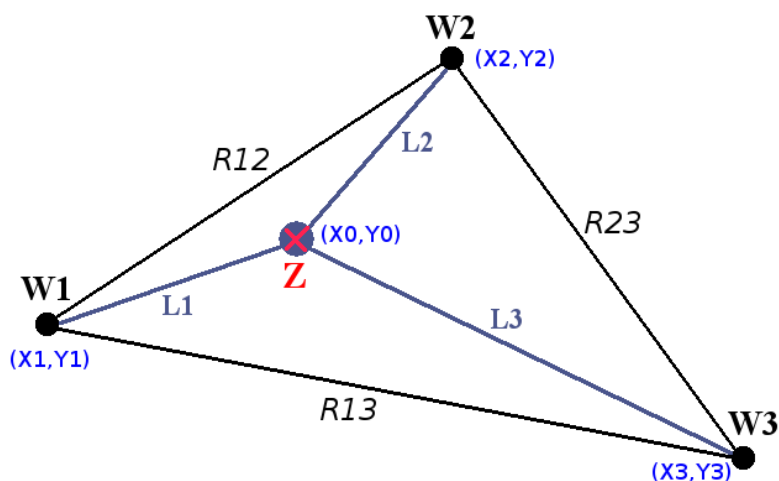


Рисунок 1 – Позиционирование по трём точкам доступа

На рисунке 1 показана ситуация, когда устройство Z получает сигналы от 3 точек доступа: $W1, W2$ и $W3$. Устройство имеет координаты $(X0, Y0)$, точки доступа соответственно: $(X1, Y1)$, $(X2, Y2)$, и $(X3, Y3)$. Расстояния от Z до каждой точки доступа соответственно: $L1, L2$ и $L3$.

Объект, снабжённый устройством позиционирования, каждый раз снимает уровни сигнала доступных точек, которые занесены в список разрешённых. Для позиционирования было бы достаточно знать координаты точек и расстояния от устройства до каждой точки доступа (L). Однако значение L не известно – обычно известен уровень сигнала, который проблематично с достаточной точностью перевести в меру расстояния.

Координаты точки Z получаются с помощью алгоритма взвешенного центроида (Weighted centroid)[1]. Алгоритм "Центроид" (Centroid) описан в [2] и представляет собой вычисление геометрического центра плоской фигуры, образованной несколькими точками доступа. В таком случае, координаты агента вычисляются как среднее арифметическое координат точек доступа. Алгоритм взвешенного центроида отличается тем, что у каждой вершины геометрической фигуры есть свой вес. В таком случае координаты определяются по формуле: $X_0 = \sum_i^N \mu_i X_i$, $Y_0 = \sum_i^N \mu_i Y_i$, где μ_i – характеристика веса.

Характеристику веса же можно вычислить зная P_i – уровни сигнала до каждой конкретной точки доступа: $\mu_i = (P_i^2 / \sum_j^N (1/P_j^2))^{-1}$.

По формулам видно, что каждая из координат рассчитывается не зависимо от других. Следовательно, возможно простое масштабирование для N -мерного пространства, в частности, трёхмерного. Так же видно, что абсолютные значения уровня сигнала так же не важны, так как при вычислении этот параметр приводится к долям единицы.

Пример отображения позиционирования на мобильном устройстве можно увидеть на рисунке 2.

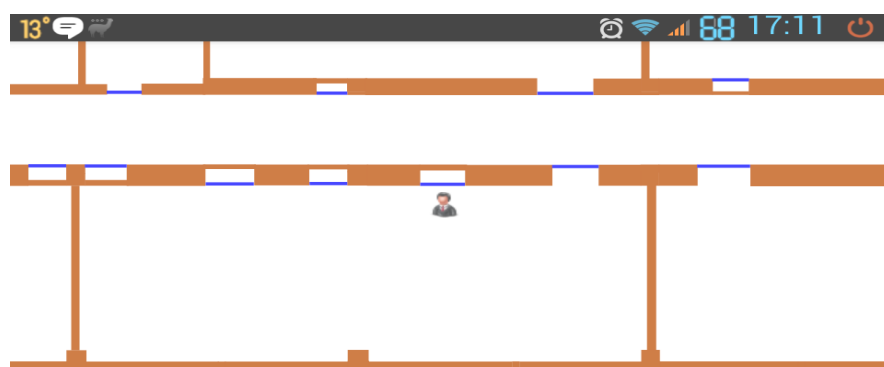


Рисунок 2 – Отображение позиционирования

Обычно здания состоят из помещений поменьше и высокая точность позиционирования не важна – достаточно знать помещение, в котором находится человек. Этого достаточно для надобностей статистики:

находится студент на паре, в медпункте или в курилке. Чтобы не перегружать сеть программа вычисляет координаты не постоянно, а раз в 5 минут.

Как правило в помещениях вдоль стен располагаются объекты такие как шкаф, стол и т.д. Это делает маловероятным позиционирование внутри стены, а не внутри помещения. В случае попадания точки ± 0.5 метра от стены можно повторно выполнить запрос. Для движущихся точек использовать отдельный алгоритм.

Для студента во время перерыва точность будет понижаться, так как большое количество студентов находятся в постоянном передвижении. Статистические данные лучше собирать во время занятий.

Предлагается клиент-серверная архитектура проекта. В этом случае часть нагрузки выполняет пользовательская, клиентская часть, а часть нагрузки ложится на сервер. Предполагается возможность аппаратной реализации клиента, так что его функционал должен быть минимальным чтобы уменьшить конечную стоимость и увеличить автономность. Также следует помнить, что данные с клиентов необходимо передавать на сервер. Уже при наличии 5000 клиентов при одновременной отправке данных на сервер возможен перегруз сети и отказ от обслуживания. Так что необходимо как минимум разделить отсылки по времени. Каждому студенту выставляется в соответствие уникальный номер. Например, уникальный номер может состоять из 2 частей: 1) год поступления 2) порядковый номер в каком-либо глобальном списке (студенты всех факультетов, отсортированные или по фамилии, или по баллам за вступительное тестирование, или ещё по какому-нибудь критерию). И если опрашивать клиентов в соответствии с их уникальным номером, то нагрузка на сеть будет снижена. Недостатком такого разбиения является неизвестность того, когда будут обработаны все студенты того или иного факультета.

Разработано программное приложение для позиционирования внутри помещения, основанное на сети WiFi и позволяющее: работать при отсутствии GPS сигнала, поверх существующей WiFi-сети, вести статистику распределения контролируемых объектов по помещению организации, то есть осуществлять контроль посещения занятий студентами). В качестве недостатков можно отметить необходимость доступа в сеть организации и достаточно низкая точность. В данный момент ведётся работа по устранению недостатков.

Литература

1. Bahl, P. Radar: An in-building RF-based user location and tracking system / P. Bahl, V. Padmanabhan // IEEE INFOCOM, Tel-Aviv, Israel. – Mar. 2000. – P. 775–784.

