

## Содержание

	Стр.
Введение.....	6
1 Обзор вариантов решения задачи.....	8
1.1 Развернутая постановка задачи.....	8
1.2 Анализ аналогичных разработок.....	8
1.3 Выбор технологии программирования.....	10
1.4 Независимость от СУБД.....	12
1.5 Шаблон проектирования MVC.....	13
1.6 Выбор сервера приложений.....	16
1.7 Выбор инструментов для разработки.....	16
1.8 Веб-службы.....	17
1.9 Использование апплетов в работе системы.....	17
Выводы к разделу 1.....	18
2 Разработка структуры программного комплекса.....	19
2.1 Механизм объектно-реляционного отображения (ORM).....	19
2.2 Структура веб-службы.....	20
2.3 Обеспечение независимости от СУБД.....	21
2.4 Проектирование с помощью шаблона MVC.....	22
2.5 Схема работы портала.....	22
Выводы к разделу 2.....	23
3 Реализация программного комплекса.....	24
3.1 Реализация безопасности.....	24
3.1.1 Шифрование данных.....	24
3.1.2 Использование SSL.....	25
3.1.3 Сертификат для веб-сервера.....	26
3.2 Реализация веб-службы.....	27
3.3 Работа с почтой.....	28
3.4 Работа с журналом.....	29
3.5 Структура таблиц базы данных.....	30
3.6 Реализация уровня модели шаблона MVC.....	32
3.7 Реализация уровня контроллера шаблона MVC.....	40
3.8 Реализация уровня представления шаблона MVC.....	41
3.8.1 Интерфейс клиента.....	41
3.8.2 Интерфейс менеджера.....	42
3.8.3 Интерфейс администратора.....	43
3.8.4 Интерфейс разработчика.....	43
3.9 Вид работы программы.....	43
Выводы к разделу 3.....	47
4 Экономический раздел.....	48
4.1 Обоснование целесообразности разработки дипломной работы.....	48

4.2	Оценка затрат на разработку системы.....	48
4.3	Оценка эксплуатационных затрат.....	51
4.4	Оценка затрат на внедрение разработки.....	52
4.5	Оценка экономической эффективности разработки.....	53
	Выводы к разделу 4.....	54
5	Безопасность жизнедеятельности.....	55
5.1	Общие вопросы безопасности труда.....	55
5.1.1	Лица и их обязанности по вопросам безопасности труда.....	55
5.1.2	Обучение работающих безопасности труда.....	56
5.1.3	Планирование и финансирование мероприятий по охране труда.....	57
5.2	Обеспечение комфортных и безопасных условий труда.....	60
	программиста.....	60
5.2.1	Характеристика условий труда программиста.....	60
5.2.2	Организация рабочего места программиста.....	60
5.2.3	Обеспечение микроклиматических параметров.....	63
5.2.4	Организация освещенности.....	64
5.2.5	Защита персонала ИВО от шума.....	65
5.2.6	Защита персонала ИВО от опасных и вредных излучений.....	68
5.2.7	Обеспечение электробезопасности.....	69
5.2.8	Обеспечение пожарной безопасности.....	70
	Выводы к разделу 5.....	71
	Заключение.....	72
	Список использованной литературы.....	74
	Приложение А Листинги программы .....	75

## Введение

Данная дипломная работа представляет собой взаимосвязь таких важных понятий как электронная коммерция и электронный бизнес. Процесс их объединения с современными технологиями является главной задачей, поставленной перед автором.

Электронная коммерция – термин, используемый для обозначения коммерческой активности в сети Интернет. Обеспечивает возможность осуществления покупок, продаж, сервисного обслуживания, проведения маркетинговых мероприятий путём использования компьютерных сетей. Электронная коммерция подразделяется на следующие направления: B2B (бизнес – бизнесу), B2G (бизнес – государству), B2C (бизнес – клиенту), C2C (клиент-клиенту), G2B (государство – бизнесу), B2A (бизнес – администрации), A2B (администрация – бизнесу).

Бизнес для клиентов (B2C) является одним из звеньев в цепочке современных бизнес-процессов, и это звено строит бизнес-отношения следующего плана взаимодействий: «Бизнес-клиент».

B2C – это концепция построения бизнес-процессов предприятия и комплекс Интернет-технологий и инструментов, обеспечивающих повышение прозрачности предприятия и облегчающих его взаимодействие с клиентами. Один из наиболее популярных инструментов B2C – Интернет-магазин (далее ИМ). Именно данный инструмент должен реализовать автор в дипломной работе.

Существуют множество причин, указывающих на необходимость ведения бизнеса в виртуальном мире. В данный момент времени каждая организация, занимающаяся торговлей, пытается привлечь как можно больше клиентов. Рекламирывать и продавать продукцию организации становится все труднее и труднее из-за обилия конкурентов на рынке. Тем не менее, ни одна реклама не может сравниться с рекламой, которую может предоставить ИМ. Один запрос в поисковую систему по соответствующему названию может вывести потенциального клиента на портал организации. Доступность данного метода может привлечь сотни клиентов, тем самым, повысив количество продаж товаров организации.

Другая причина начать электронный бизнес заключается в том, что почти каждый клиент судит о компании по наличию последней собствен-

ного портала в сети Интернет. Если выбранная клиентом компания не имеет сайта, это может насторожить его, и он в 90 процентах случаях уйдет к конкуренту, если тот имеет свой «виртуальный проспект». Таким образом, наличие хотя бы своего собственного портала, не говоря уже об ИМ, говорит о состоятельности той или иной организации.

Последняя, наиболее важная причина для организации начать ведение электронной деятельности в сети Интернет – удобство для пользователя. Пользователь может находиться на достаточном удалении от магазина организации. Если у него нет желания или времени на посещения магазина напрямую, он может дома в спокойной обстановке зайти на портал организации и заказать необходимые товары. Служба доставки в кратчайшие сроки доставит заказ до места, указанным клиентом.

Создание ИМ для организации требует значительных капиталовложений. Купить уже готовые разработки возможно, но возникают дополнительные трудности сопровождения. В связи с этим, автор ставит перед собой задачу создать максимально простой, дешевый, тем не менее, достаточно функциональный продукт, пользоваться которым в ближайшие годы не составит труда для менеджеров организации.

# 1 Обзор вариантов решения задачи

## 1.1 Развернутая постановка задачи

Основные действия, которые необходимо выполнить для решения задачи:

1. Создание информационной системы, реализующей логику ИМ.
2. Создание удобного интерфейса пользователя через браузер и сотовый телефон для управления всей системой в целом.
3. Выбор программных средств реализации поставленной задачи.
4. Программная логика должна быть реализована для четырех пользователей системы: клиента, менеджера, администратора, разработчика системы.
5. Система должна быть многопользовательской.

Доступ через браузер подразумевает использование веб-интерфейса для доступа к системе. Доступ через сотовый телефон подразумевает использование клиента на платформе J2ME для доступа к функциональности ИМ.

У автора на выбор предоставляется много технологий программирования, поэтому второстепенная задача – провести анализ существующих средств и правильно сделать выбор.

## 1.2 Анализ аналогичных разработок

Существуют множество компаний, разрабатывающих аналогичные программные продукты:

1. Компания «ВЕБ-студия ВВ2». Программисты этой фирмы давно себя зарекомендовали с хорошей стороны. Их разработки удовлетворяют современным требованиям построения интерфейса для пользователя. Но есть у них два больших недостатка. Первый недостаток – цена (1000 условных единиц) за каждый заказ (рекомендуется примерно от 50 до 100 условных единиц). Второй недостаток – они используют технологии создания приложений, о которых заказчик может только догадываться. Заказчик должен быть уверен в безопасности, масштабируемости разрабатываемой системы, заранее знать о том, как конфиденциальные данные будут передаваться по сети (в зашифрованном, незашифрованном виде), как будут храниться данные и т.п. Вся эта информация закрыта для заказчика.

2. «1С: Аркадия ИМ». Хотя фирма 1С зарекомендовала себя с лучшей стороны в области бухгалтерского учета и торговли, для крупномасштабных проектов она непригодна. Для выполнения длительных транзакций используется SQL-версия продукта 1С. Это большие накладные расходы на сервер MSSQL. Без него в сети с большим числом пользователей (каким является Интернет) использование данной разработки не имеет смысла. К тому же версия 1С не кроссплатформенная, так как она базируется на операционной системе семейства Windows, как известно – самой небезопасной системы с большим числом ошибок, эксплуатируемых хакерами.

3. «Magento» – профессиональное решение для электронной коммерции с открытым исходным кодом, распространяющееся в соответствии с Open Software License (OSL 3.0). Это программное обеспечение создано с использованием Zend Framework.

4. «osCommerce» – это механизм ИМ. Он может быть установлен на любом сервере с поддержкой PHP и MySQL [1]. Это свободное программное обеспечение, распространяющееся по лицензии GNU. Вокруг osCommerce сформировалось огромное сообщество, благодаря которому существует более 4000 различных модулей для osCommerce, позволяющие изменять и дополнять функции ИМ самым разным образом. По всему миру функционирует десятки тысяч ИМ на базе osCommerce. Среди этого множества встречаются и достаточно успешные и крупные ИМ.

Мы привели недостатки первого и второго вариантов. Теперь рассмотрим оставшиеся два варианта. Данные варианты являются наиболее подходящими вариантами. Абсолютно бесплатные программные продукты для любых организаций. Основные недостатки:

- сложность сопровождения программного продукта (нужен специалист, который мог бы совместить желание заказчика с возможностями данных продуктов);
- открытый исходный код позволяет хакерам быстрее находить уязвимости и эксплуатировать их. Должно пройти достаточно времени, чтобы данные продукты были полностью избавлены от ошибок, которые почти всегда есть в каждом программном продукте;
- весь интерфейс полностью на английском языке. Русификация данных продуктов займет достаточно времени и денег для поиска опытного переводчика, что однозначно не устраивает заказчика;

- невозможность добавления функциональности в виде веб-служб.

Отсюда сделаем единственный вывод – в текущий момент времени на рынке программных продуктов нет тех, которые бы полностью удовлетворяли заказчика. Это еще раз доказывает актуальность разработки.

### 1.3 Выбор технологии программирования

Для программирования для сети Интернет разработано много языков программирования. В числе их наиболее известные С#, PHP, Java [2].

С# – творение Microsoft. Язык, наряду с технологией ASP .Net позволяет создавать код, генерирующий динамические страницы. Microsoft .Net – это платформа, включающая серверы, клиенты и сервисы. Она содержит набор приложений, таких, как Visual Studio .Net, Tablet PC, и .Net My Services. Microsoft .Net была спроектирована таким образом, чтобы удовлетворять текущие и будущие требования заказчиков к созданию, внедрению и управлению приложениями. Фундаментальным принципом Microsoft .Net является изменение процесса разработки и внедрения программного обеспечения – в частности, переход от разработки собственных программ к покупке, настройке и интеграции готового программного обеспечения. Единственный недостаток, позволивший автору отказаться от С# – приложения, написанные на данном языке на платформе .Net до сих пор нельзя переносить между платформами.

Язык PHP – сравнительно молодой, но в то же время удивительно удобный и гибкий язык для программирования приложения для сети Интернет. С помощью него можно написать 99 процентов программ, которые обычно требуются в Интернете. Для оставшегося 1 процента придется использовать С или Perl. Главный недостаток – язык располагает к ошибкам, которые допускают программисты низкого и среднего уровня. Если и разрабатывать приложение на PHP, то необходимо знать в корне все его особенности, иначе приложение просто опасно использовать. Не счесть случаев использования глубоких знаний этого языка в корыстных целях, которые преследуют хакеры. Существуют много реальных примеров взлома ИМ, построенных на технологии PHP. Автор не рискнул его использовать в качестве технологии для своего проекта.

Язык Java – творение Sun Microsystems. Отличительная особенность Java – средства безопасности. Как язык, так и сама платформа с самого на-

чала разрабатывались с учетом безопасности. Java-платформа позволяет пользователям загружать по сети непроверенный код и запускать его в безопасной среде, в которой он не сможет нанести какой-либо ущерб. Одни эти возможности делают Java уникальной платформой.

Любой Java-код – апплет, сервлет или консольное приложение – можно запускать с ограниченными правами доступа, что предотвратит возможность нанесения какого-либо ущерба системе.

Вопросы безопасности, связанные с языком и платформой Java, тщательно рассматривались специалистами во всем мире. Платформа на практике показала себя достаточно надежной в повседневном использовании. Безусловно, она лучше всех других альтернатив.

Java приложения могут выполняться на любых операционных системах: системах уровня предприятия, домашних операционных системах, а также операционных системах для мобильных устройств.

Можно также отметить, что технология Java является открытой и построена на внутриотраслевых стандартах для программного обеспечения. Любой желающий может загрузить и изучать код Ява платформы.

Java платформа выпуска для предприятий версии два – набор спецификаций и соответствующей документации для языка Java описывающей архитектуру серверной платформы для задач средних и крупных предприятий.

Спецификации детализированы настолько, чтобы обеспечить переносимость программ с одной реализации платформы на другую. Основная цель спецификаций – обеспечить масштабируемость приложений и целостность данных во время работы системы. Данная платформа во многом ориентирована на использование в Интернете, так и в локальных сетях.

Технологию J2EE [3] рационально использовать для реализации крупных проектов, например для организации сложных веб-порталов и предоставления услуг в режиме реального времени, особенно если необходимо обеспечить бесперебойную работу многих тысяч пользователей.

Java платформа микро выпуска второй версии – подмножество платформы Java для устройств, ограниченных в ресурсах.

Данная платформа разработана под руководством все той же корпорации Sun Microsystems.



Альтернативой данной платформе для программирования сотовых телефонов в данный момент автор не увидел, поэтому и сделал окончательный вывод об использовании платформы Java для реализации проекта.

Итак, основываясь на вышеизложенном высказывании, можно сказать, что разработка проекта именно на Java имеет много перспектив в будущем.

#### 1.4 Независимость от СУБД

Одно из достоинств разрабатываемой системы – независимость от системы управления базой данных [4] на основе объектно-реляционного отображения данных (ORM), реализованного в библиотеке Hibernate [5].

Hibernate – решение для языка программирования Java. Является свободным программным обеспечением с открытым исходным кодом, которое распространяется по условиям GNU. Данное решение предоставляет легкий в использовании каркас для отображения объектно-ориентированной модели данных в традиционную реляционную базу данных.

Целью Hibernate является освобождение разработчика от значительного объема общих задач программирования по обеспечению сохранности данных после прекращения работы программы. Разработчик может начать использовать Hibernate в процессе разработки, как с нуля, так и для уже существующей базы данных.

Hibernate не только заботится о связи Java классов с таблицами базы данных, но также предоставляет средства для автоматического построения запросов и извлечения данных и может значительно уменьшить время разработки, которое обычно тратится на ручное написание SQL и JDBC кода.

Hibernate позволяет нам безболезненно хранить данные в постоянном хранилище, при этом выбор типа хранилища, установка и конфигурирование не составляют больших трудностей. Hibernate позволяет хранить объекты любого вида, поэтому приложению не обязательно знать, что его данные будут сохраняться с использованием Hibernate. Понятно, что с помощью Hibernate мы можем не только сохранять данные в хранилище, но также обновлять и удалять данные.

