

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра компьютерных технологий

«Допущен к защите»  
Заведующий кафедрой \_\_\_\_\_

(Ф.И.О., ученая степень, звание)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка программного обеспечения для декартаментов по управлению персоналом компании «SULPAK»

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Алиев М.А. БВТУ - 10  
(Фамилия и инициалы) группа

Научный руководитель ст. преподаватель Тергеусизова А.С.  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Шарипова А.А. к.э.н., проф  
(Фамилия и инициалы, ученая степень, звание)  
« 11 » июня 2014 г.  
(подпись)

по безопасности жизнедеятельности:

Шайдарбекова Н.К., к.х.н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
« 10 » юн 2014 г.  
(подпись)

по применению вычислительной техники:

Тергеусизова А.С., старший преподаватель  
(Фамилия и инициалы, ученая степень, звание)  
« 12 » июня 2014 г.  
(подпись)

Нормоконтролер: Тергеусизова А.С., старший преподаватель  
(Фамилия и инициалы, ученая степень, звание)  
« 12 » июня 2014 г.  
(подпись)

Рецензент: к.ф.-м.н. доц. Казактыу Серкелдибеков  
(Фамилия и инициалы, ученая степень, звание)  
« 09 » юн 2014 г.  
(подпись)

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Экономического образования и переподготовки специалистов  
Специальность Вычислительная техника и программные обеспечения  
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Лелия Мария Александровна  
(фамилия, имя, отчество)

Тема проекта Разработка программного обеспечения для департамента по управлению персоналом магазина "СМРАК"

утверждена приказом ректора № 115 от «24» сентября 2013 г.

Срок сдачи законченной работы «14» июня 2014 г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

Разработка программного обеспечения для упрощения работ департамента по управлению персоналом, более удобного и легкого в использовании, специально для магазина "СМРАК"

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

1. Анализ существующих и автоматизированных систем управления персоналом.
2. Разработка модели данных, выбор программных и технических средств
3. Анализ построения базы данных
4. Реализация программного обеспечения

Перечень графического материала (с точным указанием обязательных чертежей)

1. Иерархическая модель базоданных
2. Дерево функций программного комплекса
3. Интерфейсы справочников
4. Формы программы
5. Формирование отчетов.

Рекомендуемая основная литература

1. Фирсов В.В., Шеняков М.В. Delphi. Руководство разработчика баз. данных. М. 2002.
2. Архангельский А.Э. Разработка прикладных программ для Windows в Delphi. М. 2002.
3. Брайрин В.М. Программное обеспечение персональных ЭВМ. М. Наука 2002.
4. Структурированный язык запросов (SQL) СРБ: ИТМО 2006.
5. Голованов Н.Б. Введение в программирование в сетях Novell NetWare. Питер. 2004.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Безопас. инфор. Экон. часть	Шайгарбекова М.К.	2.06 - 9.06.14	Шайгарбекова
Нормоконтроль	Терехунова В.С.	12.06.2014	Терехунова
Конт. техн.	Терехунова В.С.	12.06.2014	Терехунова
Научный руковод.	Терехунова В.С.	12.06.2014	Терехунова



## **Аннотация**

Дипломный проект содержит пояснительную записку в 110 страниц. Целью данной работы является разработка автоматизированной информационной системы отдела кадров. Дается анализ современных подходов к автоматизированию систем управления персоналом. Описание основных тенденций развития рынка автоматизированной системы управления персоналом. Также представлены требования и функций, которые возлагает система. Произвелся анализ информационных потоков, и сравнительный анализ инструментов для разработки приложений, Также разработалась структура базы данных и создавалась инфологическая модель базы данных. Реализованы были алгоритмы и системный интерфейс. Описан набор, созданного с помощью запросов в систему, и опубликован свой доклад. Также учтена эффективность установленного программного обеспечения.

## **The Summary**

The degree project contains the explanatory note on 110 pages. The purpose of this work is the development of automated information system of human resources department. The analysis of modern approaches to automated control systems for the personnel is given. The description of main tendencies of development of the market of the automated personnel management systems is resulted. Also presented are the system requirements and functions to be performed. Analysis of information flows, and comparative analysis software tools for application development, carried out selection of the necessary technical means to operate the system. Also developed a database structure and created infological and datalogical database model.

describes the implemented algorithms, the system interface. Description of a set created by querying the system and issued its report. Also consider the effectiveness of established software.

## **Түсініктеме**

Дипломдық жобасы 110 баспа беттік алты тарауда көрсетілген: Автоматтау объектісінің жұмыс істеу және тұрғызу принциптері, аналог және прототиптердің талдауы жасалған, ААЖ құрастыру қажеттілігі анықталған, құрастыратын ААЖ талаптары сипатталған. Құрастырылатын жүйенің негізгі талаптары, функциональды, жүйенің сенімділік талаптары, жобалау шектеулері, сонымен қатар рұқсат етілмеген кіруден қорғау талаптары анықталған. Ақпараттық-логикалық құрылым құрастыруға арналған - осы тарауда дипломдық жобаны жүзеге асыру үшін техникалық құралдарды таңдауы, программалық және жүйелік қамтамасыз етуі қарастырылған. Сонымен қатар ААЖ мәліметтер қоймасының құрылымын жүзеге асыру көрсетілген.

## Содержание

Введение.....	14
1 Анализ современных подходов к автоматизированным системам управления персоналом.....	17
1.1 Назначение и классификация автоматизированных систем управления персоналом.....	17
1.2 Перспективы развития мирового рынка автоматизированных систем управления персоналом.....	18
1.3 Основные тенденции развития рынка автоматизированных систем управления персоналом.....	18
1.4 Анализ зарубежных систем управления персоналом.....	19
1.5 Функции отдела кадров.....	20
2 Разработка модели данных, выбор программных и технических средств.....	23
2.1 Требования к организации базы данных.....	23
2.1.1 Установление многосторонних связей.....	23
2.1.2 Производительность.....	23
2.1.3 Минимальные затраты.....	23
2.1.4 Минимальная избыточность.....	24
2.1.5 Возможности поиска.....	24
2.1.6 Целостность.....	24
2.1.7 Безопасность и секретность.....	25
2.1.8 Связь с прошлым.....	25
2.1.9 Связь с будущим.....	25
2.1.10 Простота использования.....	26
2.2 Основы построения банков данных.....	26
2.3 Архитектуры баз данных.....	29
2.4 Проблемы проектирования БД.....	34
2.5 Выбор программных средств реализации системы.....	35
2.6 Выбор технических средств реализации системы.....	45
2.7 Анализ информационных потоков.....	46
2.8 Разработка структуры базы данных.....	50
3 Реализация программного обеспечения.....	59
3.1 Пользовательское меню.....	59
3.2 Формы как средство добавления, удаления, просмотра, изменений параметров базы данных.....	60
3.3 Формирование отчетов.....	65
4 Экономическая часть.....	70
5 Безопасность жизнедеятельности.....	77
5.1 Анализ опасных и вредных факторов, воздействующих на программиста.....	77
5.1.1 Мероприятия по снижению опасных и вредных факторов.....	78
5.2 Микроклимат и вентиляция.....	80