2. Kolodziej, K.W. Local positioning systems: LBS applications and services / K.W. Kolodziej, J. Hjelm // CRC Press. – 2006. – 445 p.

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

Науч. рук. **П.Л. Чечет**, канд. тех. наук, доцент

ПРОГРАММНЫЙ СПОСОБ ГЕНЕРАЦИИ ЛАБИРИНТОВ ЭЛЛЕРА

С помощью алгоритма Эллера можно создавать лабиринты, которые имеют только один путь между двумя точками. Данный алгоритм обладает высокой скоростью по сравнению с другими алгоритмами генерации лабиринтов. Требования по памяти прямо пропорциональны количеству строк лабиринта.

В следующем примере будет создан простой лабиринт размерностью 10*3. Он будет создаваться построчно, двигаясь справа налево, сверху вниз. Все ячейки будут принадлежать множествам. Принадлежность множествам визуализирована с помощью цифр, которые указывают на то, к какому множеству относится ячейка. Каждая ячейка может иметь границу слева и/или справа. Так же будет введено предположение, что, каждая первая ячейка строки обладает границей слева, а каждая последняя – границей справа.

1 Создается первая, пустая строка.

2 Каждую ячейку, которая не присоединена к множеству, следует к нему присоединить. Результат второго шага показан на рисунке 1.

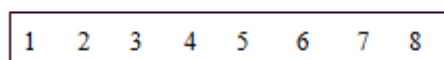


Рисунок 1 – Результат выполнения второго шага

3 Случайным образом создается граница справа. Если для ячейки не создана граница справа, то она объединяется в одно подмножество со следующей ячейкой. Результат на рисунке 2.

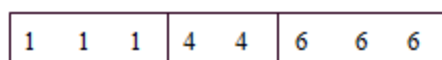


Рисунок 2 – Результат выполнения третьего шага

4 Таким же образом случайно создаются нижние границы. Следует выполнить одно условие: у каждого множества хотя бы одна ячейка не должна иметь границы снизу, в противном случае будет получена замкнутая область. Результат показан на рисунке 3.

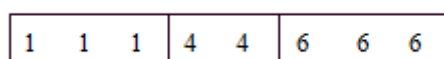


Рисунок 3 – Результат выполнения четвертого шага

4А Если требуется добавить еще одну строку, выполняется шаг 5А, иначе – шаг 6.

5А На данном этапе осуществляется создание новой строки. Для этого следует скопировать предыдущую строку на место текущей, удалить все правые границы. Далее следует осуществить удаление тех ячеек из множеств, которые имеют нижнюю границу. В конце происходит заполнение пустых клеток (присваивание множествам соответствующих ячеек) с соответствующими номерами, то есть на третью позицию третий номер, если четвёртая ячейка относится к множеству, то она пропускается. Результат выполнения данного шага показан на рисунке 4.

1	1	1	4	4	6	6	6
1			4		6	6	

Рисунок 4 – Результат выполнения шага 5А

5Б(2) Далее цикл повторяется со второго шага с единственным условием, указывающим на обязательное добавление правой границы между ячейками одного множества.

6 Последний шаг. Сейчас строится нижняя строка, отличная от остальных наличием у каждой ячейки нижней границы. Для генерации этой строки требуется выполнить следующее требование: все ячейки должны принадлежать одному множеству. Для этого следует удалять границы между ячейками различных множеств. Не нужно удалять границу между ячейками одного множества. Результат показан на рисунке 5.

1	1	3	3	7	7	8	8
1	1	1	1	1	1	1	1

Рисунок 5 – Результат выполнения шага 6

Реализация данного алгоритма генерации лабиринта была написана на языке программирования *JavaScript* с использованием тега *canvas* из *html5*. Данные технологии были выбраны в основном из-за того, что *JavaScript* позволяет очень просто генерировать графическое изображение на *canvas*, так как почти все остальные языки, для работы с графикой подразумевают подключение сторонних модулей.

HTML-код состоит из стандартных тегов (<html>, <body>, ...) и тега *canvas*, на котором и «рисуются» лабиринт:

```
<canvas id="canvas" style="background:#eeeeee"></canvas>
```

JavaScript-код состоит из анонимной функции, в которую передаётся три параметра:

- 1 id (идентификатор) тега `canvas`, на котором генерируется лабиринт;
- 2 высота лабиринта;
- 3 ширина лабиринта;

Результаты генерации лабиринта показаны на рисунках 6–8. На рисунке 6 показан сгенерированный лабиринт 10×10 .

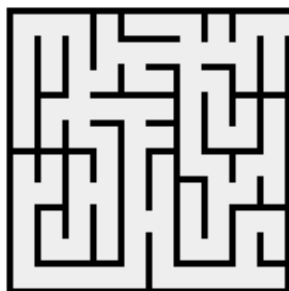


Рисунок 6 – Лабиринт 10×10

На рисунках 7 и 8 показаны граничные случаи, а именно: лабиринты 5×1 и 1×1 .



Рисунок 7 – Лабиринт 5×1



Рисунок 8 – Лабиринт 1×1

В итоге, с помощью данного алгоритма удалось получить «идеальный» лабиринт, в котором отсутствуют циклы (между двумя ячейками есть только один путь) и изолированные части (ячейки или групп ячеек, которые не связаны с другими частями лабиринта).

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

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

КРАТКОЕ ОПИСАНИЕ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

Регулярные выражения представляют своего рода шаблон, который позволяет упростить и ускорить поиск подстроки в строке. Регулярные выражения состоят из девяти групп элементов. Туда входят–якоря, символные классы, POSIX, утверждения, кванторы, специальные символы, группы и диапазоны, метасимволы.

1 Якоря позволяют указать на границу, то есть начало или конец строки, символа, слова.

`^` Начало строки

`\A` Начало текста

`$` Конец строки

`\Z` Конец текста

`\b` Граница слова

`\B` Не граница слова

2 Символьные классы отвечают за некоторый набор символов.

`\c` Управляющий символ

`\s` Пробел

`\S` Не пробел

`\d` Цифра

`\D` Не цифра

`\w` Слово

`\W` Не слово

`\xhh` Шестнадцатиричный символ `hh`

`\Oxxx` Восьмиричный символ `xxx`

3 POSIX представляет собой относительно новый элемент регулярных выражений, который очень схож с символьными классами и также позволяет указывать на группу элементов. Например для поиска букв в верхнем регистре следует писать следующее:

`[:upper:]` Буквы в верхнем регистре

4 С помощью утверждения мы можем сказать: «я хочу найти в этом документе каждое слово, включающее букву “q”, за которой не следует “werty”». Пример шаблона, который реализует это высказывание:

`[^\s]*q(?!werty)[^\s]*`

С самого начала происходит поиск всех элементов кроме пробела (`[^\s]*`), за которыми следует `q`. Далее парсер переходит к «смотрящего вперед» утверждению (`(?!werty)`). После перехода к утверждению, предшествующий элемент (символ, строка, слово) становится условным, то есть он будет соответствовать шаблону, только если утверждение верное. Так как в разбираемом примере утверждение является отрицательным (`?!`), то оно будет верным, тогда, когда то, что в нем ищется, не будет найдено. То есть при отсутствии «werty» в слове, будет произведено сравнение первого элемента с «q». Если же «werty» не было найдено, то утверждение верно, и можно начинать проверку на соответствие «q». Если «q» найдено, то продолжается поиск любых символов, кроме пробела (`[^\s]*`).