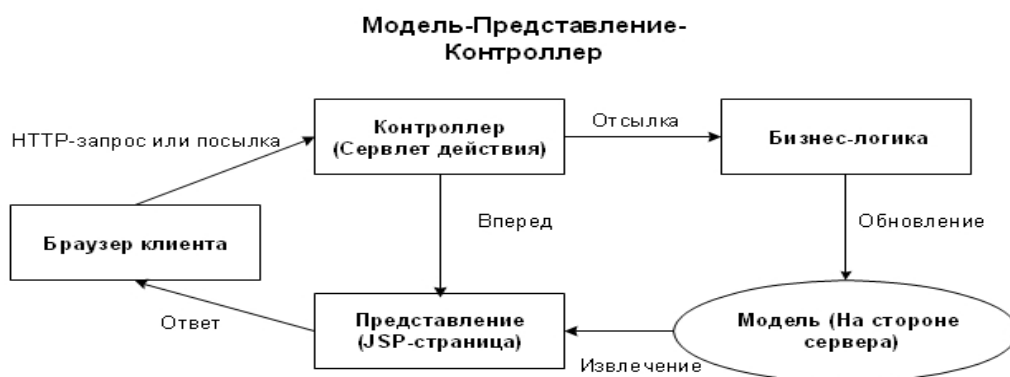
## 1.5 Шаблон проектирования MVC

Общепринятый метод проектирования приложений сводится к организации последних вокруг пользовательского интерфейса, управляемого событиями. В спроектированном шаблоне разработчик создает интерфейс, а после этого пишет код, который должен производить необходимые изменения в соответствии с действиями пользователя. Такой подход может быть успешно применен для небольших однопользовательских систем, которые не требуют значительных изменений функциональных возможностей с течением времени. Однако для больших распределенных проектов такая схема неприемлема.

Результатом применения шаблона MVC является отделение данных приложения от способов доступа к данным и их просмотра, а также от соответствия между системными событиями (такими как события пользовательского интерфейса) и поведением приложения.

Шаблон MVC состоит из трех составляющих его типов. Модель представляет данные приложения наряду с методами, которые оперируют этими данными. Компонент Представление отображает данные пользователю. Контроллер транслирует действия пользователя, такие как движение мыши или ввод с клавиатуры, и посылает их модели для дальнейшей обработки. В результате этого модель модифицирует Представление для отображения изменения данных. Далее иллюстрируются функции каждого из этих составляющих типов и отношения между ними (смотри рисунок 1.1).

Когда такая парадигма используется должным образом, Модель не принимает участия в трансляции формата входных данных. Эта трансляция должна выполняться Контроллером. Кроме того, Модель не должна нести ответственности за определение того, каким образом отображаются результаты.

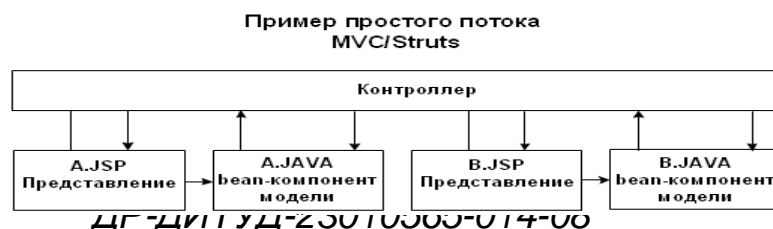


## Рисунок 1.1 – Шаблон проектирования MVC

MVC обеспечивает следующие преимущества:

1. Позволяет проектировать приложение, отделяя моделирование данных от отображения информации и взаимодействия между пользователем и системой.
2. Обеспечивает просмотр одних и тех же данных несколькими способами.
3. Позволяет нескольким пользователям одновременно просматривать одни и те же данные.
4. Упрощает анализ последствий, благодаря чему улучшается расширяемость.
5. Облегчает эксплуатацию, формируя прикладные функции за доверенными API.
6. Расширяет возможность многократного использования за счет отделения функциональных прикладных возможностей от представления.
7. Облегчает распределение приложения, поскольку границы MVC являются естественными областями распределения интерфейса.
8. Может использоваться для разделения доставки, а также для того, чтобы обеспечить возможность инкрементных изменений.
9. Позволяет усилить четкое разделение обязанностей и функционального состояния. Таким образом, облегчается контроль.

Приложения, использующие библиотеку Struts [6], соответствуют модели MVC. Три основных компонента этой модели – сервлет (Контроллер), JSP-страницы [7] (Представление) и бизнес-логика приложения (Модель) (смотри рисунок 1.2).



## Рисунок 1.2 – Реализация шаблона проектирования MVC

Сначала запрос пользователя идет на Контроллер, который инициализирует контекст/сеанс для полной транзакции и передает Представление модели страницы А. После этого Контроллер передает выполнение Представлению страницы А. Страница А отправляет данные назад на Контроллер, который вызывает модель, чтобы проверить корректность входных данных формы. Если ввод недопустим, Модель возвращает результат, который Контроллер использует, чтобы отправить или повторно отобразить страницу А.

Контроллер определяет следующего в цепочке путем использования большого числа параметров.

Контроллер представляет собой нечто вроде централизованного регулятора на перекрестке, который знает, когда требуемая последовательность шагов в транзакции закончена. Модель сохраняет «состояние» (отосланные поля формы) и логику проверки правильности хранения данных. Если действие пользователя недопустимо, Контроллеру посылается предупреждение для восстановления того же самого Представления.

Контроллер связывает вместе HTTP-запросы и направляет их к другим объектам структуры, включая JSP-страницы. После этого контроллер анализирует конфигурационный файл ресурсов, который определяет отображения действий приложения. Контроллер, используя эти отображения, преобразует HTTP-запросы в соответствующие действия приложения.

Отображение должно как минимум определить путь запроса и то, какой тип объекта должен действовать в ответ на запрос. Объект действия либо обрабатывает запрос и отвечает клиенту, либо указывает, что управление должно быть перенаправлено другому действию.

Так как объекты действия связаны с Контроллером приложения, они имеют доступ к методам сервлета. Когда объект передает управление, он может косвенно передавать один или несколько совместно используемых объектов, таких как бобы Java, путем помещения их в стандартный набор совместно используемых Java-сервлетами.

### 1.6 Выбор сервера приложений

Так как условиями задачи predetermined использование Java-технологии для решения задачи, нам необходимо использовать тот сервер, который поддерживал бы платформу J2EE. На выбор нам предоставляется множество различных серверов. Для нашей задачи подошли бы два из всего множества. Это сервер Blazix и Jakarta Tomcat Container [8]. Первый представляет собой быстродействующий сервер приложений Java. Предельно простой в настройке и администрировании. Распространяется бесплатно для некоммерческого использования. Второй имеет смысл использовать в том случае, когда будет необходимость использовать наш проект как коммерческий. Итак, выберем Tomcat версии 5.0.

Tomcat – программа-контейнер сервлетов, написанная на языке Java и реализующая спецификацию сервлетов и спецификацию серверных страниц Java (JSP), которые являются стандартами для разработки приложений уровня веб.

Tomcat разрабатывается фондом Apache Software Foundation под лицензией Apache License и является программой с открытым исходным кодом.

### 1.7 Выбор инструментов для разработки

Для разработки программного комплекса будут применяться следующие системы программирования: Htmllpad Fisherman 1.9 и NetBeans IDE 5.0.

NetBeans IDE 5.0 – свободная интегрированная среда разработки приложений на языке программирования Java. Среда разработки по умолчанию поддерживает разработку для платформ J2SE и J2EE. Для разработки программ в среде NetBeans и для успешной инсталляции и работы самой среды NetBeans должен быть предварительно установлен Sun JDK или J2EE SDK подходящей версии. Для поддержки разработки в среде NetBeans для мобильных платформ (J2ME) необходимо установить отдельно распространяемый пакет (и также бесплатный) NetBeans Mobility Pack. Благодаря этому инструменту автор разработает уровень Модели и Контролера шаблона MVC.

Htmllpad Fisherman 1.9. Многофункциональный HTML редактор с тонкой настройкой. Многооконный русскоязычный интерфейс. Подцветка синтаксиса для 14 языков. Работа с различными кодировками. Панель те-

гов, готовые скрипты и шаблоны, необъятное дерево HTML цветов, тегов HTML, свойств CSS. С помощью данного редактора будет создаваться уровень Представления шаблона MVC.

### 1.8 Веб-службы

В данной работе предполагается написание Веб-службы [9]. Рассмотрим, что они из себя представляют.

Веб-служба – программная система, идентифицируемая строкой URI, чьи общедоступные интерфейсы определены на языке XML. Описание этой программной системы может быть найдено другими программными системами, которые могут взаимодействовать с ней согласно этому описанию посредством сообщений, основанных на XML, и передаваемых с помощью протоколов сети Интернет. Важнейшие протоколы прикладного уровня, необходимые для работы служб: XML-RPC и SOAP.

Достоинства веб-служб:

- веб-службы обеспечивают взаимодействие программных систем независимо от платформы;
- веб-службы основаны на базе открытых стандартов и протоколов. Благодаря использованию XML достигается простота разработки и отладки веб-служб;
- использование протокола HTTP обеспечивает взаимодействие программных систем через межсетевой экран.

Веб сервисы развёртываются на серверах приложений. В нашем случае мы будем использовать связку Axis и Tomcat.

### 1.9 Использование апплетов в работе системы

Все апплеты являются подклассами Applet. Так как все апплеты выполняются в окне, необходимо включить поддержку для этого окна. Апплеты не исполняются интерпретатором Java времени выполнения, работающим в консольном режиме. Они выполняются браузером.

Таким образом, автор должен реализовать апплет для работы на стороне клиента. Клиент в свою очередь должен установить на локальную машину среду выполнения Java приложений (JRE).

В ряде случаев каждый апплет должен быть подписан цифровой подписью. В нашем случае апплет не будет подписан, так как автор не имеет соответствующего сертификата.

Как только апплет откомпилирован, он включается в HTML-файл. Апплет будет выполняться Java-совместимым браузером, когда тот встретит в HTML-файле тег <applet>.

Выводы к разделу 1

В данном разделе автор рассмотрел различные варианты решения поставленной задачи; определился с набором технологий, которые необходимо применять при реализации проекта.

Были определены основные моменты реализации на стороне клиента и на стороне сервера.

Автору осталось спроектировать и закодировать модули, призванные решить поставленную задачу.

Confidential

## 2 Разработка структуры программного комплекса

### 2.1 Механизм объектно-реляционного отображения (ORM)

Основой любой базы данных являются таблицы. Таблица состоит из строк и столбцов и имеет уникальное имя в базе данных. База данных содержит множество таблиц, связь между которыми устанавливается с помощью совпадающих полей. В каждой из таблиц содержится информация о каких-либо объектах одного типа.

Между таблицами в базе данных устанавливаются отношения. Имеются четыре типа отношений между таблицами: один-к-одному, один-ко-многим, много-к-одному, много-ко-многим.

Отношение один-к-одному означает, что каждая запись одной таблицы соответствует только одной записи в другой таблице. В разрабатываемом приложении нет такого вида отношений.

Наиболее часто встречающимся типом отношений в базе данных является отношение один-ко-многим. Например, пользователь может иметь несколько адресов доставок заказов.

Отношение много-к-одному аналогично рассмотренному ранее типу один-ко-многим. Тип отношения между объектами зависит от вашей точки зрения. Например, если вы будете рассматривать отношение между сделанными заказами и клиентами, то получите отношение много-к-одному.

Отношение много-ко-многим возникает между двумя таблицами в тех случаях, когда: одна запись из первой таблицы может быть связана более чем с одной записью из второй таблицы; одна запись из второй таблицы может быть связана более чем с одной записью из первой таблицы. В нашем приложении существует одна связь такого типа: в одном заказе могут находиться несколько наименований товаров, и одно наименование товара может быть в нескольких заказах.

Hibernate – библиотека, манипулирующая Java объектами. Именно с помощью нее можно преобразовать отношения в таблицах в отношения между классами.

Отметим важный момент: Hibernate должен работать с СУБД, поддерживающей реляционную модель.



Ниже представлена схема, показывающая, как происходит преобразование (смотри рисунок 2.1).

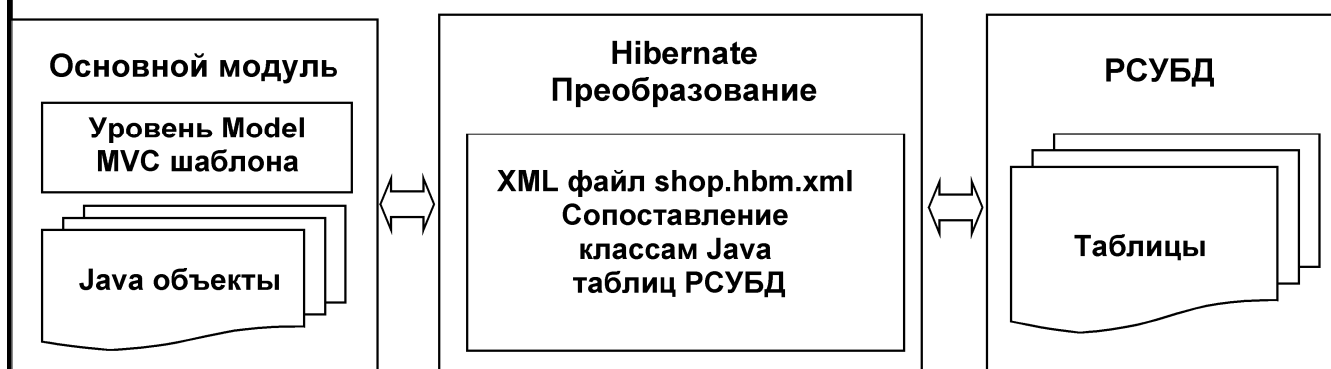


Рисунок 2.1 – Механизм ORM, используемый в приложении

В этой схеме важную роль играет файл shop.hbm.xml. Именно в нем указывается, как объекты Java преобразуются в таблицы.

## 2.2 Структура веб-службы

Веб-служба, реализованная в данном проекте позволяет администратору просматривать некоторую статистику по работе системы. Хотя доступ к ней будут иметь многие, но данные, которые можно получить с помощью нее не являются важными в плане безопасности. Итак,

- веб-служба предоставляет возможность программе-клиенту запрашивать у нее информацию по протоколу SOAP;
- SOAP – простой протокол доступа к объектам, работающий поверх HTTP протокола (FTP, SMTP, JMS);
- SOAP – протокол основан на передаче XML-сообщений;
- клиентом может быть программа, написанная на любом языке программирования.

Протокол SOAP не различает вызов процедуры и ответ на него, а просто определяет формат послания в виде документа XML. Послание может содержать вызов процедуры, ответ на него, запрос на выполнение каких-то других действий или просто текст. Спецификацию SOAP не интересует содержимое послания, она задает только его оформление. Корневой элемент посылаемого документа XML <Envelope> содержит необязательный заголовок <Header> и обязательное тело <Body>. В заголовке содержится один или несколько блоков, оформление и содержание которых никак не регламентируются. Точно так же ничего не говорится о содержании тела послания. Тем не менее, различают процедурный стиль послания

SOAP, предназначенный для вызова удаленных процедур, и документный стиль, предназначенный для обмена документами XML. Процедурный стиль часто называют RPC-стилем, а документный стиль – XML-стилем.

Пакет SAAJ обеспечивает создание SOAP-посланий и обмен ими в синхронном режиме «точка к точке» (P2P). Часто такой способ обмена сообщениями оказывается невозможным из-за того, что участники обмена не всегда одновременно находятся на связи. В таких случаях приходится обмениваться сообщениями асинхронно, не дожидаясь ответа на посланное сообщение. Так работает электронная почта. В технологии Java асинхронный обмен SOAP-посланиями обеспечивается интерфейсами и классами пакета JAXM.

Итак, разрабатываемая веб-служба будет работать в синхронном режиме, и будет реализована с помощью пакета SAAJ. Структура веб-службы представлена ниже (смотри рисунок 2.2).

Информация, которую можно получить с помощью веб-службы:

- количество завершенных заказов;
- общая сумма на счете;
- выручка с продаж.