5.2.1 Освещение.....	81
5.2.2 Защита от излучений и электромагнитных полей .....	82
5.2.3 Электробезопасность .....	83
5.3 Меры пожарной безопасности.....	78
Заключение .....	81
Список использованной литературы.....	82
Приложение А .....	84

## Введение

Кадровые службы, являясь одной из важнейших структур управления, обеспечивают укомплектование предприятий, учреждений, организаций, фирм и т.п. рабочими, специалистами, служащими, организацию подготовки, переподготовки повышения квалификации работников, аттестацию работников и руководителей, реализует меры, направленные на укрепление трудовой дисциплины, сокращение текучести кадров, осуществляет подготовку распорядительных документов по личному составу, необходимой статистической отчетности по кадрам, ведут в установленном порядке учет работников, в том числе военнообязанных, кадровое делопроизводство, оформляет документы для назначения работникам пенсий, решают иные вопросы, входящие в их компетенцию.

Структура кадровой службы, ее численность определяется организационно-штатной структурой предприятия, учреждения, организации, фирмы. При этом за основу расчета необходимого ее работников принимается: численность работающих; конкретные условия; характерные особенности, связанные с типом, масштабами и разновидностями отдельных производств; технико-экономические и социальные характеристики предприятия, учреждения, организации, фирм; другие показатели.

В отдельных случаях в составе кадровой службы могут функционировать функциональные группы: оформление приема на работу, переводов и увольнений; учета и отчетности; военного учета; подготовки и переподготовки и повышения квалификации работников, оформление отпусков.

Актуальность дипломного проекта. Вопросы продуманного управления персоналом должны занимать одно из ключевых мест в общей стратегии развития предприятия. Важная роль в оптимизации управления сотрудниками любого предприятия принадлежит автоматизированным системам управления персоналом (так называемым Human Resource systems). В своей деятельности компании вынуждены нанимать работников, как на временной, так и постоянной основе, вести документацию на каждого из них, поддерживая ее в виде, соответствующем как внутренним, так и внешним (кодекса законов о труде) требованиями. Любая автоматизированная система управления персоналом предоставляет всем заинтересованным лицам (руководству, менеджерам) мгновенный доступ ко всей необходимой информации.

Раньше использование автоматизированных систем управления считалось привилегией крупных предприятий. Поэтому масштаб и цены этих систем были соответствующими, а малым и средним компаниям большинство продуктов были не по карману. И хотя ныне практически все поставщики предлагают системы малого и среднего класса, рынок все еще далек от насыщения. Поэтому перед руководством предприятий малого и среднего бизнеса встает выбор: приобретать готовую автоматизированную систему или



использовать офисные средства. Следует отметить, что при приобретении готового программного продукта в любом случае потребуются серьезная настройка программы под конкретные требования, а также обязательное дальнейшее программное сопровождение, и обучение персонала, что выразится в ощутимых финансовых и временных затратах, а офисные средства способны в полной мере решить поставленные задачи и приемлемы по цене. Таким образом, тема данного дипломного проекта является актуальной с практической точки зрения.

Целью дипломного проекта является разработка программного обеспечения системы управления персоналом для магазина «SULPAK»

Работа отдела кадров предприятия связана с накоплением большого количества информации о личных данных сотрудников. Традиционно информация хранится на бумажных носителях. При этом трудно осуществить быстрый отбор нужных данных при приеме на работу, увольнении, составлении отчетов. Основанием разработки послужил ряд причин, возникающих в силу влияния человеческого фактора, которые отрицательно сказываются на качестве и интенсивности работы отдела кадров:

- возможность случайной ошибки в ручном заполнении бланков;
- невозможность быстро справиться с большим объемом информации;
- сложность поиска необходимой информации.

Задачи разрабатываемой системы. Целью разработки программного обеспечения системы управления персоналом для магазина «SULPAK» является максимально возможное устранение случайных ошибок, перечисленных недостатков существующей системы работы отдела кадров и привнесение в нее факторов, положительно влияющих на качество и сроки выполнения реализуемых в ней функций:

- уменьшение времени выполнения каждой функции;
- исключение многократного ввода одних и тех же данных в учетную систему;
- автоматическое создание документации и отчетов;
- простой и быстрый поиск;
- автоматическое проставление дат и табельных номеров.

Практическая ценность дипломного проекта. Современные автоматизированные системы управления персоналом предназначены для оптимизации работы, в первую очередь, руководства и персонала кадровых служб предприятий и играют большую роль в повышении производительности их труда. В частности, менеджеры по персоналу при помощи таких систем избегают от выполнения рутинных операций при работе с кадрами, подготовке и учете приказов (существуют оценки, что только на работу с документацией по персоналу кадровики тратят до 60% своего рабочего времени). Автоматизированное хранение и обработка полной кадровой информации также позволяет эффективно осуществлять подбор и перемещение сотрудников.

Научная новизна работы заключается в создании среды, осуществляющей информационно-вычислительную поддержку деятельности отдела кадров предприятия.

Апробация на предприятии. Программный продукт прошел апробацию на предприятии. В ходе апробации и внедрения был выполнен полный объем запланированных работ.

## **1 Анализ современных подходов к автоматизированным системам управления персоналом**

### **Назначение и классификация автоматизированных систем управления персоналом**

Современные автоматизированные системы управления персоналом предназначены для оптимизации работы, в первую очередь, руководства и персонала кадровых служб предприятий и играют большую роль в повышении производительности их труда. В частности, менеджеры по персоналу при помощи таких систем избавляются от выполнения рутинных операций при работе с кадрами, подготовке и учете приказов. Автоматизированное хранение и обработка полной кадровой информации также позволяет эффективно осуществлять подбор и перемещение сотрудников.

Существующие в настоящее время на рынке автоматизированные системы управления персоналом по их функциональной направленности можно разделить на следующие основные группы [1]:

- многофункциональные экспертные системы, позволяющие проводить профориентацию, отбор, аттестацию сотрудников предприятия;
- экспертные системы для группового анализа персонала, выявления тенденций развития подразделений и организации в целом;
- программы расчета зарплаты;
- комплексные системы управления персоналом, позволяющие формировать и вести штатное расписание, хранить полную информацию о сотрудниках, отражать движение кадров внутри фирмы, рассчитывать зарплату.