5 Смысл квантором лучше объяснить на примере. Если требуется выяснить, находится ли в документе строка в которой присутствуют от 10 до 20 (включительно) букв «a», то следует использовать этот шаблон:

a{10,20}

Кванторы подразделяются на «жадные» и «ленивые». Разницу между ними стоит так же рассмотреть на примере. Например следующий шаблон, является «жадным»:

".*"

Этот шаблон выводит текст заключенный в двойные кавычки. Но возьмем в качестве примера следующую строку:

```
<a href="index.html" title="Главная">Главная</a>
```

В результате выполнения приведенного выше шаблона мы получим следующую подстроку. По которой видно, что шаблон взял максимальную подстроку, которая находится между кавычек.

```
"index.html" title="Главная"
```

Следующий шаблон является «ленивым»

".*?"

Он ищет наименьшее из возможных вхождений, и поэтому в качестве результата мы получим следующие две подстроки:

```
"index.html"
```

```
"Главная"
```

6 К специальным символам относятся следующие символы:

\ Экранирующий символ

\n Новая строка

\r Возврат каретки

\t Табуляция

\v Вертикальная табуляция

\f Новая страница

\a Звуковой сигнал

[b] Возврат на один символ

\e Escape-символ

\N{name} Именованный символ

7 Диапазоны, как понятно из названия указывают на некоторые наборы и диапазоны символов.

8 Группы в основном применяются для решения трех следующих задач: когда требуется подстановка строк, для ссылки на другую часть шаблона и когда в шаблоне необходимо реализовать условие «или».

. Любой символ, кроме переноса строки (\n)

(a|b) a или b

(...) Группа

(?:...) Пассивная группа

[abc] Диапазон (a или b или c)

[^abc] Не a, не b и не c

[a-q] Буква между a и q

[A-Q] Буква в верхнем регистре между A и Q

[0-7] Цифра между 0 и 7

\n n-ая группа/подшаблон

9 В последнюю группу входят метасимволы. Метасимволы – это символы, которые имеют специальное значение. Например метасимвол «^» указывает на начало строки, а метасимвол «\$» на ее конец. Если какой-нибудь из этих символов требуется использовать как обычный, то его надо будет экранировать. Ниже приведены все метасимволы.

`^ [. $ { * (\ +) | ? < >`

Можно с уверенностью сказать, что регулярные выражения являются очень мощным и достаточно простым средством для поиска подстроки в строке и в умелых руках могут многократно упростить решение соответствующих задач.

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

Науч. рук. **Г.Л. Карасёва**, канд. физ.-мат. наук, доцент

СИСТЕМА УПРАВЛЕНИЯ САЙТАМИ ПРЕДПРИЯТИЯ

В условиях современного информационного общества все крупные предприятия нуждаются в создании и сопровождении своих сайтов в сети Интернет. При этом, чем больше предприятие, тем в большем количестве и качестве сайтов оно нуждается. Создание такой сети сайтов с учетом разнообразных требований под каждый регион либо область деятельности требует усилий со стороны большой команды разработчиков и стоит больших денег. Кроме того, каждый сайт нуждается в наполнении содержимым: это какая-либо статическая информация о деятельности компании, новости и события. Как правило, наполнением сайтов занимаются люди малознакомые с программированием, что означает, что реализация добавления информации на сайт должна быть максимально простой и понятной.

Вышеизложенные проблемы позволяет решить использование AdobeExperienceManager, написанной на Java, что дает возможность использовать все достоинства этого языка программирования, в качестве системы управления содержимым сайта. В состав АЕМ входят такие компоненты, как CRXContentRepository, реализующий JCR, который предоставляет возможность быстро и без труда создавать и изменять ноды с наполнением и получать их как на сервере, так и на клиентской стороне, используя RESTAPI, и ApacheSling, котрый позволяет быстро и удобно разрабатывать компоненты серверной части приложения. В CRX

репозитории, помимо JSP-страниц, файлов дизайна и скриптов, хранится так же наполнение сайта: статическая информация и изображения. Существует несколько способов изменения этой информации. Контент-менеджеры имеют возможность вносить изменения прямо на страничке, видя изменения «на лету», что существенно упрощает им работу с сайтами. При этом, Adobe Experience Manager позволяет хранить и редактировать не один сайт, но несколько, которые могут различаться языком, дизайном и набором используемых модулей.

Adobe Experience Manager имеет огромные возможности по масштабированию. Существует два варианта установки системы: Author и Publish. На Author-сервере происходит редактирование всего содержимого, от статического текста до стилей и скриптов. При этом посетители сайта получают информацию с Publish-серверов, на которых эти изменения не видны, пока не будут реплицированы. Перед набором Publish-серверов устанавливается балансировщик нагрузки, который распределяет запросы по разным серверам, а так же кеширует статическую информацию, что позволяет значительно снизить нагрузку на сервера с АЕМ. При этом установка или обновление модулей серверной части системы, благодаря возможности использования Maven, практически не представляет сложности, а обновления не вносят перебои в работу сайтов.

Таким образом, все приведенные возможности Adobe Experience Manager в сумме позволяют значительно сократить трудозатраты по разработке сайтов уровня предприятия со стороны разработчиков, а его сопровождением фактически сможет заниматься один-два человека.

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

Науч. рук. П.Л. Чечет, канд. тех. наук, доцент

АВТОМАТИЗАЦИЯ СБОРКИ С ИСПОЛЬЗОВАНИЕМ УТИЛИТЫ MAKE

Автоматизация сборки – этап написания скриптов (кратких описаний действий, выполняемых системой) или автоматизация широкого спектра задач применительно к ПО, применяемому разработчиками в их повседневной деятельности, включая такие действия, как:

- компиляция исходного кода в бинарный код;
- сборка бинарного кода;
- выполнение тестов;
- разворачивание программы на производственной платформе;

– написание сопроводительной документации или описание изменений новой версии.