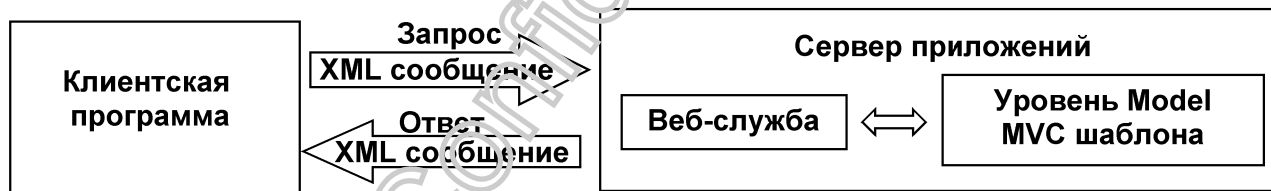


Рисунок 2.2 – Структура веб-службы

### 2.3 Обеспечение независимости от СУБД

Как было отмечено раньше, Hibernate позволяет абстрагироваться приложению от СУБД. Можно отметить следующие моменты:

- Hibernate – не СУБД, лишь библиотека, использующая основные возможности СУБД;
- требование, предъявляемое приложением к СУБД – поддержка транзакций.

Далее представлена схема, показывающая, каким образом программный компонент может работать с различными СУБД (смотри рисунок 2.3).



Рисунок 2.3 – Независимость от СУБД

При переносе приложения между СУБД важную роль играет файл hibernate.cfg.xml. В нем фактически определены все параметры для работы с текущей СУБД, такие как уровень изоляции, строка соединения и т.п.

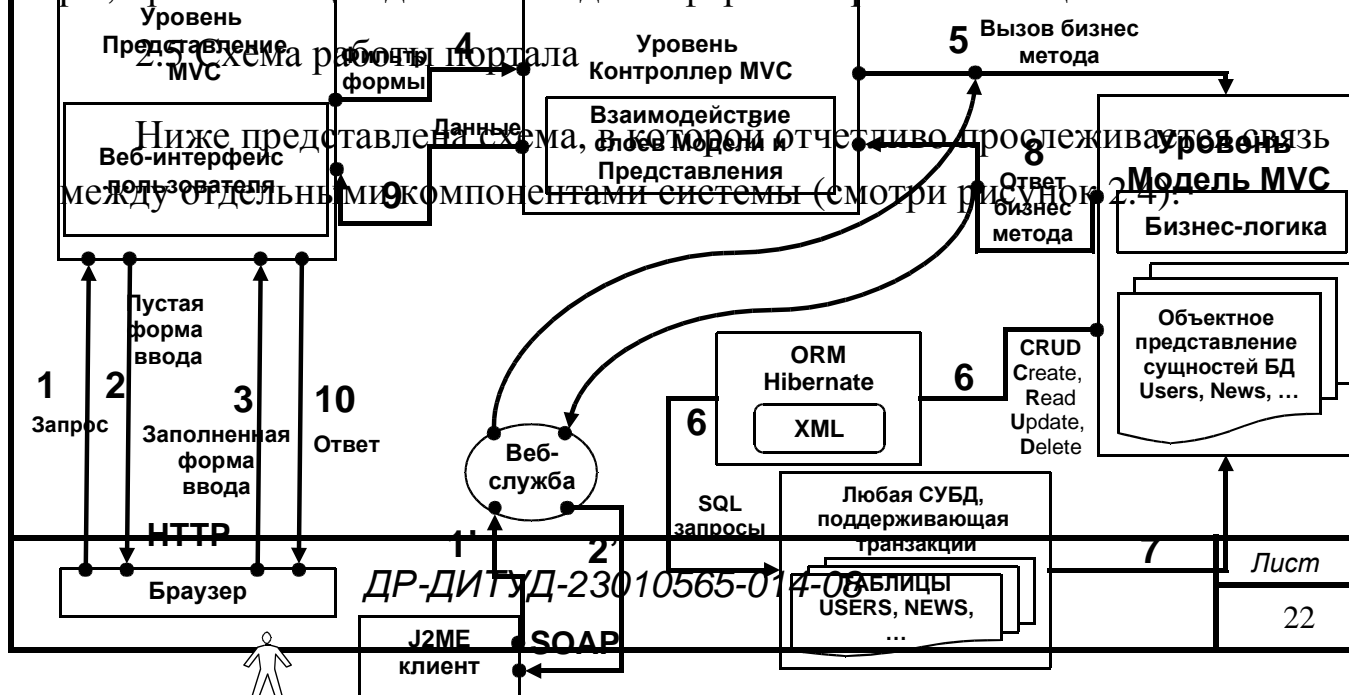
## 2.4 Проектирование с помощью шаблона MVC

Библиотека Struts, реализующая данный шаблон позволяет разделить монолитное приложение на три части: Представление, Модель и «мост» между ними – Контроллер. Предварительно нужно определить файл конфигурации struts-config.xml.

В данном файле приведены разделы action-mappings, beans, message-resources, global-forwards.

Раздел beans определяет формы, в которых определены поля ввода для пользователя. В нашем случае определены четыре формы ввода: форма для клиента, форма для менеджера и администратора, форма авторизации и форма отчетов.

Раздел action-mappings указывает, как форма ввода связывается с контроллером. Именно в данном разделе указываются классы контроллеров, принимающие данные входных форм и обрабатывающие их.



### Рисунок 2.4 – Схема работы портала

Как видно из рисунка 2.4, десять этапов определяют работу всей системы в целом.

#### Выводы к разделу 2

В данном разделе автор спроектировал структуру программного комплекса, используя выбранные технологии. Процесс программирования модулей удалось разбить на несколько этапов, что существенно облегчит поставленную задачу.

Таким образом, структура комплекса разработана полностью.

## 3 Реализация программного комплекса

### 3.1 Реализация безопасности

Каждая система электронной коммерции обязана иметь модули защиты данных пользователей. В данном разделе рассмотрим основные способы защиты, реализованные в программном комплексе.

#### 3.1.1 Шифрование данных

Никто не сомневается, что хранить номера кредитных карт и пароли клиентов в базе данных в открытом виде нельзя. Доступ к базе данных могут иметь многие пользователи, в частности персонал ИМ, работающий на предприятии. Поэтому эта мера является шагом по обеспечению конфиденциальности данных пользователей.

Чтобы добиться этого программный комплекс должен иметь модуль для шифрования данных. Рассмотрим более детально данную проблему.

Шифром принято называть обратимый способ преобразования информации с целью защиты ее от просмотра, в котором используется некий секретный элемент. Исходная информация в этом случае будет называться открытым текстом, а результат применения к ней шифра – закрытым текстом.

Шифры бывают блочными и потоковыми. Блочный шифр работает с сообщениями фиксированного размера (например, 64 бита), а поточный – шифрует весь поток данных посимвольно (например, побайтно). Известные блочные шифры – DES, IDEA, Blowfish, потоковые – RC4. Блочность шифра не означает невозможность шифрования им сообщений, превышающих по длине размер блока.

Рассмотрим, как платформа Java предоставляет возможность шифрования данных.

В JDK используется SunJCE провайдер, предоставляющий следующие возможности, необходимые в нашем проекте:

- реализация DES (FIPS PUB 46-1), тройного DES (DES-EDE), и Blowfish алгоритмов шифрования;
- ключевые генераторы для создания ключей, подходящих для DES, Triple DES, Blowfish, HMAC-MD5, и HMAC-SHA1 алгоритмов;
- «фабрики» по производству секретных ключей для обеспечения двупольного преобразования между непрозрачным представлением

ключа DES, DES-EDE и прозрачным представлением его в виде отдельного файла.

JCE входящая в JDK, включает два компонента программного обеспечения, которые мы будем использовать:

- библиотека, определяющая и поддерживающая криптографические сервисы. API находится в пакете `javax.crypto`;
- провайдер по имени «SunJCE».

Автор остановил свой выбор на блочно-симметричном способе шифрования информации под названием «DES-EDE». DES – 64-битный алгоритм с эффективной длиной ключа в 56 бит (хотя часто говорят о 8 байтах, но старший бит в байте не используется). DES-EDE – тройной DES. Алгоритм шифрования состоит в следующем: исходный текст зашифровывается DES с ключом K1, результат расшифровывается DES с ключом K2, а этот результат опять зашифровывается DES с ключом K3. Итого, эффективная длина ключа получается 168 бит. Действительная длина ключа все равно будет 192 бита или 24 байта.

Администратор должен создать один секретный ключ. Ключ будет шифровать пароли пользователей системы. Ключ должен храниться на сервере для проведения задач авторизации, как персонала, так и клиентов ИМ. Ключ должен быть скрыт от лишних глаз и очень хорошо защищен.

Перейдем непосредственно к модулю шифрования.

Модуль для шифрования представляет собой класс, находящийся в пакете `com.omendbашop`. Его название – `ShopCipher`. Класс имеет методы:

- `createKey` – создание симметричного ключа и запись его в файл;
- `readKey` – чтение из файла симметричного ключа;
- `shopDeCrypt` – дешифрация входного массива байт в строку;
- `shopEnCrypt` – шифрация входной строки в массив байт.

### 3.1.2 Использование SSL

Теперь рассмотрим, как можно безопасно передавать симметричный ключ по сети на сервер.

Клиенты, передавая свои данные через Интернет, должны иметь уверенность в конфиденциальности передаваемых данных. Сейчас стандартом для прозрачного шифрования данных является протокол SSL. Так

как веб-интерфейс является нашей главной задачей, мы должны предоставить пользователю возможность передавать свои данные через протокол HTTPS (протокол HTTP, использующий SSL туннель).

Так как архитектура разрабатываемой системы является трехуровневой, значит, информация от пользователя может пройти как минимум два важных участка на пути к базе данных.

Данные шифруются и посылаются на обработку веб-серверу с использованием шифрующего канала SSL. Участок от веб-сервера до СУБД является внутренней сетью предприятия. На этом участке пароль уже зашифрован с помощью симметричного ключа.

### 3.1.3 Сертификат для веб-сервера

С первых дней электронной коммерции остро стоял вопрос, как осуществить привязку сервера, сайта, домена к определенному юридическому или физическому лицу? Интернет – настолько свободная информационная среда, что любой может выдать себя за кого угодно.

Для решения этой проблемы были созданы сертификационные центры. Самые известные из них – Thawte и VeriSign. Эти организации занимаются выдачей электронных сертификатов юридическим и физическим лицам. Сертификат привязан к определенному домену. Использование сертификата на сайте с другим доменом невозможно – клиенту будет выдаваться предупреждение, сообщающее о несоответствии домена, указанного в сертификате реальному. Центры ведут реестр лиц, которым выданы сертификаты, содержащие паспортные данные, ИНН и другие сведения, достаточных, например, для предъявления исков конкретному лицу.

В нашем случае будет достаточно самоподписанного сертификата. Если пользователь не хочет оставлять о себе сведения на сайте, имея перед собой только лишь самоподписанный сертификат, мы можем приобрести сертификат в компании VeriSign. Это фирма является надежным посредником, которому доверяет и руководство ИМ и сам клиент. Поэтому сертификат должен быть получен именно из этой организации.

Для создания и управления сертификатами в пакете JDK предусмотрена специальная программа keytool. Эта программа управляет хранилищами ключей, базами данных сертификатов и закрытых ключей. Каждый элемент хранилища ключей имеет псевдоним.

Теперь преступим к созданию хранилища ключей (хотя введенные данные будут отображены в сертификате). Необходимо выполнить следующий сценарий:

```
keytool -genkey -keyalg rsa -alias Shopalias -keystore Shopkeystore
```

Утилита попросит ввести пароль для доступа к сертификату, затем данные о хосте, названии организации, компании, городе, области и стране. Затем необходимо подтвердить операцию, выбрав yes. После необходимо подтвердить пароль.

Если необходимо преобразовать полученный файл к виду, пригодному для обычного просмотра (сертификат, который можно просмотреть в ОС Windows XP), то необходимо выполнить следующую команду:

```
keytool -export -alias Shopalias  
-keystore Shopkeystore -file Shopalias.crt
```

Данную команду должен выполнить администратор сервера, так как необходим пароль к хранилищу ключей.

Если в настройках веб-сервера все будет прописано правильно, при заходе клиента на сайт через HTTPS, ему будет выведено окно, в котором он сможет просмотреть данный сертификат и импортировать его в хранилище сертификатов в случае полного доверия сайту.

### 3.2 Реализация веб-службы

Веб-служба с технической стороны представляет собой файл с расширением jws – в нашем случае это файл ForMidlet.jws. Данный файл помещается в веб-приложение axis и внедряется в сервер приложений вместе с ним.

Клиентом данной службы может быть любая программа, которая может работать по протоколу SOAP.

Автор реализовал клиент на платформе J2ME для веб-службы. Клиент работал с библиотекой kSoap.

### 3.3 Работа с почтой

Библиотека JavaMail представляет собой набор абстрактных API, являющихся надстройкой над системой поставщика услуг электронной почты. API JavaMail создает для клиентских приложений Java стандартную и



независимую среду, в которой они могут свободно пользоваться почтой, что позволяет разработчикам создавать программы, обеспечивающие следующие возможности:

- создание сообщений, в том числе и многокомпонентных, содержащих присоединенные файлы;
- отправка сообщений на определенные серверы;
- получение сообщений и сохранение их в папках.

Разработчику нет необходимости разбираться в соответствующих протоколах (смотри рисунок 3.1). Основную работу с почтой выполняют провайдеры услуг, так как содержат обработчики почтовых протоколов (например, POP3, IMAP4, SMTP).



Рисунок 3.1 – Архитектура JavaMail

В пакете com.omendba.shop присутствует разработанный класс для работы с почтой, использующий API, описанный выше – SendMail, расширяющий класс Thread. Поэтому данный класс определяет метод потока run, совершающий основную работу по отправлению письма.

Класс предоставляет методы для добавления в тело письма сообщений, определяемых логикой приложения:

- setContentBodyActive – добавить в тело сообщение об активации/деактивации. Менеджер может управлять активацией/деактивацией клиента в случае необходимости. При каждом изменении статуса пользователя ему на почту отправляется письмо с уведомлением о новом статусе;

- `setContentBodyFromManager` – добавить в тело письма сообщение от менеджера. Менеджер при желании может отправить письмо клиенту. Этот метод и добавляет текст менеджера в тело письма;

- `setContentBodyNews` – добавить в тело письма новость. При добавлении новости менеджером происходит отсылка новости всем подписавшимся на рассылку клиентам;

- `setContentBodyRegistration` – добавить в тело письма сообщение об активации учетной записи. При регистрации клиента после успешного завершения процедуры на почту отправляется письмо со ссылкой активации. Зайдя по ссылке, клиент в браузере увидит сообщение, о том, что он успешно активировал свой профиль;

- `setContentBodyRemind` – добавить в тело письма напоминание для клиента: логин и пароль. Если клиент забыл пароль, то он может воспользоваться услугой напоминания пароля. При этом ему на почту высылается письмо, содержащее логин и пароль.

### 3.4 Работа с журналом

Мониторинг работы приложения очень важен. Поэтому был создан модуль, записывающий все исключительные ситуации в журнал. Журнал хранится в домашнем каталоге пользователя на сервере. Домашний каталог пользователя представляет собой каталог, который имеет в своем распоряжении обладатель учетной записи хостинга.

Модуль для работы с журналом находится в пакете `com.omendba.shop` и имеет название `LogWrite`.

В журнале указывается дата, время, процедура и входные параметры процедуры, в которой произошло исключение.

### 3.5 Структура таблиц базы данных

Для хранения информации используется любая СУБД с поддержкой транзакций. База данных имеет структуру, представленную ниже (смотри рисунок 3.2).