В частности, с помощью экспертных систем сопоставляются личностные, профессиональные и психофизиологические качества претендента на занятие вакансии с аналогичными параметрами лучших специалистов предприятия. Такие программы позволяют достаточно эффективно (с формальной точки зрения) отбирать перспективных специалистов. Подобные системы достаточно дороги, их имеет смысл использовать только для крупных предприятий при наличии на них большого количества вакансий. Эти системы не позволяют автоматизировать рутинные повседневные операции персонала кадровых служб.

Комплексные системы управления персоналом используются для автоматизации кадровой работы на любом предприятии. В первую очередь, такие системы необходимы руководству для получения оперативной информации по любому вопросу, связанному со структурой предприятия, штатным расписанием, вакансиями и информацией о сотрудниках. Оперативно принять правильное решение сможет лишь руководитель, способный быстро оценить сложившуюся ситуацию на основе анализа актуальной информации о состоянии дел на предприятии. Поэтому немаловажным фактором условий использования HR-систем является также возможность интеграции системы

кадрового учета с системами бухгалтерского учета и управления предприятием.

## **1.2 Перспективы развития мирового рынка автоматизированных систем управления персоналом**

В современной высококонкурентной бизнес-среде резко возросла потребность предприятий в высококвалифицированных сотрудниках. Усилился спрос руководства и кадровых служб предприятий на более специализированные данные о персонале, которые помогают лучше спланировать будущие потребности предприятий в персонале и реализовывать свои планы путем продуманного рекрутинга, назначения и сохранения в штате талантливых и высококвалифицированных сотрудников. Именно поэтому сегмент приложений управления персоналом является сейчас одним из наиболее быстро развивающихся на мировом рынке бизнес-приложений. По прогнозу IDC (отчет «2001: A Workforce Management Odyssey, Workforce Management Applications Market Forecast and Analysis, 2000-2004») объем мирового рынка приложений управления персоналом (workforce management application) должен увеличиться с \$1,5 млрд в 2000 г. до более чем \$4 млрд в 2004 г. (с ежегодными темпами роста в 29%). Для сравнения, ежегодные темпы роста всего рынка бизнес-приложений составят лишь 15%. Доля приложений управления персоналом на мировом рынке бизнес-приложений увеличится с 1,8% в 1999 г. до 3,4% в 2003 г. (отчет «Workforce Management Applications in 2000: Overview and Vendor Strategies»).[2] По мнению IDC, в состав приложений управления персоналом входят коллаборативные, транзакционные и аналитические программные средства, которые должны помочь кадровым службам в найме, назначении и сохранении высококвалифицированных сотрудников.

Разработчики приложений управления персоналом в настоящее время стремятся к быстрой адаптации у заказчиков. Скорость адаптации приложений теперь более, чем раньше, зависит от того, какие ресурсы их разработчики инвестируют в обучение пользователей и передачу им соответствующих знаний и навыков до и после продажи ПО. Прежде всего, производителям таких приложений необходимо убедить потребителей в выгоде их использования.

## **1.3 Основные тенденции развития рынка автоматизированных систем управления персоналом**

**Консолидация.** В ближайшем будущем рынок WFM-приложений продолжит свою консолидацию (по мере того как их разработчики обеспечат в них достаточную функциональность).

**Увеличение числа разработчиков.** Постоянный спрос потребителей на

предложения ПО в сферах отслеживания кандидатов на вакансии, планирования и сохранения рабочего времени сотрудников, а также управления компенсациями для сотрудников будет способствовать появлению как новых поставщиков пакетов приложений управления персоналом, так и новых разработчиков однофункциональных приложений.

Расширение функциональности WFM-приложений. Новые функциональные возможности для WFM-приложений заключаются в автоматической синхронизации условий контракта сотрудников с фактическими результатами их работы, расширении WFM-функциональности на Интернет для найма персонала, его назначения на соответствующие должности и сохранения, а также в добавлении возможностей групповой работы, более совершенной работы с контентом и аналитики к общей функциональности WFM-приложений.

Существуют также оценки аналитиков, что в новых WFM-приложениях будут реализованы возможности управления расходами на персонал, учета рабочего времени персонала и управления его деятельностью. Позднее будут также адаптированы приложения по управлению навыками персонала, управления наймом группы сотрудников и др.[2]

#### **1.4 Анализ зарубежных систем управления персоналом**

Подавляющее большинство комплексных корпоративных информационных систем корпоративных информационных систем (КИС) зарубежной разработки построены по модульному принципу и имеют в своем составе модуль управления персоналом, реализующий автоматизированное управление кадрами (нередко управление кадрами объединено также с расчетом зарплаты). Можно назвать такие известные в мире системы, имеющие в своем составе Human Resources (HR) модули, как SAP R/3, Baan, Oracle Applications и др. Существуют и автономные программные пакеты управления персоналом, одним из примеров которых является ПО Renaissance CS Human Resources.

В общем случае принято считать, что ощутимый эффект от внедрения HR-систем заметен, когда численность персонала предприятия превышает 1000 человек. Западными разработчиками КИС утверждается, что внедрение HR-модулей позволит предприятию получить организационные, экономические и социальные эффекты (следует отметить, что эти эффекты присущи всем современным системам управления персоналом, в том числе и наиболее продвинутым отечественным HR-системам). Организационные эффекты заключаются в следующем :[3]

- сокращении времени принятия решений на всех уровнях управления предприятием;
- повышении качества кадровых решений;
- оперативности подготовки отчетности для органов государственного

управления в соответствии с российскими законодательными и нормативными требованиями.

В свою очередь, влияние экономических эффектов от внедрения HR-модулей позволяет :

- снизить затраты на управление персоналом;
- повысить производительность труда персонала;
- оптимально использовать профессиональные качества конкретного сотрудника предприятия.

Социальный эффект от внедрения HR-модулей состоит в следующем:

- персональном учете пенсионных накоплений сотрудников предприятия;
- ведении полной индивидуальной трудовой истории персонала предприятия;
- подготовке руководящего резерва и продвижении по службе наиболее перспективных сотрудников предприятия.

Безусловно, западные разработчики вложили в создание данных HR-модулей огромные человеческие и материальные ресурсы, а также накопили опыт их внедрения на сотнях и тысячах зарубежных предприятий. В то же время, существуют определенные ограничения для распространения этих модулей на российских предприятиях (не принимая во внимание такие факторы, когда собственниками предприятия являются западные инвесторы, а внедряемая система принята в качестве корпоративного стандарта). Основными из этих ограничений являются:

- высокая цена внедрения и поддержки (например, стоимость расчета зарплаты для одного сотрудника предприятия может достигать нескольких тысяч долларов);
- функциональная избыточность (например, данные HR-модули включают такие невостребованные пока на российских предприятиях функции, как индивидуальное планирование служебного роста сотрудника предприятия; отображение долгосрочных тенденций в потенциале сотрудника; учет снижения квалификации сотрудника (в зависимости от продолжительности его перерыва в работе по специальности), влияющий на размер его зарплаты; планирование мероприятий по повышению квалификации персонала с заблаговременным резервированием помещений и др.);
- недостаточная скорость адаптации к динамически изменяющемуся российскому законодательству (особенно, налоговому);
- в некоторых случаях, недостаточная полнота локализации интерфейсов.[3]

## **1.5 Функции отдела кадров**

Отдел кадров магазина «SULPAK» является структурным подразделением предприятия и имеет в своем составе одного сотрудника.