Так же в последнее время возникло понятие продвинутой автоматизации сборки. Продвинутой автоматизацией сборки предоставляет возможность удаленному пользователю управлять обработкой распределенных сборок и/или распределенной обработкой сборки. Термин «распределенные сборки» подразумевает, что вызовы компилятора и линковщика могут передаваться множеству компьютеров для ускорения скорости сборки. Данный термин часто путают с «распределенной обработкой». Распределенная обработка означает, что каждый этап процесса может быть адресован разным машинам для выполнения ими данного шага. Например, этап после сборки может потребовать выполнения множества тестовых скриптов на множестве машин. Распределенная обработка позволяет послать команду на исполнение различных тестовых скриптов на разных машинах. Распределенная обработка – не то же самое, что и распределенная сборка. Распределенная обработка не может взять скрипты от *make* или *maven*, разбить их и послать команды на компиляцию и линковку различным машинам. Распределенный процесс сборки должен обладать определенной логикой, чтобы правильно определить зависимости в исходном коде для того чтобы выполнить этапы компиляции и линковки на разных машинах. Решение автоматизации сборки должно быть способно управлять этими зависимостями, чтобы выполнять распределенные сборки. Не все инструменты автоматизации сборки могут выполнять распределенные сборки. Большинство из них лишь реализует поддержку распределенной обработки. Кроме того, большинство решений, поддерживающих распределенные сборки, могут лишь обрабатывать код на языках Си и С++. Решения автоматизации сборки, поддерживающие распределенную обработку, зачастую основаны на *make* и не поддерживают *maven* или *ant*.

Автоматизацию можно условно разделить на несколько типов:

- автоматизация по запросу (On-Demand automation): запуск пользователем скрипта в командной строке;
- запланированная автоматизация (Scheduled automation): непрерывная интеграция, происходящая в виде ночных сборок;
- условная автоматизация (Triggered automation): непрерывная интеграция, выполняющая сборку при каждом подтверждении изменения кода (commit) в системе управления версиями.

Одна из особых форм автоматизации сборки – автоматическое создание make-файлов (makefiles). *Make* – утилита, автоматизирующая процесс преобразования файлов из одной формы в другую. Чаще всего это компиляция исходного кода в объектные файлы и последующая

компоновка в исполняемые файлы или библиотеки. Утилита использует специальные *make*-файлы, в которых указаны зависимости файлов друг от друга и правила для их удовлетворения. На основе информации о времени последнего изменения каждого файла *make* определяет и запускает необходимые программы.

Существует несколько версий *make*, основанных на оригинальной утилите *make*, или написанных с нуля, использующих те же самые форматы файлов и базовые принципы и алгоритмы, а также содержащие некоторые улучшения и расширения. Например *BSDmake*, основанная на работе Адама де Бура (Adam de Voog) над версией *make*, с возможностью параллельной сборки; в той или иной форме перешла в *FreeBSD*, *NetBSD* и *OpenBSD*, или *GNUmake* – входит в большинство дистрибутивов *GNU/Linux* и часто используется в сочетании с *GNUbuildsystem*.

Программа *make* выполняет команды согласно правилам, указанным в специальном файле. Этот файл называется *make*-файл (*makefile*, мейк-файл). Как правило, *make*-файл описывает, каким образом нужно компилировать и компоновать программу.

Литература

1. Автоматизация сборки / Википедия [Электронный ресурс]. – 2013. – Режим доступа: http://ru.wikipedia.org/wiki/Автоматизация_сборки – Дата доступа: 13.02.2013.
2. nant.sourceforge.net [Электронный ресурс]. – 2013. – Режим доступа: <http://nant.sourceforge.net/release/latest/help/introduction> – Дата доступа: 13.02.2013.
3. Утилита *make* / Википедия [Электронный ресурс]. – 2013. – Режим доступа: <http://ru.wikipedia.org/wiki/Make> – Дата доступа: 13.02.2013.

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

Науч. рук. П.Л. Чечет, канд. тех. наук, доцент

АВТОМАТИЗАЦИЯ СБОРКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Рост объёма компилируемого исходного кода усложняет сборку современных проектов. Одним из направлений, упрощающих эту процедуру, является автоматизация сборки программного обеспечения.

Автоматизация сборки – этап написания определённых скриптов, выполняемых системой, или автоматизация широкого спектра задач применительно к программному обеспечению, применяемому разработчиками в их повседневной деятельности.

Так же в последнее время возникло понятие продвинутой автоматизации сборки. Продвинутая автоматизация сборки предоставляет возможность удаленному пользователю управлять обработкой распределенных сборок и/или распределенной обработкой сборки. Термин «распределенные сборки» подразумевает, что вызовы компилятора и линковщика могут передаваться множеству компьютеров для ускорения скорости сборки. Распределенная обработка позволяет послать команду на исполнение различных тестовых скриптов на разных машинах.

Решения автоматизации сборки, поддерживающие распределенную обработку, зачастую основаны на утилите *make*. До создания *make* системы сборки (компиляции) ПО Unix обычно состояли из *shell*-скриптов сборки, сопровождавших исходный код программ.

Make была создана Стюартом Фельдманом в 1977 году в *BellLabs*.

В настоящее время существует множество утилит для отслеживания зависимостей, но *make* – одна из самых широко распространённых, в первую очередь благодаря тому, что она включена в Unix, начиная с версии PWB/UNIX, которая содержала инструменты для разработки программного обеспечения.

Программа *make* выполняет команды согласно правилам, указанным в специальном файле. Этот файл называется *make*-файл (*makefile*, мейк-файл). Как правило, *make*-файл описывает, каким образом нужно компилировать и компоновать программу.

Литература

1. Автоматизация сборки / Википедия [Электронный ресурс]. – 2013. – Режим доступа: http://ru.wikipedia.org/wiki/Автоматизация_сборки – Дата доступа: 13.02.2013.

2. Утилита *make* / Википедия [Электронный ресурс]. – 2013. – Режим доступа: <http://ru.wikipedia.org/wiki/Make> – Дата доступа: 13.02.2013.

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

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

АНАЛИЗ БИНАРНОГО ФОРМАТА ПРЕДСТАВЛЕНИЯ СТРУКТУРИРОВАННЫХ ДАННЫХ СВОР

ConciseBinaryObjectRepresentation (*сжатое бинарное представление объекта*) – формат данных, который был спроектирован таким образом, чтобы обеспечить максимально простой код реализации, формирования компактных выходных данных и возможность расширения формата без необходимости обмена информацией о версии.

Стандарт формата CBOR был официально анонсирован комитетом IETF в октябре 2013 года в новом документе RFC 7049, авторами которого являются Carsten Bormann и Paul Hoffman. Формат CBOR получил MIME-тип application/cbor.

На данный момент существуют сотни всевозможных бинарных форматов для представления структурированных данных, ряд которых стандартизирован, популярен и широко применяется (например, BER и DER для ASN.1, MessagePack и BSON). К формату было предъявлено семь важных требований, и, поскольку ни один из существующих форматов в полной мере не мог им удовлетворить, был создан новый.

Требования, предъявленные к новому бинарному формату:

1 Недвусмысленное кодирование большинства популярных в Интернете стандартов типов данных. Формат должен уметь кодировать базовые типы данных и структур, используя бинарное представление. При этом нет требования к уникальному кодированию для абсолютно всех типов данных, например, число 7 можно представить и как строку, и как целое.

2 Компактная реализация для кодировщика/декодировщика, которая позволяла бы создавать реализации, не требовательные к возможностям процессора и доступному объёму памяти. Формат должен использовать привычные машинные реализации форматов (например, для целых чисел или чисел с плавающей запятой – IEEE 754).

3 Отсутствие схемы описания. Также как JSON-формат самоописывает представленные данные (объект, массив, строка и т.д.). Отсутствие схемы описания позволяет создать простой и универсальный декодировщик.