Таблица MESSAGEUSER содержит данные о сообщениях пользователя. Поле STATUS указывает на то, обработано сообщение или нет. Если поле принимает значение равное единице, значит, менеджер обработал данное сообщение. Поле TYPE\_MES отвечает за тип сообщения. Сообщения для данной системы может принимать три значения: жалоба, примечание к сообщению, заявка на ремонт автомобиля. Поле BODY содержит тело сообщения. Поле USERS\_ID содержит идентификатор пользователя, который оставил данное сообщение. В течение некоторого времени менеджер должен по почте отослать письмо с ответом, используя веб-интерфейс, позволяющий отсылать письма.

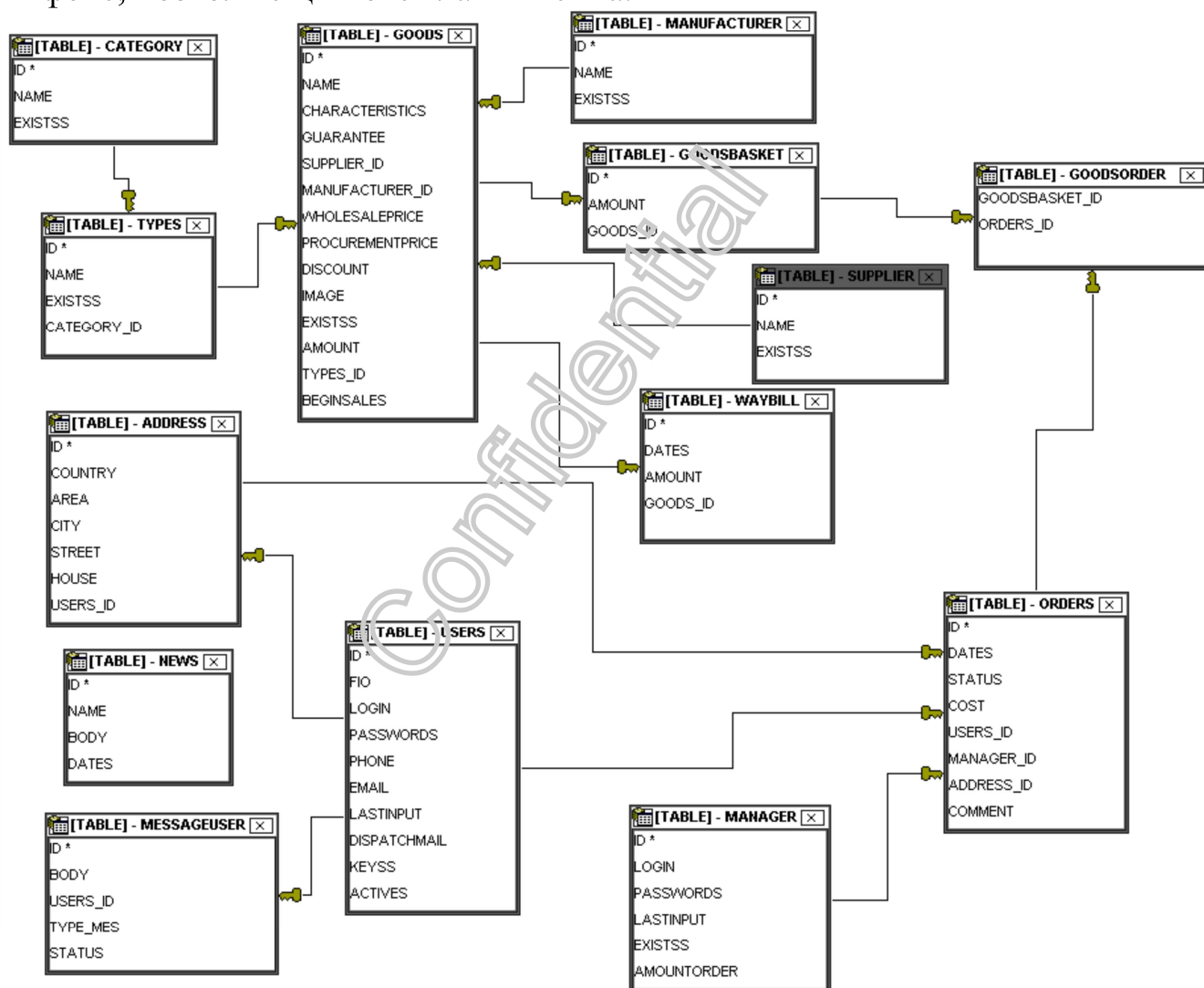


Рисунок 3.2 – Структура таблиц базы данных

Таблица NEWS содержит данные, позволяющие хранить новости. Поле NAME отвечает за заголовок новости. Поле BODY есть содержимое новости. Поле DATES есть дата новости.

Таблица ADDRESS содержит данные об адресах доставки для пользователя. Каждый пользователь вправе иметь несколько адресов доставки.

Поле COUNTRY содержит данные о стране. Поле AREA есть область доставки. Поле CITY есть город доставки. Поле STREET есть улица доставки. Дом HOUSE есть дом доставки. Поле USERS\_ID есть идентификатор пользователя.

Таблица USERS содержит сведения о пользователях-клиентах ИМ. Поле FIO есть фамилия, имя и отчество клиента. Поле LOGIN есть логин клиента, с помощью которого он авторизуется в системе. Поле PASSWORDS есть пароль в шифрованной форме. Пароль имеет тип BLOB (большие бинарные объекты). Поле PHONE представляет собой телефон клиента. Поле EMAIL есть электронный почтовый ящик клиента. Поле LASTINPUT есть дата последнего входа в ИМ. Это поле позволяет видеть дату последнего входа в ИМ. Поле DISPATCHMAIL позволяет сохранить сведения о том, подписался ли клиент на почтовую рассылку. Единица говорит о том, что клиент подписался на почтовую рассылку. Поле KEYSS есть ключ активации учетной записи. Пользователь, зарегистрировавшись в ИМ должен с помощью специальной ссылки вызвать процедуру, которая по этому полю определит, верен ли ключ активации. В случае положительного результата поле ACTIVES принимает значение, равное единице.

Таблица MANAGER содержит данные о менеджерах ИМ. Поле LOGIN содержит логин для авторизации в системе. Поле PASSWORDS есть закодированный пароль. Поле LASTINPUT содержит дату последнего входа в ИМ. Поле EXISTSS содержит целое значение равное единице, если администратор дал право менеджеру посещать ИМ. Поле AMOUNTORDER содержит количество заказов, которое обработал менеджер.

Таблица ORDERS содержит сведения о заказах клиентов. Поле DATES есть дата заказа. Поле STATUS есть статус заказа. Всего по логике приложения существуют ровно пять статусов заказов: 0 – заказ не рассматривался менеджером, 1 – заказ был просмотрен менеджером, 2 – заказ укомплектован, 3 – заказ отправлен клиенту, 4 – заказ завершен, 5 – заказ отменен. Поле COST есть цена заказа с учетом скидок. Поле USERS\_ID содержит идентификатор клиента. Поле MANAGER\_ID содержит идентификатор менеджера, который должен будет обработать заказ. Поле ADDRESS\_ID есть адрес доставки клиента. Если клиент оставит

комментарий по поводу заказа, то комментарий сохранится в поле COMMENT.

Таблица GOODS содержит сведения о товаре. Поле NAME есть название товара. Поле CHARACTERISTICS содержит характеристику товара. Поле GUARANTEE содержит сведения о гарантии товара. Поле SUPPLIER\_ID есть идентификатор поставщика. Поле MANUFACTURER\_ID есть идентификатор производителя. Поле WHOLESALEPRICE содержит сведения об оптовой цене товара. Поле PROCUREMENTPRICE есть розничная цена товара. Также товар может иметь скидку. По умолчанию поле скидки DISCOUNT равно нулю. Товар должен иметь изображение. Если такое имеется, то оно хранится на файл-сервере, то есть в домашнем каталоге пользователя. Ссылка на изображение хранится в поле IMAGE – относительный путь к файлу изображения вместе с названием изображения. Поле EXISTSS равно единице, если товар доступен, иначе поле содержит значение ноль. Поле AMOUNT содержит значение о количестве товаров. Поле TYPES\_ID есть тип, к которому принадлежит товар. Начало продаж товара содержится в поле BEGINSALES. По значению данного поля можно судить о том, новый товар или нет.

Остальные таблицы фактически представляют собой лишь имена категорий, типов, поставщиков, производителей.

Также можно отметить таблицу с накладными WAYBILL. Все данные по приходу товаров сохраняются в данной таблице.

### 3.6 Реализация уровня модели шаблона MVC

Приступим к рассмотрению реализации модели шаблона MVC. Основная логика приложения хранится в двух пакетах com.omendba.shop.manager и com.omendba.shop.client. Первый нужен для работы менеджера. Второй для работы клиента. Рассмотрим подробнее пакеты, принадлежащие данным классам.

Пакет com.omendba.shop.manager.authorization.

Пакет содержит логику контроллера и модель для авторизации менеджера и администратора. Основные методы представлены ниже:

- add – добавить менеджера;

- addRoot – добавить root. Пользователь root представляет собой администратора ИМ. При первом обращении к системе администратор инициализирует систему, задав логин root и пароль;

- auth – авторизация менеджера или администратора. Процедура преобразует пароль к потоку шифрованных байт, запрашивает поток шифрованных байт из базы данных и сравнивает их. Если удачное сопоставление, то авторизация успешна;

- exists – управление доступностью/недоступностью. Администратор имеет право заблокировать менеджера. Заказы все равно будут поступать на имя менеджера;

- update – процедура обновляет пароль для менеджера или администратора. Достаточно ввести новый логин и пароль и вызвать данную процедуру.

Пакет com.omendba.shop.manager.category.

Пакет содержит логику контроллера и модель для управления категориями товаров. Основные методы представлены ниже:

- addCategory – добавить категорию;

- exists – управление доступностью/недоступностью. Менеджер вправе заблокировать категорию. При этом блокируются все зависящие от данной категории элементы – типы и товары. Надо быть внимательным при вызове данной процедуры. Можно также отметить, что процедура займет некоторое время на выполнение логики.

Пакет com.omendba.shop.manager.type.

Пакет содержит логику контроллера и модель для управления типами товаров. Основные методы представлены ниже:

- addType – добавить тип товара;

- exists – управление доступностью/недоступностью. Менеджер вправе заблокировать тип товара. При этом блокируются все зависящие от данного типа элементы – товары. Надо быть внимательным при вызове данной процедуры. Можно также отметить, что процедура займет некоторое время на выполнение логики.

Пакет com.omendba.shop.manager.supplier.

Пакет содержит логику контроллера и модель для управления поставщиками товаров. Основные методы представлены ниже:

- addSupplier – добавить поставщика;

- exists – управление доступностью/недоступностью. Менеджер вправе заблокировать поставщика, если он больше не поставляет товары. При этом блокируются все зависящие от данного типа элементы – товары. Надо быть внимательным при вызове данной процедуры. Можно также отметить, что процедура займет некоторое время на выполнение логики.

Пакет com.omendba.shop.manager.manufacturer.

- Пакет содержит логику контроллера и модель для управления производителями товаров. Основные методы представлены ниже:

- addManufacturer – добавить производителя;

- exists – управление доступностью/недоступностью. Менеджер вправе заблокировать производителя, если он больше не поставляет товары. При этом блокируются все зависящие от данного типа элементы – товары. Надо быть внимательным при вызове данной процедуры. Можно также отметить, что процедура займет некоторое время на выполнение логики.

Пакет com.omendba.shop.manager.goods.

- Пакет содержит логику контроллера и модель для управления товарами. Основные методы представлены ниже:

- addGoods – добавить товар. Все сведения о товаре, включая изображение товара, заносятся в базу данных с помощью данной процедуры;

- getGoods – запрос информации о товаре. Метод запрашивает сведения о товаре по его идентификатору;

- saveGoods – сохранить изменения товара в базе данных.

Пакет com.omendba.shop.manager.news.

- Пакет содержит логику контроллера и модель для управления новостями. Основные методы представлены ниже:

- addNews – добавить новость о товаре. Как только добавляется новость, она сразу отсылается всем подписчикам.

Пакет com.omendba.shop.manager.users.

- Пакет содержит логику контроллера и модель для управления пользователями. Основные методы представлены ниже:

- activationUser – активация и деактивация учетной записи. Если пользователь ведет себя неадекватно в системе, то его можно заблокировать. Как менеджер, так и администратор может это сделать;

- getAmount – получить количество заказов (исключая отменные заказы). С помощью данной процедуры можно получить статистику по работе

ИМ для данного пользователя. Считаются заказы, не только завершённые, но и текущие;

- `getCancel` – получить количество отменённых заказов. Заказы могут отмениться, поэтому их учёт также важен для администратора;

- `getSum` – получить сумму заказов. Данная характеристика может быть важна администратору;

- `getUser` – запрос информации о клиенте. С помощью идентификатора клиента можно узнать все о нём;

- `messageReady` – установить статус, равный единице (прочитано) для данного сообщения. Обработывая сообщения от пользователя, менеджер управляет статусом сообщений. Если сообщение имеет статус, равный нулю, то сообщение не прочитано и не обработано. Прочитав сообщение, менеджер может отправить письмо с ответом клиенту;

- `sendMail` – послать письмо клиенту ИМ. Зная идентификатор клиента, можно отправить письмо ему. Сначала потребуется запросить информацию о нём, затем совершить данную операцию.

Пакет `com.omendba.shop.manager.waybill`.

Пакет содержит логику контролёра и модель для управления накладными товарами. Основные методы представлены ниже:

- `addWaybill` – добавить накладную на товар. Накладная говорит о том, что на склад поступило определенное количество данного товара. Данная процедура прибавляет к текущему количеству товаров на складе количество из накладной.

- `getAllAmount` – получить количество товара на данный момент на складе;

- `getWaybillAmount` – получить количество данного товара по накладной.

Пакет `com.omendba.shop.manager.orders`.

Пакет содержит логику контролёра и модель для управления заказами. Основные методы представлены ниже:

- `changeStatus` – изменение статуса заказа. Заказ имеет статус. Менеджер меняет статус по своему усмотрению. Если заказ завершен или отменен, то система не позволит изменить статус, так как это противоречит здравому смыслу;

- `getCount1` – получить общее количество заказов;



- getCount2 – получить количество законченных заказов;
- getCount3 – получить количество отмененных заказов;
- getCount4 – получить количество заказов в процессе;
- getCount5 – получить общую сумму;
- getCount6 – получить выручку. Выручка считается на основании

суммы заказа и оптовой цены товаров, входящих в заказ;

- getOrder – узнать все о конкретном заказе.

Пакет com.omendba.shop.web.

Данный пакет содержит единственный класс апплет, работающий на стороне клиента.

Основная работа апплета заключается в генерации кода и отправка его сервлету. Данный код – есть код защиты от автоматической регистрации.

Алгоритм работы апплета заключается в следующем:

1. сгенерировать код на стороне клиента;
2. отослать код сервлету. Сервлет, в свою очередь, помещает код в сессию;
3. клиент вводит код в окне ввода и посылает запрос;
4. запрос обрабатывает сервлет, и, получая код вместе с данными, сверяет значение кода из запроса клиента и код, хранящийся в сессии. Если они совпадают, значит, это не робот, а клиент. Иначе не выполняются никакие действия.

Апплет получает на вход от сервлета идентификатор сессии. Так как сервлетов много, апплету обязательно нужно знать идентификатор сервлета, вызвавшего его, иначе он его не найдет. Подключившись к серверу, апплет передает сгенерированный ключ сервлету, тот в свою очередь сохраняет его в сессии и, получая данные от клиента при регистрации, сравнивает ключ в сессии с ключом, введенным пользователем.

Пакет com.omendba.shop.client.activate.