Сотрудник отдела кадров обеспечивает предприятие работниками требуемых специальностей и квалификаций, а также следит за соблюдением действующего законодательства в работе с кадрами и за документированием кадровой деятельности предприятия. Сотрудник назначается на должность и освобождается от должности директором предприятия и находится у него в непосредственном подчинении. На эту должность назначается лицо с высшим образованием и имеющее стаж работы в данной сфере не менее двух лет. Сотрудник в своей работе руководствуется действующими законодательными актами Республики Казахстан о труде, нормативными актами Правительства РК, организационными и распорядительными документами предприятия, указаниями директора, правилами внутреннего трудового распорядка и должностной инструкцией. Инспектор по кадрам должен знать трудовое законодательство, положения, инструкции и другие руководящие материалы и нормативные документы по учету личного состава, оформлению пенсий, ведению и хранению трудовых книжек работников, унифицированные формы документации, структуру предприятия и его подразделений, правила внутреннего трудового распорядка, правила и нормы охраны труда.

Сотрудник отдела кадров обязан:

Вести учет личного состава предприятия, его подразделений и оформлять кадровые документы;

Оформлять прием, перевод и увольнение работников в соответствии с трудовым законодательством, положениями, инструкциями и приказами директора;

Заполнять, учитывать и хранить трудовые книжки, производить подсчет трудового стажа;

Выдавать справки о настоящей и прошлой трудовой деятельности работников;

Вести учет предоставления отпусков работникам, осуществлять контроль над составлением и соблюдением графиков очередных отпусков;

Осуществлять контроль над своевременным исполнением приказов, поручений и распоряжений по кадрам директора предприятия;

Вести учет приказов и других документов по правовым вопросам, заполнять учетные документы (журналы, картотека).

Сотрудник имеет право:

Требовать у работников предприятия представления документов, необходимых для осуществления своих функций;

Требовать у руководителей подразделений своевременного предоставления сведений и документов для составления документов по личному составу, графиков отпусков, табельного учета, ведения трудовых книжек и т.д.;

Визировать приказы директора по личному составу;

Подписывать личные справки работникам предприятия;

Вести переписку по вопросам кадровой работы с другими организациями.

К основным кадровым функциям можно отнести [4]:  
прием на работу новых сотрудников;  
увольнение сотрудников;  
кадровое перемещение (изменение должности, изменение графика работы, перевод на другой объект);  
оформление отпусков;  
оформление больничных листов;  
ведение штатного расписания;  
отчетность (перед бухгалтерией, перед статистическими органами, перед руководством).



## **2 Разработка модели данных, выбор программных и технических средств**

### **2.1 Требования к организации базы данных**

#### **2.1.1 Установление многосторонних связей**

Различным программистам требуются различные логические файлы. Эти файлы получают из одной и той же совокупности данных. Между элементами запоминаемых данных могут существовать различные связи. Некоторые базы данных будут содержать сложные переплетения взаимосвязей. Метод организации данных должен быть таким, чтобы обеспечивалась возможность удобного представления этих взаимосвязей и быстрого согласования вносимых в них изменений. Система управления базами данных должна обеспечивать возможность получения требуемых логических файлов из имеющихся данных и существующих между ними связей. Необходимо, чтобы существовало хотя бы небольшое сходство между представлением логического файла в прикладной программе и способом физического хранения данных.

#### **2.1.2 Производительность**

Базы данных, специально разработанные для использования их оператором терминала, обеспечивают время ответа, удовлетворительное для диалога человек – терминал. Кроме того, система баз данных должна обеспечивать соответствующую пропускную способность. В системах, рассчитанных на небольшой поток запросов, пропускная способность накладывает незначительные ограничения на структуру базы данных. В системах с большим потоком запросов, например в системах резервирования авиабилетов, пропускная способность оказывает решающее влияние на выбор организации физического хранения данных.

В системах, предназначенных только для пакетной обработки, время ответа не так важно и метод физической организации может выбираться из условий обеспечения эффективной пакетной обработки.

#### **2.1.3 Минимальные затраты**

Для уменьшения затрат на создание и эксплуатацию базы данных выбираются такие методы организации, которые минимизируют требования к внешней памяти, а так же при ограниченных ресурсах рабочего места оператора. При использовании этих методов физическое представление данных в памяти может сильно отличаться от того представления, которое использует

прикладной программист. Преобразование одного представления в другое осуществляет программное обеспечение либо, если возможно, аппаратные или микропрограммные средства, что приводит к экономии времени на предприятии. В таких случаях приходится выбирать между затратами на алгоритм преобразования и экономией памяти, и машинного времени.

#### **2.1.4 Минимальная избыточность**

В системах обработки, существовавших до использования систем управления базами данных, информационные фонды обладали очень высоким уровнем избыточности. Большинство ленточных библиотек содержало большое количество избыточных данных. Даже при использовании баз данных по мере возрастания информации, объединяемой в интегрированные базы данных, потенциальная возможность появления избыточных данных постепенно увеличивается. Избыточные данные дороги в том смысле, что они занимают больше памяти, чем это необходимо, и требуют более одной операции обновления. Целью организации базы данных должно быть уничтожение избыточных данных там, где это выгодно, и контроль за теми противоречиями, которые вызываются наличием избыточных данных.

#### **2.1.5 Возможности поиска**

Пользователь базы данных может обращаться к ней с самыми различными вопросами по поводу хранимых данных. В большинстве современных коммерческих приложений типы запросов predetermined, и физическая организация данных разрабатывается для их обработки с требуемой скоростью. Возросшие требования к системам заключаются в обеспечении обработки таких запросов или формирования таких ответов, которые заранее не запланированы.

#### **2.1.6 Целостность**

Если база данных содержит данные, используемые многими пользователями, очень важно, чтобы элементы данных и связи между ними не разрушались. Необходимо учитывать возможность возникновения ошибок и различного рода случайных сбоев. Хранение данных, их обновление, процедуры включения данных должны быть такими, чтобы система в случае возникновения сбоев могла восстанавливать данные без потерь. Необходимо, чтобы вычислительная система гарантировала целостность хранимых в ней данных.