4 Сериализация данных должна быть как можно более компактной, но не в ущерб простоте написания кодировщика/декодировщика. За верхнюю планку объёма закодированных данных принимается объём представления в JSON-формате.

5 Формат должен быть одинаково применим как в приложениях на ограниченных ресурсах, так и в приложениях, работающих с огромными объёмами данных. Реализация должна быть бережлива к CPU при кодировании и декодировании данных.

6 Формат должен поддерживать все существующие типы в JSON для конвертирования данных из и в JSON.

7 Формат должен быть расширяемым, и расширенный формат должен успешно декодироваться ранними версиями декодировщика. Предполагается, что формат будет использоваться десятилетиями и при этом сохранять обратную совместимость, таким образом, чтобы будущие версии стандарта формата могли нормально обрабатываться декодировщиком для ранних версий.

Спецификация CBOR

Чтобы понять насколько успешно формат решает поставленные задачи, и, чтобы иметь возможность сравнить его с другими форматами, необходимо рассмотреть процесс кодирования данных в формат CBOR.

Каждый элемент данных начинается с байта, который содержит описание основного типа (*major type*) данных и дополнительной информации (*additional info*). Основной тип занимает три старших бита и может иметь значения от 0 до 7. Дополнительная информация может получать значения от 0 до 31 (рисунок 1).

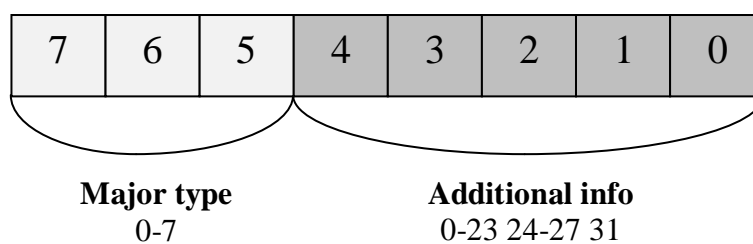


Рисунок 1 – Элемент данных

В значении дополнительной информации выделено три условных диапазона:

1. Значение от 0 до 23 используется непосредственно как целое число. Например, если основной тип – это целое число, то значение дополнительной информации – это и есть значение элемента. Если основной тип – это строка, то данное число указывает на длину строки.

2. Если значение от 24 до 27, то последующие байты содержат целое число переменной длины, соответственно 1-, 2-, 4- и 8-байтные неотрицательные целые. Например, если основной тип это целое число, то в последующих байтах содержится его значение, а если строка – то её длина.

3. Значение 31 является специальным и указывает на то, что длина элемента не определена. Например, в случае строки, это означает, что длина строки неизвестна.

Значения 28–30 зарезервированы для будущих версий стандарта.

В зависимости от значения основного типа разный смысл приобретает значение дополнительной информации.

Обработка ошибок

Стандарт CBOR допускает различный подход к обработке ошибок в формате. Ошибки могут быть связаны как с самим форматом (неизвестные значения в дополнительной информации, простых значений, некорректное число элементов массива и т.д.), так и со значениями (например, невалидный UTF-8 код в текстовой строке или дублирующиеся ключи в ассоциативном массиве).

Декодировщик может выдавать предупреждения на некорректных данных, может выполнять преобразования, чтобы исправить ошибку, может игнорировать ошибки или прекращать обработку сразу при обнаружении первой ошибки. Стандарт предлагает использование строгого (*strict*) режима, в случае, если данные пришли из ненадёжного источника, в этом случае декодировщик должен отклонять все данные, которые не прошли валидацию или могут быть по-разному восприняты разными декодировщиками. Нестрогий режим используется, если данные из надёжного источника и больше важна скорость обработки (например, не проверять валидность UTF-8 строк).

Существующие реализации CBOR

Сразу после выпуска стандарта стали появляться и первые реализации кодировщика/декодировщика различных языках программирования. Например, C, Perl, Ruby, Python, Java, Go и множество других.

Не все из этих реализаций полностью поддерживают стандарт и, возможно, существуют или появятся ещё много других реализаций. Всё это отлично подтверждает, что основная цель стандарта достигнута: реализация кодировщика/декодировщика – это задача небольшой сложности. Вполне вероятно, что скоро на собеседованиях программистов начнут просить реализовать нестрогий декодировщик CBOR в качестве тестового задания.

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

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

АНАЛИЗ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ВИРТУАЛИЗАЦИИ

В настоящее время все большую популярность приобретают технологии виртуализации. И это не случайно – вычислительные мощности компьютеров растут. В результате развития технологий, появляются шести-, восьми-, шестнадцатиядерные процессоры. Растет пропускная способность интерфейсов компьютеров, а также емкость и отзывчивость систем хранения данных. В результате возникает такая ситуация, что имея такие мощности на одном физическом сервере, можно перенести в виртуальную среду все серверы, функционирующие в организации. Это возможно сделать с помощью современной технологии виртуализации.

Технологии виртуализации становятся одним из ключевых компонентов современной IT-инфраструктуры крупных предприятий. Сейчас

сложно представить построение нового серверного узла компании без использования технологии виртуализации. Определяющими факторами такой популярности, несмотря на некоторые недостатки, можно считать экономию денег и времени, а также высокий уровень безопасности и обеспечение непрерывности бизнес-процессов.

Существует несколько способов осуществления виртуализации, с помощью которых достигаются одинаковые результаты через разные уровни абстракции. У каждого способа есть свои достоинства и недостатки, но каждый из них находит свое место в зависимости от области применения.

Можно считать, что самая сложная виртуализация обеспечивается *эмуляцией аппаратных средств*. В этом методе виртуальная машина аппаратных средств создается на хост-системе, чтобы эмулировать интересующее оборудование.

Другое использование эмуляции – это *эмуляция оборудования*, которая заключается в совместном развитии встроенного ПО и аппаратных средств. В этом методе виртуальная машина аппаратных средств создается на хост-системе, чтобы эмулировать интересующее оборудование.

Эмуляция оборудования использует виртуальную машину, чтобы моделировать необходимые аппаратные средства.

Вместо того чтобы дожидаться, когда реальные аппаратные средства будут в наличии, разработчики встроенного ПО могут использовать виртуальное оборудование для разработки и тестирования ПО.

Главная проблема при эмуляции аппаратных средств состоит в существенном замедлении выполнения программ в такой среде. Однако эмуляция аппаратных средств имеет существенные преимущества. Например, используя эмуляцию аппаратных средств, можно управлять неизменной ОС, предназначенной для PowerPC® на системе с ARM процессором. Также можно управлять многочисленными виртуальными машинами, каждая из которых будет моделировать другой процессор.

Полная (аппаратная) виртуализация, или «родная» виртуализация, является другим способом виртуализации. Эта модель использует менеджер виртуальных машин (гипервизор), который осуществляет связь между гостевой ОС и аппаратными средствами системы.