Пакет содержит логику контроллера и модель для управления активацией клиента. Основные методы представлены ниже:

- activateUser – вызывая данный метод, клиент может активировать личную учетную запись. После регистрации клиент все еще не может войти в систему. Для привязки его к почтовому ящику и предназначена данная процедура. После регистрации ему на почту приходит письмо с

ключом активации. Данный ключ генерируется на основе сессии и имеет длину 32 символа. Зайдя по ссылке в письме, клиент активирует свою учетную запись. Только после этого он может входить в систему.

Пакет com.omendba.shop.client.registration.

Пакет содержит логику контроллера и модель для регистрации клиента. Основные методы представлены ниже:

- addUser() – данный метод добавляет клиента в базу данных. Также клиенту на почтовый ящик приходит письмо со ссылкой активации учетной записи.

Пакет com.omendba.shop.client.authorization.

Пакет содержит логику контроллера и модель для авторизации клиента. Основные методы представлены ниже:

- authorizationUser – процедура авторизации клиента. Данная процедура возвращает true, если клиент существует в базе данных и его учетная запись активирована;

- getCount – получить количество заказов данного пользователя. Данные сведения будут отображаться в личном окне пользователя на уровне представления;

- getDispatchmail – подписан ли клиент на рассылку. При добавлении менеджером новости, если значение установлено в единицу, клиент по почте получит новость;

- getLastInput – получить дату последнего входа для клиента. Данные сведения будут отображаться в личном окне пользователя на уровне представления;

- getSum – получить сумму заказов данного пользователя. Данные сведения будут отображаться в личном окне пользователя на уровне представления.

Пакет com.omendba.shop.client.profile.

Пакет содержит логику контроллера и модель для изменения профиля клиента. Основные методы представлены ниже:

- changeProfile – изменить профиль клиента. В личном кабинете клиент вправе изменить свой профиль по своему усмотрению. Данная процедура поможет ему это сделать.

Пакет com.omendba.shop.client.news.

Пакет содержит логику контроллера и модель для изменения профиля клиента. Основные методы представлены ниже:

- `getLastNewsFromDB` – запросить объект `News`, представляющий собой последнюю новость. Заголовок данной новости будет отображаться на главной странице представления в отдельном окне. Также там будет видна дата новости;

- `getNewsSizeN` – запросить количество объектов-новостей в количестве семнадцати (по умолчанию). Именно столько новостей может вместить окно представления для отображения пользователю. Отображаются последние семнадцать новостей. При изменении интерфейса клиента можно изменить данную цифру, а в перспективе позволить пользователю динамически менять данное число.

Пакет `com.omendba.shop.client.message`.

Пакет содержит логику контроллера и модель для управления сообщениями от клиента. Основные методы представлены ниже:

- `addMessageUser` – добавить сообщение от пользователя.

Пакет `com.omendba.shop.client.order`.

Пакет содержит логику контроллера и модель для управления заказами. Пакет содержит класс, реализующий логику корзины. Рассмотрим методы корзины:

- `addBasket` – добавить товар в корзину. Нельзя добавить один и тот же товар дважды. При желании можно лишь изменить его количество в корзине;

- `clearBasket` – очищаем корзину;

- `delBasket` – удалить товар из корзины;

- `getAmountGoodsInBasket` – получить количество наименований товаров в корзине;

- `getCostGoods` – получить полную цену одного товара в корзине с учетом скидки;

- `getCostGoodsAmount` – получить полную цену одного товара, умноженную на количество товаров того же типа в корзине с учетом скидки;

- `getFullCostGoodsInBasket` – получить полную цену всех товаров в корзине с учетом скидки;

- `setAmountGoodsInBasket` – установить новое значение количества данного товара в корзине.

Рассмотрим основные методы модели:

- addOrders – оформить заказ;
- getAmountGoods – получить количество данного товара из базы данных.

Пакет com.omendba.shop.client.findgoods.

Пакет содержит логику контроллера и модель для управления поиском товаров. Основные методы представлены ниже:

- getCountAllViewGoods – получить число, равное количеству запрошенных объектов товаров;
- init – инициализация списка товаров, получаемых в результате поиска;
- left – переместиться влево в списке товаров при постраничном просмотре;
- right – переместиться вправо в списке товаров при постраничном просмотре.

Пакет com.omendba.shop.client.goodsview.

Пакет содержит логику контроллера и модель для управления просмотром товаров. Основные методы представлены ниже:

- getCountAllViewGoods – получить число, равное количеству запрошенных объектов товаров;
- init – инициализация списка товаров;
- left – переместиться влево в списке товаров при постраничном просмотре;
- right – переместиться вправо в списке товаров при постраничном просмотре.

Пакет com.omendba.shop.client.catalogue.

Пакет содержит логику контроллера и модель для управления каталогом товаров. Основные методы представлены ниже:

- getCategories – запросить список категорий из базы данных;
- getCategory – получить категорию из списка категорий;
- getType – получить тип из списка типов;
- getTypes – запросить список типов из базы данных, принадлежащие данной категории.

Пакет com.omendba.shop.client.address.

Пакет содержит логику контроллера и модель для управления адресами клиента. Основные методы представлены ниже:

- `addAddress` – добавить адрес доставки клиента;
- `getAddress` – запросить список адресов из базы данных;
- `getDataAddress` – получить данные об одном адресе.

Пакет `com.omendba.shop.client.historyorder`.

Пакет содержит логику контроллера и модель для управления историей заказов. Основные методы представлены ниже:

- `getContentOrder` – запросить список товаров из базы данных для текущего заказа;
- `getCost` – получить сумму заказа;
- `getDate` – получить дату заказа;
- `getId` – получить номер заказа;
- `getOrder` – запросить список заказов из базы данных для текущего клиента;
- `getStatus` – получить статус заказа.

### 3.7 Реализация уровня контроллера шаблона MVC

Логика работы контроллера хранится в пакете `com.omendba.shop.action`. Пакет содержит четыре класса, каждый из которых отвечает за обработку запросов от клиента, менеджера и администратора. Запросы есть нечто иное, как данные, посланные с помощью форм в браузере.

Класс `IndexAction` обрабатывает запросы клиента; `LoginAction` обрабатывает форму авторизации менеджера; `ManagerAction` обрабатывает запросы, как менеджера, так и администратора; `ReportAction` служит для обработки запросов со страницы отчета.

Все классы определяют логику своей работы в методе `execute`, который должен быть переопределен при наследовании.

### 3.8 Реализация уровня представления шаблона MVC

«Представление» с точки зрения программиста представляет собой совокупность файлов JSP, включающих теги `html`, `jsp`, `struts` и `jstl`.

### 3.8.1 Интерфейс клиента

Для клиента создан интерфейс, предоставляющий ему следующую функциональность:

- Авторизация и регистрация: парольная аутентификация; напоминание пароля по почте; регистрация и защита от автоматической регистрации; необходимость активации профиля пользователя после успешной регистрации.
- Новости.
- Каталог товаров – классификация товаров по типам и категориям.
- Поиск товаров.
- Фильтр и сортировка товаров.
- Личный кабинет: изменение профиля клиента; управление адресами доставки; просмотр истории заказов и содержимого конкретного заказа; обратная связь с менеджером; личный кабинет.
- Корзина: управление товарами в корзине; выбор адреса доставки; оформление заказа.

В результате разработано двадцать пять JSP-страниц. Список страниц представлен ниже:

- bottom – страница, включающая нижнюю часть главной страницы;
- cabinet – страница личного кабинета пользователя;
- catalogue – страница, генерирующая каталог товаров;
- contentsorder – страница, входящая в раздел личных страниц, отвечающая за отображение содержимого заказа;
- delivery – страница, показывающая информацию об оплате;
- filter – страница, отображающая окно фильтра товаров;
- find – страница, отображающая окно поиска;
- goods – страница, окно товаров при просмотре и при поиске товаров;
- heading – страница, отображающая заголовок страницы;
- historyorder – страница, генерирующая историю заказов;
- index – главная страница, включающая все остальные страницы;
- message – страница, окно с сообщениями для клиента;
- messageuser – страница, окно для показа сообщения клиенту;
- minibasket – страница, представляющая собой корзину;
- news – страница, с помощью которой можно просмотреть новости;

- orderother – страница, на которой отображается информации о заказе; пользователь может выбрать адрес доставки и добавить комментарий;
- private – личная страница пользователя;
- profile – страница изменения профиля клиента;
- prompt – страница, управляющая заставкой главной страницы;
- registration – страница регистрации;
- repair – страница, выполняющая некоторую логику;
- about – страница, на которой можно узнать сведения о компании;
- address – страница управления адресами клиента;
- authorization – страница авторизации;
- basket – корзина клиента; данная страница генерирует содержимое корзины.

### 3.8.2 Интерфейс менеджера

Для менеджера создан интерфейс, предоставляющий ему следующую функциональность:

- Управление клиентами.
- Управление новостями.
- Управление поставщиками, производителями, категориями, типами.
- Управление товарами и накладными для товаров.
- Управление заказами клиентов: статистика по заказам; денежная статистика; обработка заказов.
- Просмотр карточки заказа.

В результате разработано двенадцать JSP-страниц. Список страниц представлен ниже:

- manager – главная страница;
- manufacturer – страница управления производителями;
- news – страница управления новостями;
- orders – страница, на которой происходит управление заказами;
- report – страница-отчет о заказе;
- supplier – страница управления поставщиками;
- type – страница управления типами товаров;
- users – страница управления пользователями;
- waybill – страница управления накладными на товары;
- category – страница управления категориями товаров;

- goods – страница управления товарами;
- login – страница входа для менеджера и для администратора.

### 3.8.3 Интерфейс администратора

Для администратора создан интерфейс, предоставляющий ему следующую функциональность:

- Управление менеджерами: добавление новых менеджеров; блокировка/разблокировка менеджеров; смена пароля для менеджеров.
- Доступ к веб-службе. Получение статистики работы ИМ.

### 3.8.4 Интерфейс разработчика

Для программиста был разработан интерфейс, включающий функции:

- Закачка файлов конфигурации на удаленную машину.
- Просмотр файла журнала: время возникшего исключения; метод, в котором произошло исключение; входные параметры для метода.
- Очистка файла журнала.

## 3.9 Вид работы программы

В данном разделе рассмотрим рабочий вид программной системы:

- web-интерфейс разработчика (смотри рисунок 3.3);
- интерфейс веб-службы (смотри рисунок 3.4);
- web-интерфейс администратора (смотри рисунок 3.5);
- web-интерфейс менеджера (смотри рисунок 3.6);

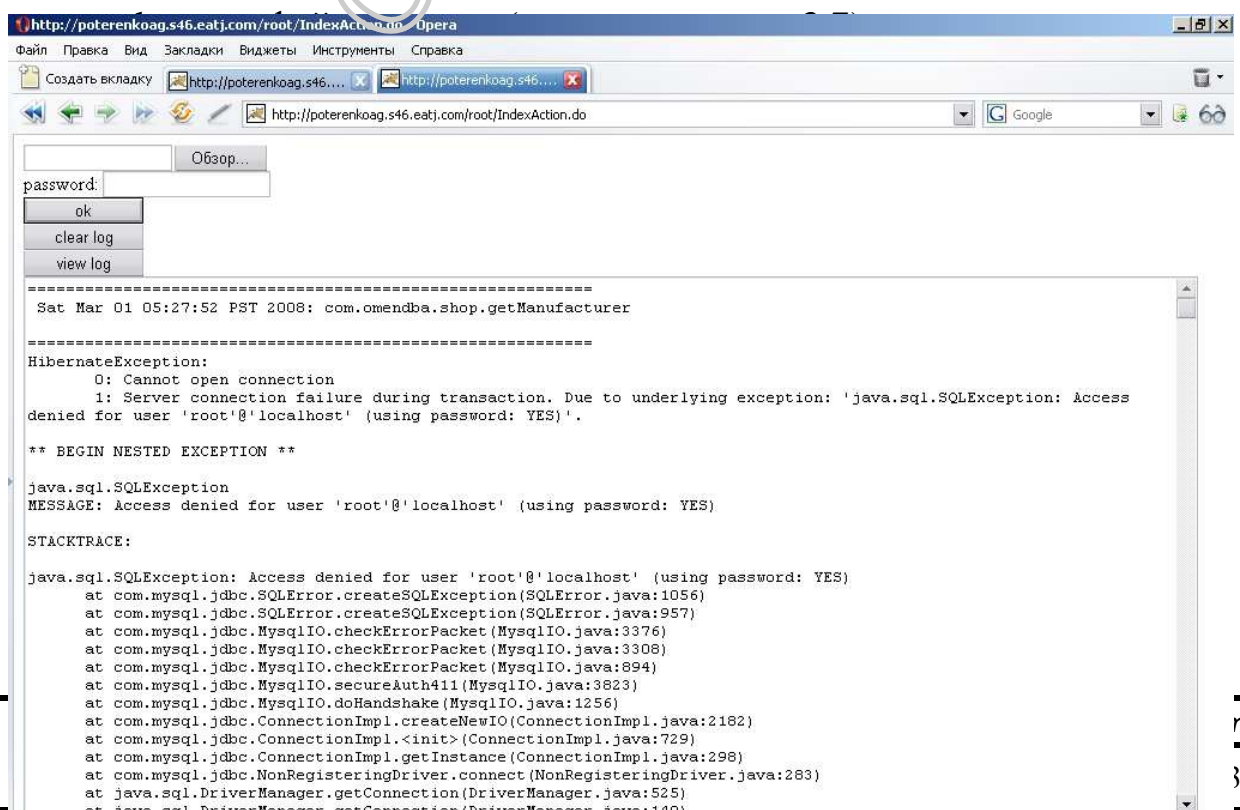




Рисунок 3.3 – Web-интерфейс разработчика



Рисунок 3.5 – Web-интерфейс администратора

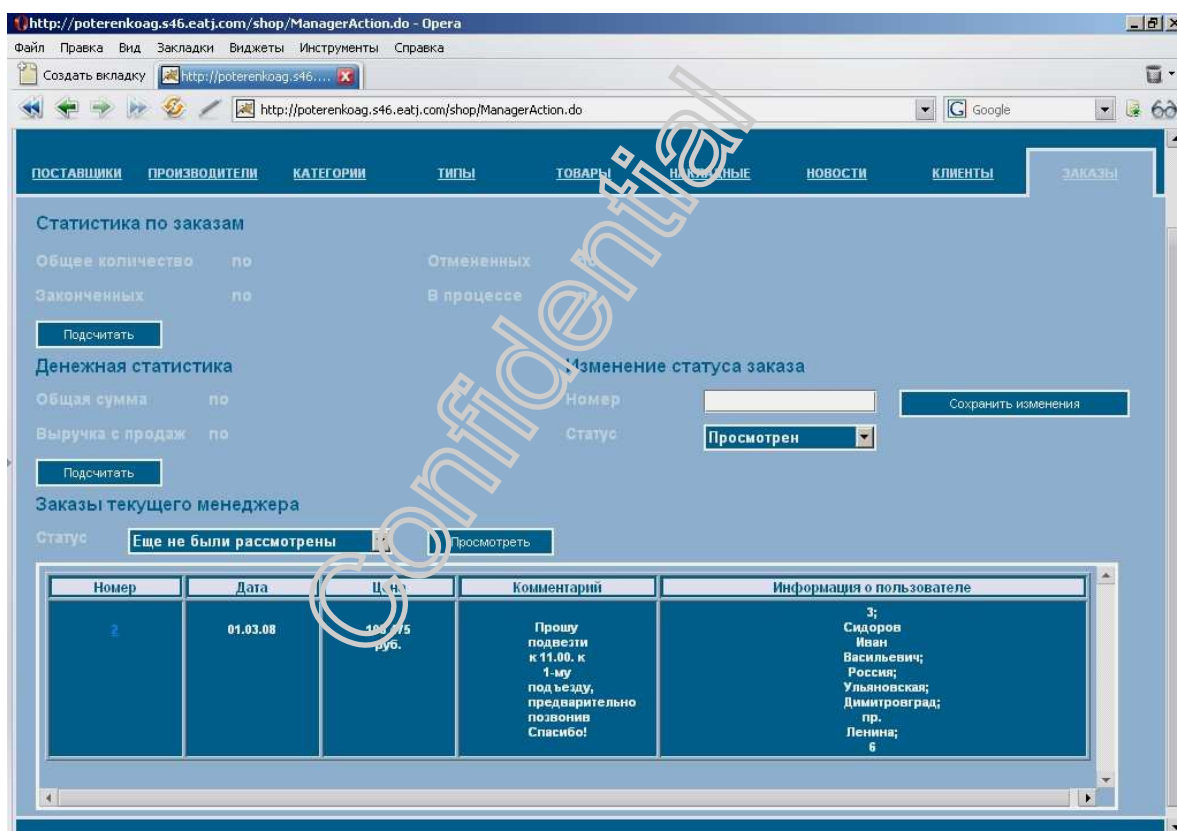


Рисунок 3.6 – Web-интерфейс менеджера

http://poterenkoag.s46.eatj.com/shop/IndexAction.do - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

http://poterenkoag.s46.eatj.com/shop/ManagerAction.do?org.apache.struts.taglib.html.TOKEN=178dde7241c1ac035eb2d72f340998e0&o... - Opera

Файл Правка Вид Закладки Виджеты Инструменты Справка

Создать вкладку http://poterenkoag.s46... X

http://poterenkoag.s46.eatj.com/shop/ManagerAction.do?org.apache.struts.taglib.html.TOKEN=178dde7241c1ac035eb2d72f340998e0&o... Google

ООО "САЛЮТ-АВТО" 01.03.08 [Назад](#)

Заказ № 2

**Данные клиента**

ФИО Сидоров Иван Васильевич

Телефон 43553342

Сумма заказа 108 475 руб.