### **2.1.7 Безопасность и секретность**

Данные в системах баз данных должны храниться в тайне и сохранности. Запоминаемая информация иногда очень важна для использующего ее учреждения. Она не должна быть утеряна или похищена. Для увеличения жизнестойкости информации в базе данных важно защищать ее от аппаратных или программных сбоев, от катастрофических и криминальных ситуаций, от некомпетентного или злонамеренного использования лицами, которые могут ее неправильно употребить.

Под безопасностью данных понимают защиту данных от случайного или преднамеренного доступа к ним лиц, не имеющих на это право, от неавторизованной модификации данных или их уничтожения.

Секретность определяют как право отдельных лиц или организаций определять, когда, как и какое количество соответствующей информации может быть передано другим лицам или организациям.

### **2.1.8 Связь с прошлым**

Организации, которые в течение какого-то времени эксплуатируют системы обработки данных, затрачивают значительные средства на написание программ, процедур и организацию хранения данных. В том случае, когда фирма начинает использовать на вычислительной установке новое программное обеспечение управления базами данных, очень важно, чтобы при этом она могла работать с уже существующими на этой установке программами, обрабатываемые данные можно было бы соответствующим образом преобразовывать. Такое условие требует наличия программной и информационной совместимости, и ее отсутствие может стать основным сдерживающим фактором при переходе к новым системам управления базами данных. Важно, однако, чтобы проблема связи с прошлым не сдерживала развитие средств управления базами данных.

### **2.1.9 Связь с будущим**

Особенно важной представляется связь с будущим. В будущем данные и среда их хранения изменятся по многим направлениям. Любая коммерческая организация со временем претерпевает изменения. Особенно дорогими эти изменения оказываются для пользователей системами обработки данных. Огромные затраты, которые требуются для реализации самых простых изменений, сильно тормозят развитие этих систем. Эти затраты расходуются на преобразование данных, перезапись и отладку прикладных программ, явившихся результатом внесения изменений. Со временем число прикладных

программ в организации растет, и поэтому перспектива перезаписи всех этих программ кажется нереальной. Одна из самых важных задач при разработке баз данных – запланировать базу данных таким образом, чтобы изменения ее можно было выполнять без модификации прикладных программ.

### **2.1.10 Простота использования**

Средства, которые используются для представления общего логического описания данных, должны быть простыми и изящными.

Интерфейс программного обеспечения должен быть ориентирован на конечного пользователя и учитывать возможность того, что пользователь не имеет необходимой базы знаний по теории баз данных.

## **2.2 Основы построения банков данных**

Вычислительная техника с каждым годом все шире применяется в различных сферах человеческой деятельности. Резкий рост объемов перерабатываемой информации и накопленный опыт пользования электронно-вычислительной техникой в различных областях человеческой деятельности приводят к необходимости пересматривать такую, традиционную область обработки информации, как управление данными.

При создании баз данных (БД) необходимо уделить особое внимание тому, чтобы данные можно было широко использовать в различного рода приложениях и чтобы способы использования данных можно было легко и быстро изменять. До появления БД было чрезвычайно трудно изменить способ организации используемых данных.

Для обеспечения гибкости использования данных необходимо учитывать два аспекта разработки БД:

– во-первых, данные должны быть независимы от программ для того, чтобы данные можно было добавлять или перестраивать без изменения программ;

– во-вторых, должна быть обеспечена возможность запрашивать и отыскивать нужную информацию в БД без трудоемкого написания программ на обычном языке программирования. Таким образом, проектирование БД должно основываться на вполне определенной системе положений – четко сформулированной концепции.

Продолжающийся значительный рост использования ЭВМ в различных областях промышленности, в управлении и научных исследованиях привел к автоматизации обработки огромнейшего количества данных. В конце 50-х начале 60-х годов XX века многие организации начали накапливать и хранить данные в виде файлов, доступных ЭВМ. С течением времени организации постепенно осознавали необходимость централизации управления данными и

приложениями.

База данных может быть определена как совокупность предназначенных для машинной обработки данных, которая служит для удовлетворения нужд многих пользователей в рамках одной или нескольких организаций. Основным моментом является то, что база данных предназначена для использования всеми членами организации, которым необходима информация, содержащаяся в базе данных. Информация хранится в базе данных, которая может включать много различных типов логических записей. База данных ориентирована на интегрированные требования, а не на одну программу, как было с частными файлами данных.

Однако наличие только базы данных само по себе не разрешает полностью проблем организации в области обработки данных и принятия решений. Управление базой данных, являющейся достоянием многих пользователей внутри организации, должно осуществляться с пользой для всей организации и с точки зрения организации в целом, а не отдельных пользователей. Без централизованного управления базой данных ее полезность со временем снижается.

Для решения проблемы регулирования и управления базами данных были развиты две концепции. Во-первых, программное обеспечение развивалось в направлении, обеспечивающем поддержание общего интерфейса между всеми пользователями и интегрированной базой данных. Пользователи не могут хранить информацию независимым образом, они должны использовать и обновлять информацию в соответствии с требованиями организации. Обеспечение, известное как система управления базами данных (СУБД), позволяет осуществить контроль данных с использованием ЭВМ, СУБД – это специальный пакет программ, посредством которого реализуется централизованное управление базой данных и обеспечивается доступ к данным.

В каждой СУБД прежде всего есть трансляторы или интерпретаторы с языка описания данных (ЯОД) и с языка манипулирования данными (ЯМД), единые для всей базы данных (БД).

Описание структуры данного некоторого типа на формализованном языке называют схемой этого данного. Язык описания данных (ЯОД) – это язык высокого уровня, предназначенный для задания схемы базы данных. С его помощью описываются типы данных, подлежащих хранению в базе или выборке из нее, их структура и связи между собой. Исходные тексты, написанные на этом языке, после трансляции отображаются в управляющие таблицы: адресных констант, констант и другую информацию, необходимую для работы с данными программ СУБД. В соответствии с полученным описанием СУБД может найти в базе требуемые данные, правильно преобразовать их и переработать, например, в прикладную программу, которой они потребовались. При записи данных в базу СУБД определяет место в памяти ЭВМ, куда их требуется поместить, преобразует к заданному виду, устанавливает необходимые связи.

ЯМД представлен системой команд манипулирования данными. В нем

могут быть, например, следующие команды:

- произвести выборку из базы данных конкретного данного, значение которого удовлетворяет заданным условиям;
- произвести выборку из БД всех данных определенного типа, значения которых удовлетворяют заданным условиям и т.д.

Системы управления базой данных подразделяют на две группы в зависимости от способа реализации ЯМД:

- СУБД с включающим языком;
- СУБД с базовым языком.