Взаимодействие между гостевой ОС и оборудованием осуществляется посредством гипервизора. Внутри гипервизора должна быть установлена и настроена определенная защита, потому что основные аппаратные средства не принадлежат ОС, а разделяются гипервизором. При построении крупных корпоративных систем, как правило, используется именно аппаратная виртуализация.

Паравиртуализация – этот способ имеет некоторые сходства с полной виртуализацией. Этот метод использует гипервизор для разделения доступа к основным аппаратным средствам, но объединяет код, касающийся виртуализации, непосредственно в ОС. Этот подход устраняет потребность в любой перекомпиляции или перехватывании, потому что сами ОС кооперируются в процессе виртуализации.

Паравиртуализация требует, чтобы гостевая ОС была изменена для гипервизора, и это является недостатком метода. Однако паравиртуализация предлагает высокую производительность, почти как у реальной системы. При этом, как и при полной виртуализации, одновременно могут поддерживаться различные ОС. Но определенным недостатком паравиртуализации можно считать ограниченное количество поддерживаемых ОС.

Виртуализация уровня операционной системы. Эта техника виртуализирует серверы непосредственно над ОС. Этот метод поддерживает единственную ОС и, в самом общем случае, просто изолирует независимые виртуальные серверы друг от друга. Для разделения ресурсов одного сервера между контейнерами, данная виртуализация требует внесения изменений в ядро ОС, но при этом преимуществом является родная производительность, без «накладных расходов» на виртуализацию устройств.

Преимущества использования виртуализации:

1. Сокращение затрат на приобретение и поддержку оборудования. В современных условиях практически в каждой компании всегда найдется один или два сервера имеющие несколько ролей, например, почтовый сервер, файловый сервер, сервер базы данных и т.д. Безусловно, на одной физической машине можно поднимать по несколько программных комплексов (серверов), выполняющих различные задачи. Но часто бывают ситуации, когда установка нового ПО требует независимой серверной единицы. В таком случае как раз и придет на выручку виртуальная машина с требуемой ОС. Сюда же можно отнести случаи, когда в сети необходимо иметь несколько независимых друг от друга виртуальных серверов со своим набором служб и своими характеристиками, которые должны существовать как независимые узлы сети. Типичный пример – это услуги VPS-хостинга.

2. Сокращение серверного парка. Преимущество виртуализации состоит в том, что можно значительно сократить количество физических ЭВМ. В результате меньше времени и денег тратится на поиск, закупку и замену оборудования. Наряду с этим сокращаются площади, выделяемые под содержание серверной базы.

3. Сокращение штата IT-сотрудников. На обслуживание меньшего количества физических ЭВМ требуется меньше людей. С точки зрения руководства компании, сокращение штата – это сокращение серьезной статьи расходов предприятия.

4. Простота в обслуживании. Добавить жесткий диск или расширить существующий, увеличить количество оперативной памяти, все это занимает определенное время в случае с физическим сервером. Отключение, отсоединение из стойки, подключение нового оборудования, включение – в случае использования виртуализации все эти действия опускаются, и операция сводится к нескольким щелчкам мыши или командам администратора.

5. Клонирование и резервирование. Еще одним плюсом виртуализации является простота клонирования виртуальных машин. Например, компания открывает новый офис. При этом серверная инфраструктура центрального офиса стандартизирована и представляет собой несколько серверов с одинаковыми настройками. Развертывание такой инфраструктуры сводится к простому копированию образов на сервер нового офиса, конфигурировании сетевого оборудования и изменению настроек в прикладном ПО.

Сейчас сложно представить себе IT-отрасль без виртуализации. Развитие информационных систем организаций тесно связано с применением технологий виртуализации. Данные технологии позволяют значительно сократить расходы, связанные с приобретением и обслуживанием серверных систем, сократить время на восстановление информации или развертывания аналогичных систем в новом оборудовании.

В.С. Шмидт (УО «ГГУ им Ф. Скорины», Гомель)
Науч. рук. **Д.С. Кузьменков**, канд. физ.-мат. наук, доцент

ВЗАИМОДЕЙСТВИЕ ПРИЛОЖЕНИЙ SAP R/3 И LOTUS NOTES

SAP R/3 (Systems, Applications and Products in Data Processing) – единая система, которая позволяет управлять всеми процессами, проходящими на предприятии. Благодаря этому программному продукту становится возможным в рамках одной системы получить управление бухгалтерским учетом, складами, логистикой и производством, а также персоналом [1]. Таким образом, этот продукт является мировым лидером среди программ для бизнеса.

IBM LotusNotes/Domino – программный продукт, платформа для автоматизации совместной деятельности рабочих групп. Данный продукт

смог включить в себя наиболее важные аспекты для работы предприятий [2,3]:

1. среда исполнения приложений автоматизации групповой деятельности;
2. криптозащита (шифрование и электронная подпись);
3. клиент электронной почты;
4. почтовый сервер;
5. персональный и групповой календари, планировщик задач;
6. набор офисных приложений IBM Lotus Symphony (текстовый редактор, электронные таблицы, подготовка презентаций);
7. клиент среды обмена мгновенными сообщениями (Instant messenger) IBM Sametime (сервер Sametime является самостоятельным продуктом);
8. веб-браузер;
9. веб-сервер;
10. сервер каталогов LDAP;
11. сервер приложений;
12. репликация – синхронизация между дистанционно удалёнными экземплярами баз данных;
13. службы интеграции данных DECS (Domino Enterprise connection services);
14. средство хранения вложенных файлов вне баз данных DAOS (Domino attachment and object services).

Организация взаимодействия между SAP и IBM Lotus Notes открывает новые горизонты использования этих приложений: для растущего сегмента компаний, использующих для связи и взаимодействия Lotus Notes, а для обработки бизнес-информации – систему SAP R/3, подобная интеграция дает очевидные преимущества. Благодаря упрощенному механизму управления бизнес решениями SAP нет нужды в дополнительном создании более сложных модулей, функций аналогов этому в Lotus, в тоже время высокий уровень безопасности характерный для Notes, обеспечивает защищённую работу с данными системы SAP. Для разработчиков и администраторов Lotus Notes и Domino, а также SAP R/3 существуют важные элементы: гибкость настройки Lotus Notes и всеобъемность элементов системы R/3. Они позволяют на клиентском уровне интегрировать Lotus Notes с SAP и дают возможность настраивать оба приложения под требования отдельных пользователей или компаний.

Для организации такого взаимодействия в среде Lotus Domino Designer были разработаны специальные страницы для визуального отображения полученных данных, называемые XPage. XPage – это технология для быстрой разработки мобильных и web-приложений. При-

ложения, созданные с помощью технологии XPages, работают в интернет-браузере и являются кроссплатформенными. Также были разработаны WEB-сервисы, для двух приложений. Эти web-сервисы решают задачу сохранения и поддержания типов данных двух приложений. Например, передача данных в реализованном web-сервисе со стороны SAP R/3 обеспечивается при помощи созданных структур входных и выходных параметров. Основную задачу обработки входных параметров выполняет функциональный модуль, в котором производится выборка данных из таблиц системы R/3, заполнение выходной структуры, либо сообщение об ошибке. Для доступа к WEB-сервису был создан прокси-сервер, который стал основой для создания логического порта подключения. Защита сервиса реализована через процедуру аутентификации, а именно, доступ осуществляется при помощи имени пользователя и пароля. Полученные данные принимаются WEB-сервисом со стороны Lotus и отображаются на XPage.