Сумма доставки \_\_\_\_\_

**Дополнительные сведения**

Прошу подвезти к 11.00. к 1-му подъезду, предварительно позвонив  
Спасибо!

**Адрес доставки**

Страна Россия

Область Ульяновская

Город Димитровград

Улица пр. Ленина

Дом 6

**Товары**

№3 БИ-КСЕНОН ALPHA OPTIMA G5 BI-XENON H4, HB1(9004), HB5(9007)(G) 11 500 руб.

Рисунок 3.8 – Форма отчета о покупке

### Выводы к разделу 3

Автору удалось полностью реализовать программный комплекс. Также удалось провести тестирование методом «Черного ящика», хотя в данном разделе оно не было описано.

Реализация включила в себя не только кодирования логики, но и создания достаточно удобного интерфейса пользователя.

Таким образом, приложение полностью готово к эксплуатации с учетом всех требований заказчика.

Confidential

## 4 Экономический раздел

### 4.1 Обоснование целесообразности разработки дипломной работы

В данном разделе описывается экономическое обоснование целесообразности использования данной работы.

ИМ – сайт, принимающий заказы на материальные товары от посетителей в режиме реального времени. Используются следующий вариант способов оплаты: наличный расчет – товар оплачивается курьеру наличными деньгами при получении покупателем товара.

Выбор того или иного способа оплаты зависит от множества факторов, среди которых можно отметить статус покупателя (физическое или юридическое лицо), особенности товара, цена, способов доставки (невозможно принять наличные при отправке товара за границу).

После отправки заказа с покупателем связывается продавец и уточняет место и время, в которое следует доставить заказ. Доставка осуществляется либо собственной курьерской службой, либо компанией, предоставляющей услуги доставки.

Актуальность разработки портала очевидна и основывается на следующих фактах:

1. Собственная система обеспечит выполнение всех необходимых требований, необходимых для ведения электронного бизнеса в сети Интернет.

2. Организация, которая является объектом внедрения проекта такого рода, получает возможность экономии денежных средств и времени.

3. Имея в штате организации программиста, который является разработчиком системы, появляется возможность снизить затраты на обслуживание системы, а так же при необходимости внесения изменений или дополнений в систему руководителю предприятия не надо будет думать о том, как это сделать.

### 4.2 Оценка затрат на разработку системы

Стоимостная оценка затрат определяется с учетом степени затрат на всех этапах инновационного процесса: проектирование, производство, эксплуатация.

В состав затрат на разработку и реализацию проекта включается стои-

мость всех расходов, необходимых для реализации комплекса работ, составляющих содержание данной разработки. Смета затрат включает в себя следующие статьи расходов:

- заработная плата исполнителям;
- дополнительная заработная плата;
- отчисления во внебюджетные фонды;
- накладные расходы;
- расходы на оборудование.

Рассчитаем заработную плату. Месячный оклад исполнителя составляет:

- инженер-программист ООО «САЛЮТ-АВТО», г. Димитровград.

$$З_{и} = 5000 \text{ руб.}$$

Учитывая участие специалиста в разработке по количеству дней, а также число рабочих дней в месяце (25 дней), найдём заработную плату исполнителя. Результат расчётов сведён в таблицу 4.1.

Таблица 4.1 – Основная заработная плата исполнителя

Исполнитель	Оклад, руб.	Трудоёмкость, дней	Дневная ставка, руб.	Основная з.п., руб.
Инженер	5000	75	200	15000

Полную заработную плату всех исполнителей рассчитаем по формуле 4.1:

$$З_o = \sum_{i=1}^N З_i , \quad (4.1)$$

где  $З_o$  – основная зарплата всех исполнителей;

$З_i$  – заработная плата  $i$ -го исполнителя;

$N$  – количество исполнителей.

Используя данные таблицы 4.1, находим по формуле 4.1  $З_o$  :

$$З_o = 15000 = 15000 \text{ руб.}$$

Рассчитаем дополнительную зарплату.

Подсчитаем величину районного коэффициента  $P_K$  по формуле 4.2, с учётом поясного коэффициента ( $K_{п} = 30$  процентов).

$$P_K = З_o \cdot K_{п} = 15000 \cdot 0.3 = 4500 \text{ руб.} \quad (4.2)$$

Подсчитаем величину отпускных и премиальных отчислений  $O$ , кото-

рые составляют 10 процентов от основной заработной платы:

$$O = 15000 \cdot 0.1 = 1500 \text{ руб.}$$

Рассчитаем дополнительную заработную плату по формуле 4.3, которая складывается из величины отпускных и премиальных отчислений и районного коэффициента:

$$Z_{\text{доп}} = P_K + O = 4500 + 1500 = 7000 \text{ руб.} \quad (4.3)$$

Зная основную и дополнительную зарплату, найдём фонд заработной платы (ФЗП) по формуле 4.4:

$$\Phi_{\text{ЗП}} = Z_{\text{доп}} + Z_O = 7000 + 15000 = 22000 \text{ руб.} \quad (4.4)$$

Отчисления на единый социальный налог составляют 26 процентов от фонда заработной платы и находятся по формуле 4.5.

$$O_{\text{ВФ}} = 22000 \cdot 0.26 = 5720 \text{ руб.} \quad (4.5)$$

Накладные расходы составляют 20 процентов от основной заработной платы, то есть:

$$НР = 15000 \cdot 0.2 = 3000 \text{ руб.}$$

Вычислим затраты на амортизацию вычислительной техники по формуле 4.6:

$$Z_{\text{АМ}} = \frac{(C_{\text{БАЛ}} \cdot K_{\text{АМ}} \cdot T_{\text{РАБ}} \cdot H)}{\Phi_{\Gamma}} \quad (4.6)$$

где  $C_{\text{БАЛ}}$  – балансовая стоимость вычислительной техники (16000 руб.);

$K_{\text{АМ}}$  – коэффициент годовой амортизации техники (12 процентов);

$T_{\text{РАБ}}$  – время работы вычислительной техники (100 дней);

$H$  – количество единиц техники (1);

$\Phi_{\Gamma}$  – эффективный фонд времени оборудования (220 дней).

$$Z_{\text{АМ}} = (16000 \cdot 0.12 \cdot 100 \cdot 1) / 220 = 872 \text{ руб.}$$

Полная смета затрат.

На основе вышеприведённых расчётов составим полную смету затрат на дипломную работу (смотри таблицу 4.2).

Таблица 4.2 – Смета затрат на выполнение дипломной работы

Статья расходов	Сумма, руб.
Заработная плата исполнителям	15000
Дополнительная заработная плата	7000
Отчисления во внебюджетные фонды	5720
Накладные расходы	3000
Амортизационные расходы	872
Итого	31592

#### 4.3 Оценка эксплуатационных затрат

К эксплуатационным относятся затраты, связанные с обеспечением нормального функционирования АСУ и их функциональных подсистем, поэтому эти затраты называют также текущими затратами. Это могут быть затраты на ведение информационной базы, эксплуатацию комплекса технических средств, эксплуатацию систем программно-математического обеспечения, реализацию технологического процесса обработки информации по задачам, эксплуатация системы в целом.

Определим состав сотрудников. Система полностью автоматизирована. Это значит, что на ее обслуживание нужен лишь программист для доработки системы в случае необходимости. Следует заметить, что программист может не работать на предприятии, но он проживает в том же городе, где находится предприятие, и разово предоставляет свои услуги за определенную плату.

При использовании проекта руководитель предприятия, в случае возникновения или выявления неполадок в работе системы, а так же в случае необходимости внесения изменений в систему, получает возможность оперативного реагирования и решение возникших проблем. При использовании аналогов: в лучшем случае решение будет в виде ответов на вопросы по электронной почте или же консультации по телефону, которые малоэффективны.

Разработчик системы может обслуживать данную систему, находясь в другом городе или стране. Руководитель предприятия может позвонить разработчику и сообщить о некорректном поведении системы и перевести на счет программиста деньги. Программист-разработчик перепроверит журнал системы и сможет: либо перепрограммировать код в случае обнаружения ошибки в нем, либо объяснить руководителю о том, что сотруд-



ники сами неправильно работают с системой, указав на то, где они неправильно ввели данные, например.

Пусть каждое обращение к программисту стоит 500 руб. Допустим, что потребность в программисте возникает 2 раза в месяц, т.о. затраты на программиста составляют  $Z_{II} = 1000$  в месяц. Тогда годовой эффект от использования проекта составит 12000 руб.

Подсчитаем годовую себестоимость работ по обеспечению функционирования разработанной системы.

Годовая себестоимость работ в случае использования предлагаемого проекта вычисляется по формуле 4.7:

$$C_1 = 12 \cdot Z_{II} = 12000 \text{ руб.} \quad (4.7)$$

где  $Z_{II}$  – месячные затраты на услуги программиста.

Теперь рассмотрим стоимость обслуживания любого аналога.

Стоимость консультаций по электронной почте и телефону стоят 2000 руб. в месяц. Аналогичный проект на рынке программного обеспечения стоит порядка 60000 руб.

Годовая себестоимость работ в случае использования аналога, без учета стоимости системы также вычисляется по формуле 4.7:

$$C_2 = 12 \cdot Z_{СПЕЦ} = 24000 \text{ руб.}$$

где  $Z_{СПЕЦ}$  – месячные затраты на услуги специалиста из компании конкурента.

Итог:

1. Годовая себестоимость проекта: 12000 руб.

2. Годовая себестоимость аналога: 24000 руб.

#### 4.4 Оценка затрат на внедрение разработки

Для предприятия, учреждения, организации, решившей организовать систему учета и контроля сетевого трафика с помощью данной разработки, понадобятся компоненты, указанные в таблице 4.3.

Таблица 4.3 – Затраты на внедрение разработки

Необходимые компоненты	Стоимость, руб.
Подключение к Интернет	1000-3000
Покупка доменного имени	1000
Покупка хостинга под разработку (в год)	1500

Итого, если на предприятии не подключено к сети Интернет, не имеет хостинга и не имеет доменного имени, капитальные вложения (К) на внедрение системы составят от 3500 до 5500 руб.

#### 4.5 Оценка экономической эффективности разработки

Расчет годового экономического эффекта в случае экономии при выполнении одной и той же работы производится по формуле 4.8.

$$\mathcal{E} = ((C_2 - C_1) - E \cdot K), \quad (4.8)$$

где  $C_1, C_2$  – себестоимость работ, выполняемых по проектному варианту и варианту аналога, руб.;

$E$  – нормативный коэффициент экономической эффективности ( $E=0,33$ );

$K$  – капитальные вложения на проектирование и внедрение проекта, руб.

В предыдущих разделах были определены следующие значения:

$$C_1 = 12000 \text{ руб.}$$

$$C_2 = 24000 \text{ руб.}$$

$$K = 5500 \text{ руб. (максимум)}$$

Отсюда по формуле 4.8:

$$\mathcal{E} = (24000 - 12000) - 0,33 \cdot 5500 = 10185 \text{ рублей.}$$

Расчетный коэффициент экономической эффективности показывает величину экономии на текущих эксплуатационных затратах, образующихся за счет внедрения новой системы, на один рубль единовременных вложений и рассчитывается по формуле 4.9.

$$E = \frac{\mathcal{E}}{K} \quad (4.9)$$

$$E = 10185/5500 = 1,85$$

Так как  $E > 0,33$ , то проектирование и внедрение проектируемой системы эффективно.

Условный расчетный срок окупаемости капитальных вложений опре-

деляется как величина, обратная расчетному коэффициенту экономической эффективности определяется по формуле 4.10:

$$T_{ок} = \frac{K}{\Xi}, \quad (4.10)$$

где  $T_{ок} = 5500/10185 = 0,54$  года или примерно 197 дней;

$T_K$  – нормальный срок окупаемости равен трем годам.

В данном случае  $T_K > T_{ок}$ , поэтому условие выполняется.

Выводы к разделу 4

Автор в данном разделе подсчитал «экономическую сторону проекта». Были найдены такие важные характеристики как: оценка затрат на разработку системы, оценка эксплуатационных затрат, оценка затрат на внедрение разработки, оценка экономической эффективности разработки.

Удалось обосновать целесообразность разработки и внедрения дипломной работы.

Можно сделать единственный вывод: разработка и внедрение системы являются эффективным решением.

## 5 Безопасность жизнедеятельности

Безопасность жизнедеятельности [10] – совокупность мероприятий, направленных на защиту человека, сохранение его здоровья под влиянием внешних факторов, разработку методов защиты путем уменьшения влияния вредных факторов до соответствующих значений.

Каждое рабочее место программиста характеризуется воздействием на него опасных и вредных факторов.

Важные основные моменты курса безопасности жизнедеятельности:

- обнаружение и изучение факторов окружающей среды, отрицательно влияющих на здоровье человека;
- ослабление действия этих факторов до безопасных пределов или исключение их, если это возможно;
- ликвидация последствий катастроф и стихийных бедствий.

### 5.1 Общие вопросы безопасности труда

#### 5.1.1 Лица и их обязанности по вопросам безопасности труда

Рассматривая статью 212 Трудового Кодекса Российской Федерации федерального закона от 30 июня 2006 года, можно увидеть, что работодатель должен реализовать условия, согласно которым обеспечивается:

- применение сертифицированных средств индивидуальной и коллективной защиты работников;
- условия труда на рабочем месте соответствующие требованиям охраны труда;
- безопасность работников при эксплуатации оборудования, осуществлении технологических процессов;
- режим труда и отдыха работников в соответствии с трудовым законодательством и иными нормативными правовыми актами, содержащими нормы трудового права;
- обучение безопасным методам и приемам выполнения работ и оказанию первой помощи, пострадавшим на производстве;
- недопущение к работе лиц, не прошедших в установленном порядке обучение и инструктаж по охране труда.