В СУБД с базовым языком разрабатывается собственный алгоритмический язык, позволяющий кроме манипулирования данными выполнять арифметические операции, операции ввода – вывода на терминалы и так далее.

Во многих СУБД имеются специальные средства обеспечения защиты данных от некомпетентного их использования и сбоев технических средств, средства контроля достоверности данных, средства автоматического накопления статистики использования тех или иных данных различными категориями пользователей.

Другой концепцией является концепция администратора базы данных (АБД). Администратор базы данных – это лицо (или группа лиц), реализующие управление базой данных. Он должен уметь поддерживать взаимосвязи как с руководством высшего уровня, так и с пользователем, обрабатывающим данные, а также руководить штатом технических специалистов.

Этот штат должен включать лиц, имеющих опыт работы в таких областях, как программное обеспечение СУБД, операционные системы, техническое обеспечение ЭВМ, прикладное программирование, системное программирование.

Основная функция АБД – обеспечение структур данных и взаимосвязей между ними, эффективным методом обслуживания коллектива пользователей.

Функции АБД следующие:

- решать вопросы организации данных об объектах программного обеспечения и установление связей между ними с целью объединения информации о различных объектах, согласовывать запросы пользователей;
- координировать все действия по проектированию, реализации и ведению БД;
- учитывать перспективные и текущие требования пользователей;
- решать вопросы, связанные с расширением БД в связи с изменением границ ПО;
- разрабатывать и реализовывать меры по обеспечению защиты данных от некомпетентного их использования, от сбоев технических средств, обеспечение секретности определенной части данных и разграничение доступа к ним;
- контролировать избыточность и противоречивость данных, их достоверность;

– координировать работу технического обеспечения системы, системных программистов и прикладных программистов.

Место АБД было определено тогда, когда организации осознали необходимость централизованного управления ресурсами данных, обработки данных и другие аспекты, связанные с базой данных. Группы пользователей и отдельные пользователи должны обслуживаться всеми средствами, исходя из целей и возможностей организации в целом. АБД является ответственным за анализ потребностей пользователей, проектирование БД, ее внедрение, обновление, реорганизацию, консультацию и обучение пользователей.

### 2.3 Архитектуры баз данных

Для рассмотрения способов организации баз данных нужно определить несколько понятий.[5]

Ядро БД отвечает за управление данными во внешней памяти, управление буферами оперативной памяти, управление транзакциями и журнализацию. Соответственно, можно выделить такие компоненты ядра (по крайней мере, логически, хотя в некоторых системах эти компоненты выделяются явно), как менеджер данных, менеджер буферов, менеджер транзакций. Ядро БД обладает собственным интерфейсом, не доступным пользователям напрямую и используемым в программах. Ядро БД является основной резидентной частью СУБД. При использовании архитектуры "клиент-сервер" ядро является основной составляющей серверной части системы.

Основной функцией компилятора языка БД является компиляция операторов языка БД в некоторую выполняемую программу.

В отдельные утилиты БД обычно выделяют такие процедуры, которые слишком накладно выполнять с использованием языка БД, например, загрузка и выгрузка БД, сбор статистики, глобальная проверка целостности БД и т.д. Утилиты программируются с использованием интерфейса ядра БД, а иногда даже с проникновением внутрь ядра.

Общий состав средств, необходимых для работы готового приложения с БД, показан на рисунке 2.1. Согласно этой общей схеме, мы имеем цепочку

Приложение → Ядро БД → Базы данных. В структуре приложения имеется цепочка Невизуальные компоненты → Визуальные компоненты. Невизуальные компоненты предоставляют программисту некоторые функции по управлению ядром базы данных, а также самими данными. С помощью Визуальных компонентов данные отображаются на экране (таблицы, списки, выпадающие списки, графики и другие). Местоположение ядра БД и самих баз данных в этой цепочке не отражены.

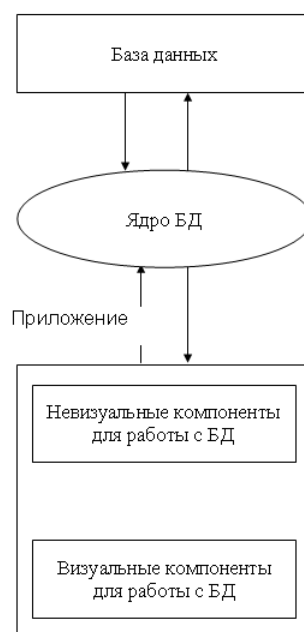


Рисунок 2.1 - Общий состав средств для работы готового приложения с БД

Местоположение Ядра БД и баз данных зависит от используемой архитектуры. Имеется три разновидности архитектур баз данных:[5]

локальные базы данных и архитектура "файл-сервер";

архитектура "клиент-сервер";

многозвенная (трехзвенная N-tier или multi-tier) архитектура.

Использование той или иной архитектуры накладывает сильный отпечаток на общую идеологию работы приложения, на программный код в приложении, на состав компонентов для работы с БД, используемых в приложении (прежде всего это касается невидуальных компонентов).

При работе с локальными базами данных сами БД расположены на том же компьютере, что и приложения, осуществляющие доступ к ним. Работа с БД происходит в однопользовательском режиме. Ядро БД расположено на компьютере пользователя. Приложение ответственно за поддержание целостности БД и за выполнение запросов к БД. Общая схема однопользовательской архитектуры показана на рисунке 2.2.

При работе в архитектуре "файл-сервер" БД и приложение расположены на файловом сервере сети (например, Novell NetWare, Windows 2000, Windows NT 4.0). Возможна многопользовательская работа с одной и той же БД, когда каждый пользователь со своего компьютера запускает приложение, расположенное на сетевом сервере.



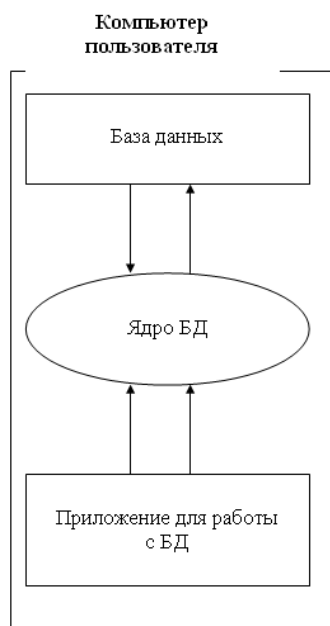


Рисунок. 2.2 - Однопользовательская архитектура при работе с локальными БД

Тогда на компьютере пользователя запускается копия приложения. По каждому запросу к БД из приложения, данные из таблиц БД перегоняются на компьютер пользователя, независимо от того, сколько реально нужно данных для выполнения запроса. После этого выполняется запрос.