Литература

1. Вейс, В. Разработка приложений SAP R/3 на языке ABAP/4 + CD-ROM/ Р. Кречмер, В. Вейс. – М. : Лори, 1998. – 368 с.
2. Линд, Д. Lotus Notes и Domino 5/6. Энциклопедия программиста / Д. Линд, С. Керн; пер. с. англ. – 2-е изд., перераб. и доп. – К. : ООО «ТИД ДС», 2003. – 1024 с.
3. Speed, T. Lotus Notes Domino 8: Upgrader's Guide / T. Speed, D. McCarriick, B. Gibson. – Packt Publishing, 2008. – 276 p.

Содержание

Секция 3 «Автоматизация исследований»

Абрамов Е.С. Модель проектирования MVC.....	3
Абрамов Е.С. Автоматизация определения погрешности измерителей линейных размеров.....	5
Алексеев П.Н. АРМ для расчета и распределения учебной нагрузки на кафедре ВУЗа	6
Анашкин А.С. Ведение логов в приложениях на JAVA	7
Астапович Ю.М. Автоматизация расчета смет по проектно-обследовательским работам.....	8
Бакунов М.В. Разработка фреймворка Xetax engine для Corona SDK	11
Бендерская С.Я. Практические и теоретические вопросы по модернизации ЛВС на примере РУП «Гомельэнерго».....	12
Богданчук К.А. Анализ конструкций оптических головок CD-приводов и их применение в измерительных системах	15
Бокий М.Ю. Алгоритм отслеживания изменений данных в универсальном хранилище финансовой отчетности	18
Борихин И.А. О моделировании деятельности ИТ-специалистов..	21
Васильев О.А. Анализ результатов тестирования с применением графов	22
Васильев О.А. Визуализация процесса выполнения автоматических параллельных тестов	23
Вераксих А.Н. Использование машинного зрения в учебном процессе.....	24
Воробьева А.С. Проблемы распределения IP-адресов в сложных сетях.....	25
Ворочай С.В. Задачи введения на предприятии дополнительных рынков сбыта при помощи высоких технологий в Интернет	27
Гетиков Д.В. Разработка Web-ориентированной базы данных в корпоративной среде lotusnotes.....	31
Глухов Н.А. Решение оптимизационных задач	32
Годлевский А.О. Blink: новый браузерный движок для Chromium.....	33
Годлевский А.О. Использование языка Python для создания компьютерных игр.	35

Гордиевич А.В. Информационный сайт для «Брестгортранс» ...	36
Гореликов В.А. Использование технологии ALV для формирования отчетов в системе SAP R3.....	37
Гриневич М.И. Разработка и создание объектов конфигурации для предприятий торговли	38
Грудовенко А.В. Разработка журнала учета товарно-материальных ценностей для гмк «Хатынь».....	39
Громыко А.О. Виртуализация рабочих мест.....	40
Гузова В.В. Разработка сетевого приложения под операционную систему Android	42
Давиденко А.В. Разработка проекта оперативной визуализации ремонтных работ на трамвайных путях.....	43
Дементьев П.В. Организация вещания медиапотока в гетерогенной локальной сети мобильных устройств	44
Демиденко С.С. Проект автоматизации рабочего места сотрудника землеустроительной службы исполнительного комитета.....	47
Демиденко С.С. Особенности автоматизации рабочего места сотрудника землеустроительной службы исполнительного комитета..	48
Демченко К.С. Внешние обработки отчетов в среде 1С: Предприятие 8.2	51
Деревянко Д.В. Анализ шаблонов проектирования.....	53
Деревянко Д.В. Этапы разработки веб-портала на базе Ibm Portal Server.....	55
Диваков Н.Н. Исчерпание IPv4-адресов	57
Диваков Н.Н. Использование протокола DHCP	59
Диваков Н.Н. IPv6 протоколы.....	60
Дикун Н.Г. Система компоновки данных 1С:Предприятие 8.2 ...	62
Дробышевская О.В. Self-organizing incremental neural network – алгоритм обучения без учителя.....	63
Дробышевская О.В. Новые системы хранения Fujitsu Eternus DX.....	66
Жук И.С. Средства автоматизации документооборота для конференций	69
Жуковец Г.С. Средство представления учебно-методического материала в формате СНМ (Формат справки Windows)	70
Зайцев А.С. Нанопроводниковый аккумулятор.....	72
Заяц А.В. Имитационное моделирование как метод исследования сложных систем.....	74

Зинин В.А. Задачи ввода, хранения и обработки информации на предприятии	76
Зинчук Е.И. Решение задачи преобразования адресов NAT44 на интеллектуальных сетевых устройствах.....	78
Кириков С.А. Автоматизация деятельности ветеринарной лаборатории.....	80
Кириков С.А. Автоматизация учета промышленного оборудования, поддерживающего работу по протоколу TCP/IP V4	83
Клименков А.А. Роль лазерной трёхмерной печати в создании нового экспериментального материала.....	84
Климович М.А. Тестирование функциональных возможностей бизнес-приложения для управления веб-хостингом.....	86
Коваленко М.Н. Многокамерная видеоаналитика.....	87
Ковалев Д.П. Пространственное моделирование хранилища данных с использованием IBM Infosphere data architect	89
Кожемякин А.П. Проект модернизации ЛВС ЧУП «Гомельская универсальная база».....	92
Козлов И.А. Применение современных технологий в повседневной жизни	93
Козлов И.А. Сравнительный анализ наиболее популярных СУБД.....	96
Коледа Я.А. Функциональные возможности облачного сервиса для управления учебным семинаром	98
Коледа Я.А. Автоматизация управления внешним тренингом предприятия «Эпам Системз»	99
Колюцей В.В. Использование технологии создания гелиевых жестких дисков	102
Коляскин И.И. Технические аспекты игровой ситуации	105
Кононович А.И. Разработка WEB-сайта НИЛ «Физика волновых процессов»	106
Кончиц Е.М. Разработка краткого справочника по механике в системе Android.....	107
Копачев В.Н. Разработка сетевого приложения на языке Java с использованием библиотеки Swing	110
Корж К.В. Работа с пользователями в системе 1С:Предприятие 8..	111
Коржов И.В. Разработка Web-сайта для ЧТПУП «Медиа-мир».....	114