Рассматривая статью 214 Трудового Кодекса Российской Федерации федерального закона от 30 июня 2006 года, можно увидеть, что работник должен:

- адекватно применять средства индивидуальной и коллективной защиты;
- соблюдать требования охраны труда;
- обучаться безопасным методам и приемам выполнения работ и оказанию первой помощи, пострадавшим на производстве;
- незамедлительно извещать своего непосредственного руководителя о любой ситуации, угрожающей жизни и здоровью людей, о каждом несчастном случае, происшедшем на производстве, или об ухудшении состояния своего здоровья;
- проходить предварительные (при поступлении на работу) и периодические (в течение трудовой деятельности) медицинские, а также проходить внеочередные медицинские осмотры по направлению работодателя в случаях, предусмотренных федеральными законами.

#### 5.1.2 Обучение работающих безопасности труда

Согласно статье 225 главы 36 трудового кодекса РФ производственный процесс предприятия начинается с правильного и качественного проведения инструктажа.

Основополагающим нормативным документом по организации и проведению инструктажа является ГОСТ 12.0.004-90 «ССБТ Организация обучения по безопасности труда. Общие положения».

Различают пять видов инструктажа: вводный, первичный, повторный, внеплановый и целевой. Рассмотрим вкратце их особенности и различия.

Вводный инструктаж проводится со всеми без исключения работниками, принимаемыми на работу, а также с учащимися, прибывшими для производственного обучения или практики, и прочими людьми, впервые выполняющими работы на данном предприятии. Вводный инструктаж имеет своей целью ознакомление работника с действующим трудовым распорядком, структурой предприятия, используемым оборудованием и технологией производства, мерами пожарной безопасности, основными инструкциями охраны труда и прочими правилами.

Первичный инструктаж проводится на рабочем месте индивидуально с каждым работником, впервые приступающим к выполнению данного вида работ, включая командированных сотрудников других организаций, принимающих участие в производственном процессе, работников, выполняющих работы на территории организации по договору подряда, и т.п. Важность данного вида инструктажа очевидна: он имеет большую практическую значимость для обеспечения безопасности труда.

Повторный инструктаж проводится периодически по программе первичного инструктажа для закрепления и расширения полученных ранее знаний по вопросам охраны труда, а также для их проверки. Повторный инструктаж может проводиться либо индивидуально с каждым сотрудником, либо с группой работников, использующих однотипное оборудование в пределах одного рабочего места.

Внеплановый инструктаж проводится при изменении условий труда, действующих нормативных актов по охране труда, при выявлении нарушений работниками правил охраны труда, приведших или могущих привести к несчастным случаям или аварии, а также в некоторых других случаях. Внеплановый инструктаж проводится индивидуально или с группой лиц, работающих по одной профессии.

Целевой инструктаж проводится перед выполнением разовых работ, не связанных с основной специальностью и трудовыми обязанностями сотрудника, при ликвидации последствий аварий.

### 5.1.3 Планирование и финансирование мероприятий по охране труда

Планирование и финансирование мероприятий по охране труда [11] описывается в статье 226 трудового кодекса РФ. Номенклатура мероприятий по охране труда является основой для подготовки комплексного плана улучшения условий труда, охраны труда и санитарно оздоровительных мероприятий, а также для разработки раздела коллективного договора – соглашения об охране труда.

Мероприятия по охране труда: модернизация технологического, приведение производственных зданий и сооружений в соответствие с требованиями охраны труда; приведение уровней шума, вибрации и др. вредных воздействий на рабочих местах в соответствии с требованиями стандартов.

Источники финансирования: фонд целевого назначения; амортизационный фонд; банковский кредит; государственные и нецентрализованные капиталовложения, когда затраты носят капитальный характер.

Разработанные администрацией и обсуждены на заседании комитета и в трудовых коллективах проекты плановых номенклатур мероприятий по охране труда за подписями директора направляются в вышестоящие хозяйственные органы для рассмотрения и утверждения запроектированных ассигнований на их осуществление.

Статья 16 и 17. Планирование и финансирование предприятий по охране труда.

Финансирование охраны труда осуществляется за счет прибыли предприятий, а также фондов охраны труда. Фонды охраны труда формируются на трех уровнях: федеральный, территориальный и фонд предприятия.

Федеральный фонд – за счет целевых ассигнований, выделяемых советом министров, правительством РФ, правительством республик в составе РФ, части средств фондов охраны труда, предприятий, суммой штрафов, налагаемых на должностные лица за нарушение законодательства по охране труда, отчислений из фонда государственного социального страхования, добровольных отчислений предприятий, добровольных взносов граждан и прочих поступлений.

Территориальный фонд формируется за счет ассигнований из бюджетов, национально-государственных, административно-территориальных образований РФ, части средств фондов охраны труда предприятий, расположенных на соответствующих территориях, добровольных отчислений предприятий, добровольных взносов граждан и прочих поступлений.

Фонд предприятия: ежегодно предприятие выделяет на охрану труда необходимые средства в объемах определяемых коллективными договорами или соглашениями.

Помимо перечисленных фондов охраны труда могут создаваться общественные фонды охраны труда.

Система управления предусматривает коллективную ответственность за конечный результат работы в области охраны труда, то есть от рабочего до инженерно-технического работника и руководителя предприятия.

Директор отвечает за распределение денежных средств за выполнение коллективного договора и соглашения, контролирует режим работы пред-

приятия, прием рабочих на работу, структурные подразделения и заслушивает отчеты по охране труда.

#### 5.1.4 Виды ответственности за нарушение правил по безопасности труда

В статье 419 ТК РФ установлены виды ответственности за нарушение трудового законодательства и иных нормативных правовых актов, содержащих нормы трудового права: «Лица, виновные в нарушении трудового законодательства и иных актов, содержащих нормы трудового права, привлекаются к дисциплинарной и материальной ответственности в порядке, установленном настоящим Кодексом и иными федеральными законами, а также привлекаются к гражданско-правовой, административной и уголовной ответственности в порядке, установленном федеральными законами».

За нарушение трудового законодательства, правил и норм по охране труда, невыполнении коллективных договоров должностные лица привлекаются к ответственности:

- Дисциплинарная ответственность – наложение взыскания на виновное должностное лицо вышестоящим по подчиненности должностным лицом или органов (замечание, выговор, понижение в должности, освобождение от занимаемой должности).

- Административная ответственность – наложение денежного штрафа на лиц, не принявших мер к соблюдению правил по технике безопасности. Право налагать штраф предоставлено органам государственного надзора.

- Уголовная ответственность – лишение свободы, наложение штрафа, освобождение от занимаемой должности и др. ответственность за нарушение охраны труда возложена, как правило, отдела, лаборатории.

Вышестоящие органы несут уголовную ответственности в тех случаях, если:

- не созданы необходимые условия по обеспечению безопасных условий труда;
- не приняты меры для устранения нарушения правил охраны труда;
- не организованы должным образом безопасные условия труда.
- Материальная ответственность возникает в тех случаях, когда предприятие несет ответственность за ущерб, причиненный рабочим или слу-



жащим увечьем или иным повреждением здоровья в период выполнения ими своих трудовых обязанностей.

Возмещение ущерба состоит в выплате потерпевшему денежных сумм в размере той части заработка, который он лишился вследствие утраты трудоспособности.

## 5.2 Обеспечение комфортных и безопасных условий труда программиста

### 5.2.1 Характеристика условий труда программиста

Работа с компьютером характеризуется значительным умственным напряжением и нервно-эмоциональной нагрузкой операторов, высокой напряженностью зрительной работы и достаточно большой нагрузкой на мышцы рук при работе с клавиатурой ЭВМ. Большое значение имеет рациональная конструкция и расположение элементов рабочего места, что важно для поддержания оптимальной рабочей позы человека-оператора.

В процессе работы с компьютером необходимо соблюдать правильный режим труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

### 5.2.2 Организация рабочего места программиста

Рабочее место и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологическим требованиям. Должны быть соблюдены основные условия: оптимальное размещение оборудования, входящего в состав рабочего места, достаточное рабочее пространство.

Создание комфортных условий труда и правильное эстетическое оформление рабочих мест на производстве имеет большое значение для облегчения труда, положительно влияющего на производительность труда.

Эргономическими аспектами [12] проектирования видеотерминальных рабочих мест, в частности, являются: высота рабочей поверхности,

размеры пространства для ног, требования к расположению документов на рабочем месте, характеристики рабочего кресла, требования к поверхности рабочего стола.

Главными элементами рабочего места программиста являются стол и кресло. Основным рабочим положением является положение сидя.

Рабочая поза сидя вызывает минимальное утомление программиста. Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, средств труда и документации. То, что требуется для выполнения работ чаще, расположено в зоне легкой досягаемости рабочего пространства.

Для комфортной работы стол должен удовлетворять следующим условиям: высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотники; нижняя часть стола сконструирована так, чтобы программист мог удобно сидеть, не был вынужден поджимать ноги; поверхность стола должна обладать свойствами, исключающими появление бликов в поле зрения программиста; высота рабочей поверхности рекомендуется в пределах 680-760мм. Высота поверхности, на которую устанавливается клавиатура, должна быть около 350мм.

Большое значение придается характеристикам рабочего кресла. Так, рекомендуемая высота сиденья над уровнем пола находится в пределах 420-550мм. Поверхность сиденья мягкая, передний край закругленный, а угол наклона спинки – регулируемый.

Необходимо предусматривать при проектировании возможность различного размещения документов: сбоку от видеотерминала, между монитором и клавиатурой и т.п. Кроме того, в случаях, когда видеотерминал имеет низкое качество изображения, например заметны мелькания, расстояние от глаз до экрана делают больше (около 700мм), чем расстояние от глаза до документа (300-450мм).

Большое значение также придается правильной рабочей позе пользователя. При неудобной рабочей позе могут появиться боли в мышцах, суставах и сухожилиях. Требования к рабочей позе пользователя видеотерминала следующие:

- голова не должна быть наклонена более чем на 20 градусов;
- плечи должны быть расслаблены;

- локти – под углом 80...100 градусов;
- предплечья и кисти рук – в горизонтальном положении.

Причина неправильной позы пользователей обусловлена следующими факторами: нет хорошей подставки для документов, клавиатура находится слишком высоко, а документы – низко, некуда положить руки и кисти, недостаточно пространство для ног.

- В целях преодоления указанных недостатков даются общие рекомендации: лучше передвижная клавиатура; должны быть предусмотрены специальные приспособления для регулирования высоты стола, клавиатуры и экрана, а также подставка для рук.

Существенное значение для производительной и качественной работы на компьютере имеют размеры знаков, плотность их размещения, контраст и соотношение яркостей символов и фона экрана. Если расстояние от глаз оператора до экрана дисплея составляет 60...80 см, то высота знака должна быть не менее 3мм, оптимальное соотношение ширины и высоты знака составляет 3:4, а расстояние между знаками – 15...20 процентов их высоты. Соотношение яркости фона экрана и символов - от 1:2 до 1:15.

- Во время пользования компьютером медики советуют устанавливать монитор на расстоянии 50-60 см от глаз. Специалисты также считают, что верхняя часть видео-дисплея должна быть на уровне глаз или чуть ниже. Когда человек смотрит прямо перед собой, его глаза открываются шире, чем когда он смотрит вниз. За счет этого площадь обзора значительно увеличивается, вызывая обезвоживание глаз. К тому же если экран установлен высоко, а глаза широко открыты, нарушается функция моргания. Это значит, что глаза не закрываются полностью, не омываются слезной жидкостью, не получают достаточного увлажнения, что приводит к их быстрой утомляемости.

- Создание благоприятных условий труда и правильное эстетическое оформление рабочих мест на производстве имеет большое значение, как для облегчения труда, так и для повышения его привлекательности, положительно влияющей на производительность труда.

### 5.2.3 Обеспечение микроклиматических параметров

Параметры микроклимата могут меняться в широких пределах, в то время как необходимым условием жизнедеятельности человека является поддержание постоянства температуры тела благодаря терморегуляции, т.е. способности организма регулировать отдачу тепла в окружающую среду. Принцип нормирования микроклимата – создание оптимальных условий для теплообмена тела человека с окружающей средой.

Вычислительная техника является источником существенных тепло-выделений, что может привести к повышению температуры и снижению относительной влажности в помещении. В помещениях, где установлены компьютеры, должны соблюдаться определенные параметры микроклимата. В санитарных нормах СанПиН-2.2.1/1.1.1278-03 установлены величины параметров микроклимата, создающие комфортные условия. Эти нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера производственного помещения (смотри таблицу 5.1).

Объем помещений, в которых размещены работники вычислительных центров, не должен быть меньше 20м<sup>3</sup>/человека с учетом максимального числа одновременно работающих в смену. Нормы подачи свежего воздуха в помещения, где расположены компьютеры, приведены ниже (смотри таблицу 5.2).

Таблица 5.1 – Параметры микроклимата для помещений, где установлены компьютеры

Период года	Параметр микроклимата	Величина
Холодный	Температура воздуха в помещении	22...24°C
	Относительная влажность	40...60%
	Скорость движения воздуха	до 0,1м/с
Теплый	Температура воздуха в помещении	23...25°C
	Относительная влажность	40...60%
	Скорость движения воздуха	0,1...0,2м/с

Таблица 5.2 – Нормы подачи свежего воздуха в помещения, где расположены компьютеры

Характеристика помещения	Объемный расход подаваемого в помещение свежего воздуха, м <sup>3</sup> /на одного человека в час
--------------------------	---

Объем до 20м <sup>3</sup> на человека 20...40м <sup>3</sup> на человека Более 40м <sup>3</sup> на человека	Не менее 30 Не менее 20 Естественная вентиляция
--	---

Для обеспечения комфортных условий используются как организационные методы (рациональная организация проведения работ в зависимости от времени года и суток, чередование труда и отдыха), так и технические средства (вентиляция, кондиционирование воздуха, отопительная система).

#### 5.2.4 Организация освещенности

Требования к организации производственного освещения отражены в санитарных нормах и правилах СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

Помещения для эксплуатации ПЭВМ должны иметь естественное и искусственное освещение. Эксплуатация ПЭВМ в помещениях без естественного освещения допускается только при соответствующем обосновании и наличии положительного санитарно-эпидемиологического заключения, выданного в установленном порядке.

Естественное и искусственное освещение должно соответствовать требованиям действующей нормативной документации. Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. Оконные проемы должны быть оборудованы регулирующими устройствами типа: жалюзи, занавесей, внешних козырьков и др.

Искусственное освещение в помещениях эксплуатации ПЭВМ осуществляется системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, разрешено применение системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов). Коэффициент пульсации не должен превышать 5%.

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк, также допускается установка светильников местного освещения для подсветки документов, но с таким условием, чтобы оно не создавало бликов на поверхности экрана и не увеличивало освещенность экрана более чем на 300 лк.

В качестве источников света при искусственном освещении должны применяться преимущественно люминесцентные лампы типа ЛСПО. Лампы данного типа отличаются наличием трехслойного люминоформа, пониженным содержанием ртути. Их светоотдача не изменяется в течении всего срока эксплуатации, они являются бесшумными лампами, с частотой мерцания меньше 5%. При устройстве отраженного освещения в административно-общественных помещениях допускается применение металлогалогенных ламп мощностью до 250 Вт. Допускается применение ламп накаливания в светильниках местного освещения.

Общее освещение следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядом расположении ПЭВМ. При периметральном расположении компьютеров линии светильников должны располагаться локализовано над рабочим столом, ближе к его переднему краю, обращенному к оператору.

Для обеспечения нормируемых значений освещенности в помещениях использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

#### 5.2.5 Защита персонала ИВО от шума

Шум [13] – нежелательный для человека звук, мешающий восприятию полезных сигналов.