Каждый пользователь имеет на своем компьютере локальную копию данных, время от времени обновляемых из реальной БД, расположенной на сетевом сервере. При этом изменения, которые каждый пользователь вносит в БД, могут быть до определенного момента неизвестны другим пользователям, что делает актуальной задачу систематического обновления данных на компьютере пользователя из реальной БД. Другой актуальной задачей является блокирование записей, которые изменяются одним из пользователей: это необходимо для того, чтобы в это время другой пользователь не внес изменений в те же данные. В архитектуре "файл-сервер" вся тяжесть выполнения запросов к БД, управления целостностью БД ложится на приложение пользователя. БД на сервере является пассивным источником данных. Общая схема архитектуры "файл-сервер" показана на рисунке 2.1.

Кардинальных различий с точки зрения архитектуры между однопользовательской архитектурой и архитектурой "файл-сервер" нет. И в том и в ином случае в качестве СУБД применяются так называемые "персональные" (или "локальные") СУБД такие как Paradox, dBase и пр. Сама база данных в этом случае представляет собой набор таблиц, индексных файлов, файлов полей комментариев (мемо-полей) и пр., хранящихся в одном каталоге на диске в виде отдельных файлов.[5]

Архитектура "файл-сервер" неэффективна, по крайней мере, в двух отношениях:

- при выполнении запроса к базе данных, расположенной на файловом

сервере, в действительности происходит запрос к локальной копии данных на компьютере пользователя. Поэтому перед выполнением запроса данные в локальной копии обновляются из реальной БД. Данные обновляются в полном объеме. Так, если таблица БД состоит из 1000 записей, а для выполнения запроса (например, выдать сумму премий за октябрь в отделе Y) реально нужно 10 записей, все равно перегоняются все 1000 записей. Таким образом, не нужно иметь слишком много пользователей и запросов от них, чтобы серьезно "забить" сеть, что, конечно же, не может не сказаться на ее быстродействии;

– обеспечение целостности БД производится из приложений. Это потенциальный источник ошибок, нарушающих физическую и логическую целостность БД, поскольку различные приложения могут производить контроль целостности БД по-разному, взаимоисключающими способами, или не проводить такого контроля вовсе. Намного эффективнее управлять БД из единого места и по единым законам, нежели из разных приложений и по потенциально разным законам (все зависит от того, как написано приложение). Поэтому безопасность при работе в архитектуре "файл-сервер" невысока и всегда присутствует элемент неопределенности. Секретность и конфиденциальность при работе с БД в архитектуре "файл-сервер" обеспечить также тяжело – любой, кто имеет доступ в каталог сетевого сервера, где хранится БД, может изменять таблицы БД любым образом, копировать их, заменять и так далее.

Архитектура "клиент-сервер" разделяет функции приложения пользователя (называемого клиентом) и сервера.

Приложение-клиент формирует запрос к серверу, на котором расположена БД, на структурном языке запросов SQL. Удаленный сервер принимает запрос и переадресует его SQL-серверу БД. SQL-сервер – это специальная программа, управляющая удаленной базой данных. SQL-сервер обеспечивает интерпретацию запроса, его выполнение в базе данных, формирование результата выполнения запроса и выдачу его приложению-клиенту. При этом ресурсы клиентского компьютера не участвуют в физическом выполнении запроса; клиентский компьютер лишь отправляет запрос к серверной БД и получает результат, после чего интерпретирует его необходимым образом и представляет пользователю. Так как клиентскому приложению посылается результат выполнения запроса, по сети "путешествуют" только те данные, которые необходимы клиенту. В итоге снижается нагрузка на сеть. Поскольку выполнение запроса происходит там же, где хранятся данные (на сервере), нет необходимости в пересылке больших пакетов данных. Кроме того, SQL-сервер, если это возможно, оптимизирует полученный запрос таким образом, чтобы он был выполнен в минимальное время с наименьшими накладными расходами.

Все это повышает быстродействие системы и снижает время ожидания результата запроса.

При выполнении запросов сервером существенно повышается степень безопасности данных, поскольку правила целостности данных определяются в

базе данных на сервере и являются едиными для всех приложений, использующих эту БД. Таким образом, исключается возможность определения противоречивых правил поддержания целостности. Мощный аппарат транзакций, поддерживаемый SQL-серверами, позволяет исключить одновременное изменение одних и тех же данных различными пользователями и предоставляет возможность откатов к первоначальным значениям при внесении в БД изменений, закончившихся аварийно. Таким образом, функциями приложения-клиента являются:

- посылка к серверу запросов;
- интерпретация результатов запросов, полученных от сервера, и представление их пользователю в требуемой форме;
- реализация интерфейса пользователя.

SQL-сервер – это программа, расположенная на компьютере сетевого сервера. SQL-сервер должен быть загружен на момент принятия запроса от клиента. Функциями сервера БД являются:

- прием запросов от приложений-клиентов, интерпретация запросов, выполнение запросов в БД, отправка результата выполнения запроса приложению-клиенту;
- управление целостностью БД, обеспечение системы безопасности, блокировка неверных действий приложений-клиентов;
- хранение бизнес-правил, часто используемых запросов в уже интерпретированном виде;
- обеспечение одновременно безопасной и отказоустойчивой многопользовательской работы с одними и теми же данными.

В архитектуре "клиент-сервер" используются так называемые "удаленные" (или "промышленные") СУБД. Промышленными они называются из-за того, что именно СУБД этого класса могут обеспечить работу информационных систем масштаба среднего и крупного предприятия, организации, банка. Локальные СУБД предназначены для однопользовательской работы или для обеспечения работы информационных систем, рассчитанных на небольшие группы пользователей.

К разряду промышленных СУБД принадлежат: Oracle, Gupta, Informix, Sybase, MS SQL Server, DB2, InterBase и ряд других.

Как правило, SQL-сервер управляется отдельным сотрудником или группой сотрудников (администраторы SQL-сервера). Они управляют физическими характеристиками баз данных, производят оптимизацию, настройку и переопределение различных компонентов БД, создают новые БД, изменяют существующие и т.д., а также выдают привилегии (разрешения на доступ определенного уровня к конкретным БД, SQL-серверу) различным пользователям.

Кроме этого, существует отдельная категория сотрудников, называемых администраторами баз данных. Как правило, это администраторы сервера, разработчики БД или пользователи, имеющие привилегии на создание, изменение, настройку оптимальных параметров отдельных серверных БД.

Администраторы БД также отвечают за предоставление прав на разноуровневый доступ к сопровождаемым ими БД для других пользователей.

Использование архитектуры "клиент-сервер":

- резко уменьшает сетевой трафик;
- понижает сложность приложений-клиентов (поскольку тем уже нет необходимости обеспечивать целостность и безопасность БД и следить за параметрами многопользовательской работы с БД);
- понижает требования к аппаратным средствам, на которых эти приложения функционируют (т.е. к компьютерам пользователей-клиентов);
- повышает надежность БД, ее целостность, безопасность и секретность.