Котлярчук Ю.А. Совершенствование деятельности коммуналь- ных жилищных предприятий.....	117
Кравцов В.С. Анализ типовых топологий вычислительных сетей..	120
Кулаков Ю.В. Разработка проекта модернизации ЛВС для ЧТУП «Медиамир»	122
Кулешов А.С. Создание комплексного учебного приложения.....	123
Курник Н.В. Совершенствование процесса оплаты в 1С	125
Кучеренко Р.В. Анализ современных систем удаленного монито- ринга вычислительных ресурсов	129
Лавинецкий В.С. Разработка и программная реализация системы регистрации	131
Лапушкина А.П. Связь системы 1С:Предприятие 7.7 с другими программами	132
Лебедев Р.А. Создание игр с использованием MICROSOFT XNA.....	135
Леванцов С.В. Организация резервирования функции default gateway с помощью технологии HSRP	136
Леванцов С.В. Сравнительный анализ современных технологий построения высокоскоростных сетей	137
Левинская Е.С. Мультиплатформенный инструмент для разра- ботки игр Unity	138
Левшунов Е.В. Разработка функционала создания табелей учета рабочего времени в среде Lotus Notes 8.5	142
Левшунов Е.В. Описание основных свойств IBM Lotus Notes для интеграции Lotus Notes с календарями Google	144
Лутков М.В. Групповая разработка прикладных решений в системе 1С: Предприятие 8.2	145
Малиновский В.Е. Использование метода аналитических сетей в системах поддержки принятия решений	147
Малиновский В.Е. Методы и технологии системы поддержки принятия решений в области разработки программного обеспечения.	148
Мартынов А.П. Управление функцией освещения помещений с помощью мобильных устройств на платформе ANDROID.....	149
Масленченко Н.В. Разработка сайта «Заказ билетов»	152
Матвеевко А.И. Автоматизация расчетов сдельной заработной платы для предприятия КСУП «Тепличное».....	153

Матюшин Р.В. Разработка системы управления содержимым веб-сайта для железнодорожного (г. Гомеля) отдела департамента охраны	156
Мельникова А.В. Оптимизация динамических систем с фазовыми ограничениями	158
Мкртычян А.С. Ансамблевый гидродинамический прогноз погоды	160
Мойсеенко А.А. Значение компьютерных сетей в производстве ..	163
Мордвинов В.А. Разработка форума «Starpoint»	166
Моцар М.В. Создание цветовой схемы WEB-сайта	168
Никитин С.В. Разработка онлайн системы автоматизации обработки заявок для предприятия	171
Никитина О.В. Разработка сайта для предприятия в Joomla 3.....	174
Новик Д.А. Модернизация ЛВС для отдела образования, спорта и туризма железнодорожного района города Гомеля.....	175
Носиков А.А. Моделирование процесса разработки сайта	178
Осипова О.А. Автоматизация управления персоналом на предприятии	182
Пигуль С.С. Компьютерная графика в среде 3D Studio Max	183
Пинчук А.Н. Автоматизация дебиторской кредиторской задолженности для ОАО «Строительный трест №14».....	184
Пинязьков И.А. Разработка WEB-сайта автоматизации процесса купли-продажи учебных материалов	186
Пищик Е.В. Реализация функций учета труда в конфигурации 1С:Предприятие «Персонал. Кадры. Зарплата»	187
Пищик Е.В. Автоматизация построения отчетов в конфигурации 1С: Предприятие «Персонал. Кадры. Зарплата»	190
Плесский А.В. Разработка WEB-сервиса для ONLINE заказа билетов	191
Побыловский А.А. Применение алгоритма вычисления ОПН для разработки приложения «Инженерный калькулятор» под ОС Android	192
Поддубный С.А. Функциональные возможности сервиса учёта заказов в IT-компании.....	194
Поддубный С.А. Автоматизация учёта заказов сотрудников в IT-компании.....	195
Приходько М.М. Сервисы для подбора ключевых слов при поисковой оптимизации	198

Приходько М.М. Анализ важнейших элементов сайта, привлекающих пользователей и поисковые системы	199
Розинко Р.О. Автоматизация расчета заработной платы сотрудников КЖУП «Гомельский райжилкомхоз»	201
Руденя П.А. Использование архитектуры «модель-представление» при разработке Desktop-приложений	203
Русецкий А.А. Механизмы обеспечения совместимости протоколов IPv4 и IPv6	204
Сакович А.С. Разработка проекта модернизации ЛВС для гомельского интернет-провайдера «Гарант».....	207
Самусев А.А. Разработка сайта для интернет-торговли автозапчастями	209
Сапанович А.В. Программный комплекс для автоматического тестирования мобильных платежных приложений	212
Семенякин Д.Л. Выбор видеокамер для видеонаблюдения.....	215
Сердюков Д.А. Высокочастотные запросы	219
Скринникова Т.С. Модуль распознавания лиц	221
Слатвинский В.Ф. Разработка базы данных для приложения по автоматизации и учету оборудования на предприятии	223
Слесаренко К.И. Приемы и методы порождающего программирования	225
Слука А.А. Использование электронного пособия на тему «Корреляционно-регрессионный анализ в разных программных системах» в физике	226
Сосновский М.В. Криптографическая защита информации.....	229
Сосновский М.В. Основные понятия в криптографической защите информации	230
Стадник Т.Г. Задачи синхронизации в аспектно-ориентированном программировании	231
Старовойтов Н.В. Разработка проекта управления заказами для ООО «Центр технических решений»	232
Столярова В.А. Представление учебных материалов в формате SCORM.....	234
Тищенко Т.Ю. Обзор возможностей комплекса 1С: Предприятия с другими системами.....	236
Устимчук Ю.В. Параметризованные компоненты в родовом программировании	239

Хомицкий И.В. Автоматизация учета материалов на складе для Витебского филиала предприятия «Белватормет»	240
Храбров Д.Е. Методики локального позиционирования в помещении университета на основании существующей WiFi-сети.....	242
Чеботаревский А.С. Программный способ генерации лабиринтов Эллера	245
Чеботаревский А.С. Краткое описание регулярных выражений...	247
Шереметов И.О. Система управления сайтами предприятия	250
Шидловский А.С. Автоматизация сборки с использованием утилиты Make	251
Шидловский А.С. Автоматизация сборки программного обеспечения	253
Шинкарёва В.А. Анализ бинарного формата представления структурированных данных CBOR	254
Шинкарёва В.А. Анализ современных технологий виртуализации..	257
Шмидт В.С. Взаимодействие приложений SAP R/3 и Lotus Notes	260

Научное издание

**АКТУАЛЬНЫЕ ВОПРОСЫ
ФИЗИКИ И ТЕХНИКИ**

Материалы III Республиканской
научной конференции
студентов, магистрантов и аспирантов

(Гомель, 17 апреля 2014 года)

В двух частях

Часть 1

Подписано в печать 03.09.2014. Формат 60×84 1/16.
Бумага офсетная. Ризография. Усл. печ. л. 14,2 .
Уч.-изд. л. 15,5. Тираж 30 экз. Заказ 453.

Издатель и полиграфическое исполнение:
учреждение образования
«Гомельский государственный университет
имени Франциска Скорины».

Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 1/87 от 18.11.2013.
Специальное разрешение (лицензия) № 02330 / 450 от 18.12.2013.
Ул. Советская, 104, 246019, Гомель.