В помещениях с низким уровнем общего шума источниками шумовых помех могут стать вентиляционные установки, кондиционеры или периферийное оборудование для ЭВМ (плоттеры, принтеры). Длительное воздействие этих шумов отрицательно сказывается на эмоциональном состоянии персонала.

Согласно СанПиН 2.2.2/2.4.1340-03 фоновый уровень шума не должен превышать 40 дБ, а во время работы на ПЭВМ 50 дБ. В качестве мер по снижению шума можно предложить следующее:

- облицовка потолка и стен звукопоглощающим материалом;
- экранирование рабочего места;
- установка в компьютерных помещениях оборудования, производящего минимальный шум;
- рациональная планировка помещения.

Разработаем программу расчета уровня шума и проверим соответствие уровня шума в вычислительной лаборатории нормируемым показателям.

Уровень звукового давления в помещении рассчитывается по формуле 5.1

$$L = L_p + 10 \lg \left( 1 + \frac{4}{\alpha_1 S_1} \right), \quad (5.1)$$

где  $L_p$  – суммарный уровень звукового давления, создаваемый в помещении установленным оборудованием, дБ;

$\alpha_1$  – коэффициент звукопоглощения поверхностей ( $\alpha_1 = 0.12$ );

$S_1$  – площадь помещения, отражающая звук, м<sup>2</sup>.

Площадь помещения рассчитывается по формуле 5.2.

$$S_1 = S_{\text{ПОТОЛКА}} + S_{\text{ПОЛА}} + S_{\text{СТЕН}} \quad (5.2)$$

Если уровни звукового давления установленного оборудования одинаковы, то  $L_p$  определяется по формуле 5.3.

$$L_p = L_0 + 10 \lg n, \quad (5.3)$$

где  $L_0$  – уровень шума от единицы оборудования, дБ (см. таблицу 5.4);

$n$  – количество установленного оборудования.

Если машины имеют разные уровни звукового давления, то сначала определяются уровни звукового давления, создаваемые каждой группой машин, то есть:

$$L_{p1} = L_{01} + 10 \lg n_1$$

...

$$L_{pN} = L_{0N} + 10 \lg n_N$$

Разность уровней шума от первых групп машин составляет:

$$L_{p1} - L_{p2} = \Delta_1$$

По величине  $\Delta_1$  определяют значение  $\Delta L_1$  по таблице 5.3.

Таблица 5.3 – Числовые значения добавки  $\Delta L$

Разность уровней звукового давления двух источников	0	1	2,5	4	6	8	10
$\Delta L$	3	2,5	2	1,5	1	0,5	0

Расчет уровня звукового давления от двух групп машин согласно правилу:

$$L_{p1} > L_{p2} \quad L_{p1,2} = L_{p1} + \Delta L_1$$

$$L_{p2} > L_{p1} \quad L_{p1,2} = L_{p2} + \Delta L_1$$

Определяем разницу  $L_{p1,2} - L_{p3} = \Delta_2$ .

По величине  $\Delta_2$  определяют значение  $\Delta L_2$  по таблице 5.3.

Добавку  $\Delta L_2$  прибавляем к большему уровню шума:

$$L_p = L_{p1,2} + \Delta L_2$$

Сравнение  $L$  с ПДУ (50 дБ). Если  $L$  больше, то необходимо предусмотреть мероприятия по снижению шума (например, облицовка звукоизолирующим материалом).

Таблица 5.4 – Уровень звукового давления от источников

Оборудование	дБ
Матричный принтер	65
Струйный принтер	50
Системный блок	50
Лазерный принтер	45
Монитор	20

Результаты работы программы (смотри рисунок 5.2) и вид работы программы (смотри рисунок 5.1) изображены ниже.

Уровень звукового давления в дБ 64,326

Площадь помещения, отражающая звук

Площадь потолка (м.кв.) 70

Площадь пола (м.кв.) 70

Площадь стен (м.кв.) 120

Коэффициент звукопоглощения поверхностей 0.12

Лаборатория № 40

Оборудование	Уровень шума (дБ)	Количество (шт)
Системный блок	50	24
Монитор	20	24
Хаб	13	1



Рисунок 5.1 – Программа для расчета заземления

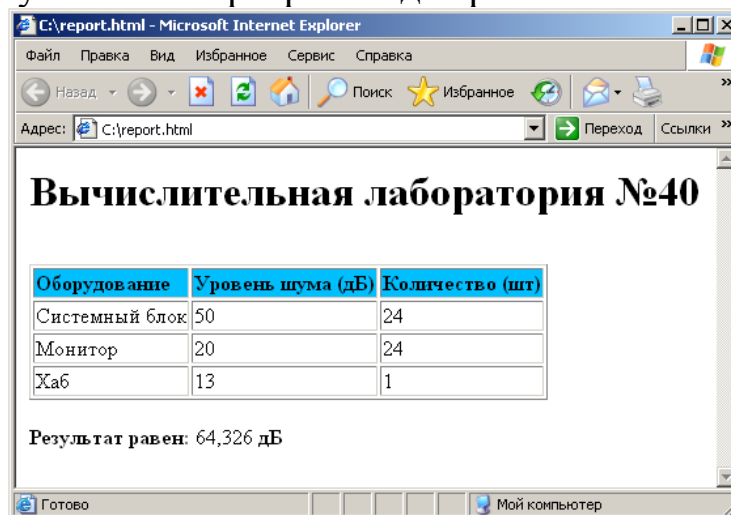


Рисунок 5.2 – Результаты работы программы

#### 5.2.6 Защита персонала ИВО от опасных и вредных излучений

Большинство ученых считают, что как кратковременное, так и длительное воздействие всех видов излучения от экрана монитора не опасно для здоровья персонала, обслуживающего компьютеры. Исчерпывающих данных относительно опасности воздействия излучения от мониторов на работающих за компьютерами не существует.

Допустимые значения параметров, не ионизирующих электромагнитных излучений от монитора компьютера представлены в табл. 5.5.

Максимальный уровень рентгеновского излучения на рабочем месте оператора компьютера обычно не превышает 10мкбэр/ч, а интенсивность ультрафиолетового и инфракрасного излучений от экрана монитора лежит в пределах 10...100мВт/м<sup>2</sup>.

Таблица 5.5 Допустимые значения параметров неионизирующих электромагнитных излучений (в соответствии с СанПиН-2.2.1/1.1.1278-03)

Наименование параметра	Допустимые значения
Напряженность электрической составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора	10В/м
Напряженность магнитной составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора	0,3А/м

<p>Напряженность электростатического поля не должна превышать:</p> <ul style="list-style-type: none"> <li>• для взрослых пользователей</li> <li>• для детей дошкольных учреждений и учащихся средних специальных и высших учебных заведений</li> </ul>	<p>20кВ/м</p> <p>15кВ/м</p>
--	-----------------------------

Для снижения воздействия этих видов излучения рекомендуется при-  
 менять мониторы с пониженным уровнем излучения (ТСО-99, ТСО-03),  
 соблюдать регламентированные режимы труда и отдыха.

### 5.2.7 Обеспечение электробезопасности

Все ремонтные работы должны выполняться на отключенном оборудо-  
 вании.

Компьютер без заземления вполне работоспособен и, как правило, с  
 успехом выполняет возложенные на него пользователем задачи. В  
 большинстве блоков питания компьютеров на входе стоит элементарный  
 фильтр, состоящий из двух конденсаторов, задача которого сводится к  
 тому, чтобы не пропустить высокочастотную составляющую. В результа-  
 те, в зависимости от емкости конденсаторов, получаем на корпусе  
 компьютера потенциал порядка 100 В относительно фазного и нулевого  
 провода. Иначе говоря, при определенных условиях при прикосновении к  
 корпусу компьютера можно получить удар электрическим током.

Основным способом обеспечения электробезопасности является пра-  
 вильное выполнение защитного заземления.

В качестве естественных заземлителей рекомендуется использовать:  
 проложенные в земле водопроводные и другие металлические трубопро-  
 воды, канализации и центрального отопления; обсадные трубы скважин;  
 металлические и железобетонные конструкции зданий и сооружений, на-  
 ходящиеся в соприкосновении с землей; металлические шунты гидротех-  
 нических сооружений, водоводы, затворы и т. п.; свинцовые оболочки ка-  
 белей, проложенных в земле. Алюминиевые оболочки кабелей не допус-  
 кается использовать в качестве естественных заземлителей. Если оболоч-  
 ки кабелей служат единственными заземлителями, то в расчете заземляю-  
 щих устройств они должны учитываться при количестве кабелей не менее  
 двух; заземлители опор ВЛ, соединенные с заземляющим устройством  
 электроустановки при помощи грозозащитного троса ВЛ, если трос не

изолирован от опор ВЛ; нулевые провода ВЛ до 1 кВ с повторными заземлителями при количестве ВЛ не менее двух; рельсовые пути магистральных неэлектрофицированных железных дорог и подъездные пути при наличии преднамеренного устройства перемычек между рельсами.

Заземлителем должны быть связаны с магистралями заземлений не менее чем двумя проводниками, присоединенными к заземлителю в разных местах.

Для искусственных заземлителей следует применять сталь. Искусственные заземлители не должны иметь окраски.

Не следует располагать заземлители в местах, где земля подсушивается под действием тепла трубопроводов и т. п.

Траншеи для горизонтальных заземлителей должны заполняться однородным грунтом, не содержащим щебня и строительного мусора.

В качестве искусственных заземлителей допускается применение заземлителей из электропроводящего бетона.

#### 5.2.8 Обеспечение пожарной безопасности

Потенциальная опасность пожаров на предприятиях ИВО существует постоянно, но благодаря примитивным мероприятиям пожары – редкое явление.

Высокая опасность пожара связана с наличием большого количества и широким применением различных электроустановок. Основными причинами пожаров является неисправность электрооборудования, нарушение правил эксплуатации ВДТ и ЭВМ, как следствие искры, короткие замыкания и др.

Для защиты от пожаров должны применяться различные виды сигнализации, которые должны быстро и точно сообщать о пожаре с указанием места возгорания. На предприятиях ИВО как правило применяют автоматические электрические пожарные сигнализация. Она включает пожарные сигнализации, установленные в защищаемых помещениях и включенные в сигнальную линию; звуковые и световые средства сигнализации, приемную станцию и блок питания. По требованиям ГОСТа противопожарная система должна быть оснащена источником аварийного питания.

Важнейшим элементом системы сигнализации являются пожарные сигнализации, преобразовывающие физические параметры, характеризую-

щие пожар, в электрические сигналы. И подразделяются по способу приведения в действие: ручные и автоматические. По требованию ГОСТа должны применяться оба вида датчиков.

При возникновении возгорания на предприятии ИВО нельзя применять большинство видов средств пожаротушения, так как они содержат воду или иным способом могут повредить дорогостоящую технику или вызвать замыкание электропроводки. Поэтому на предприятиях ИВО предпочтительно использовать углекислотные огнетушители.

#### Выводы к разделу 5

Рассмотрены требования, необходимые для создания комфортных условий работы на ПЭВМ.

Для того чтобы удовлетворить приведенным требованиям по шуму, вибрациям и т.п., рекомендовано проводить закупки оборудования с небольшой вибрацией, пониженным шумом. В качестве конкретных мер можно предложить пользоваться нематричными (лазерными или струйными) принтерами, а систему вентиляции, снабдив звукопоглотителями и, по возможности, виброгасителями, вынести за пределы рабочего помещения.

Разработана программа, выполняющая расчет уровня шума. С помощью нее проверено соответствие уровня шума в вычислительной лаборатории согласно нормируемым показателям.

#### Заключение

В результате проделанной работы был разработан механизм веб-доступа к любой транзакционной РСУБД. Данный механизм позволяет получить доступ к информации из внешних приложений. Приложение спроектировано с помощью шаблона MVC, что поможет с легкостью сопровождать его в дальнейшем. Механизм защиты продуман до мелочей, таких как создание и использование сертификата для сайта. Независимость от СУБД позволит хранить данные прозрачно от самой системы.

Помимо основного модуля была написана и внедрена веб-служба, с помощью которой администратор может мгновенно получать информацию с

помощью клиента J2ME. Данная возможность показывает, как легко можно создавать сложные вещи, не прибегая к большим усилиям.

Переносимость основного модуля между программными платформами делает его более гибким. Поэтому внедрить его в любую операционную систему, поддерживающую платформу Java, не составит большого труда.

○ Все это делает данный проект успешным и готовым к использованию.

Следовательно, предприятие, учреждение или организация может существенно расширить сферу своей деятельности, автоматизировать ее, не прибегая к использованию другой учетной системы. Это намного эффективнее и продуктивнее, позволяет автоматизировать деятельность и, следовательно, снизить расходы на ее поддержание.

Очень важно то, что разработка позволяет осуществлять многопоточную работу с данными, хранящимися в базе данных, и механизм транзакций, реализованный в СУБД, позволяет это сделать с легкостью. Механизм синхронизации потоков, реализованный в Java, и используемый в данной работе, позволяет «общаться» с базой данных множеству потоков в одном процессе сервера. Все это необходимо при реализации полноценных приложений веб, таких как ИМ.

○ Разработанный проект конечно не идеален и автор в дальнейшем планирует реализовать следующие возможности:

- добавить поддержку оплаты с помощью Web Money и Credit Card;
- добавить форум по обсуждению товаров и работы ИМ;
- получить сертификат от VeriSign;
- создать удобную систему рейтингов товаров, тем самым, обеспечив еще больший скачок продаж;
- добавить дополнительную систему отчетов;
- разработать улучшенный пользовательский интерфейс при помощи языка JavaScript;
- разработать дополнительные веб-службы для взаимодействия с ИМ клиентам организации;
- организовать взаимодействие с платформой «1С Торговля и склад» на основе языка XML. Тем самым удастся синхронизировать товары на локальной машине и удаленной.

○ Был обоснован экономический эффект от внедрения разработки.

Итак, автор достиг цели – создал полноценное приложение, реализующее механизм работы ИМ.

Confidential

#### Список использованной литературы

1. Аткинсон Л. MySQL. Библиотека профессионала.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 624 с.
2. Allimant F. The Java Tutorials J2SE 6. Sun Microsystems, Inc. 2006. – 1398 с.
3. Mukhar K. Beginning Java EE 5. Apress, Inc. 2006. – 673 с.
4. Коннолли Т. Базы данных. Проектирование. 3-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1440 с.
5. Minter D. Beginning Hibernate. Apress, Inc. 2006. – 359 с.
6. Doray A. Beginning Apache Struts. Apress, Inc. 2006. – 513 с.

7. Chopra V. Beginning JavaServer Pages. Wiley Publishing, Inc. 2005. – 1297 с.
8. Brittain J. Tomcat: The Definitive Guide. O'Reilly & Associates, Inc. 2003. – 336 с.
9. Topley K. Java Web Services in Nutshell. O'Reilly & Associates, Inc. 2003. – 996 с.
10. Архипов П.С. Безопасность жизнедеятельности. М.: МФТИ, 2000. – 250с.
11. Васильчук М.В. Основы охраны труда. Киев: Просвита, 1997. – 420с.
12. Бутенко Е.С. Основы эргономики. М.: МФТИ, 1999. – 218с.
13. Демин С.В. Борьба с шумом на производстве. М.: МФТИ, 1999. – 240с.
14. Макгрегор Д. Тестирование объектно-ориентированного программного обеспечения. Практическое пособие: Пер. с англ. – К.: ООО «ТИД ДС», 2002. – 432 с.
15. Орлов С.А. Технологии разработки программного обеспечения. Учебное пособие. 2-е изд. СПб.: Питер, 2003. – 480 с.