## 2.4 Проблемы проектирования БД

Объединение программного обеспечения СУБД, прикладного программного обеспечения, реализованной базы данных, операционной системы (ОС) и аппаратных средств в одну систему для информационного обслуживания пользователей известно под названием система баз данных. Хотя технология применения СУБД, ОС и прикладных программ, хорошо известна, необходимо уделить внимание эффективному использованию этих средств с различными структурами баз данных. Так, главная проблема, стоящая перед администратором БД, заключается не в том, как использовать ее наиболее эффективно. Эта проблема может быть сформулирована в виде нескольких вопросов, возникающих в течение жизненного цикла приложения:

- что представляют собой требования пользователей и в какой форме они могут быть выражены;
- как эти требования могут быть преобразованы в эффективную структуру базы данных;
- как часто и каким образом структура базы данных должна перестраиваться в соответствии с новыми и/или изменяющимися требованиями.

Процесс разработки структуры базы данных в соответствии с требованиями пользователей называется проектированием базы данных.

Достижение приемлемого для всех пользователей уровня эксплуатационных характеристик базы данных является сложной задачей. Проектировщик БД должен постоянно помнить о стоимости различных услуг, предоставляемых пользователем одной или нескольких интегрированных БД. Ожидаемая экономия памяти и широкое использование базы данных в деятельности организации должна сопровождаться критическим анализом потенциального снижения качества обслуживания некоторых пользователей. Этой невозможности необходимо избегать. Целью должно быть – приемлемые эксплуатационные характеристики для всех пользователей.

Другим аспектом функционирования БД является ее гибкость. БД, тесно привязанные к текущим приложениям, могут иметь слишком ограниченную сферу применения в других подобных организациях. Быстрое изменение

требований и введение новых типов элементов данных могут иметь следствием повышение стоимости сопровождения программ, разложение временных файлов и сортировок, а также снижение производительности системы. [4]

## 2.5 Выбор программных средств реализации системы

Выбор программного средства важная деталь реализации программного средства, так как вся дальнейшая работа по реализации будет складываться из свойств и возможностей данной программной среды.

Сегодня основы современной информационной технологии составляют базы данных (БД) и системы управления базами данных (СУБД), роль которых как единого средства хранения, обработки и доступа к большим объемам информации постоянно возрастает. База Данных – это структурированная определенным образом совокупность данных, относящихся к конкретной задаче. СУБД представляет собой комплекс инструментальных средств (программных и языковых) реализующих централизованное управление БД и обеспечивающих доступ к данным (изменения, добавления, удаления, резервного копирования и т.д.). СУБД должна обеспечивать поиск, модификацию и сохранность данных, а также оперативный доступ (время отклика), защиту целостности данных от аппаратных сбоев и программных ошибок, разграничение прав и защита от несанкционированного доступа, поддержка совместной работы нескольких пользователей с данными.

Современные СУБД в основном являются приложениями Windows, так как данная среда позволяет более полно использовать возможности персональной ЭВМ, нежели среда DOS. Снижение стоимости высокопроизводительных ПК обусловил не только широкий переход к среде Windows, где разработчик программного обеспечения может в меньшей степени заботиться о распределении ресурсов, но также сделал программное обеспечение ПК в целом и СУБД в частности менее критичными к аппаратным ресурсам ЭВМ.

Реализация дипломного проекта проводится в системе программирования Borland Delphi, располагающей широкими возможностями по созданию приложений баз данных. Уже с более ранних версии система Delphi снабжена необходимым набором драйверов для доступа к самым известным форматам баз данных, удобными и развитыми средствами для доступа к информации, расположенной как на локальном диске, так и на удаленном сервере. В поставку продукта входит большое количество коллекций визуальных компонент для построения отображаемых на экране окон, что необходимо для создания удобного интерфейса между пользователем и исполняемым кодом.

Поскольку использование баз данных является одним из краеугольных камней, на которых построено существование различных организаций, пристальное внимание разработчиков приложений баз данных вызывают

инструменты, при помощи которых такие приложения можно было бы создавать. Выдвигаемые к ним требования в общем виде можно сформулировать как: «быстрота, простота, эффективность, надежность».

Среди большого разнообразия продуктов для разработки приложений Delphi занимает одно из ведущих мест. Delphi отдают предпочтение разработчики с разным стажем, привычками, профессиональными интересами. С помощью Delphi написано колоссальное количество приложений, десятки фирм и тысячи программистов-одиночек разрабатывают для Delphi дополнительные компоненты [4].

В основе такой общепризнанной популярности лежит тот факт, что Delphi, как никакая другая система программирования, удовлетворяет изложенным выше требованиям. Действительно, приложения с помощью Delphi разрабатываются быстро, причем взаимодействие разработчика с интерактивной средой Delphi не вызывает внутреннего отторжения, а наоборот, оставляет ощущение комфорта. Delphi-приложения эффективны, если разработчик соблюдает определенные правила (и часто - если не соблюдает). Эти приложения надежны и при эксплуатации обладают предсказуемым поведением [4].

Пакет Delphi - продолжение линии компиляторов языка Pascal корпорации Borland. Pascal как язык очень прост, а строгий контроль типов данных способствует раннему обнаружению ошибок и позволяет быстро создавать надежные и эффективные программы. Корпорация Borland постоянно обогащала язык. Когда-то в версию 4.0 были включены средства отдельной трансляции, позже, начиная с версии 5.5, появились объекты, а в состав шестой и седьмой версий пакета вошла полноценная библиотека классов Turbo Vision, реализующая оконную систему в текстовом режиме работы видеоадаптера. Это был один из первых продуктов, содержащих интегрированную среду разработки программ.

В классе инструментальных средств, для начинающих программистов продуктам компании Borland пришлось конкурировать со средой Visual Basic корпорации Microsoft, где вопросы интеграции и удобства работы были решены лучше. Когда в начале 70-х годов Н. Вирт опубликовал сообщение о Pascal, это был компактный, с небольшим количеством основных понятий и зарезервированных слов язык программирования, нацеленный на обучение студентов. Язык, на котором предстоит работать пользователю Delphi, отличается от исходного не только наличием множества новых понятий и конструкций, но и идейно: в нем вместо минимизации числа понятий и использования самых простых конструкций (что, безусловно, хорошо для обучения, но не всегда оправдано в практической работе), предпочтение отдается удобству работы профессионального пользователя. Как язык Turbo Pascal естественно сравнивать с его ближайшими конкурентами - многочисленными вариациями на тему языка Basic (в первую очередь с Visual Basic корпорации Microsoft) и с C++ [4]. Turbo Pascal существенно превосходит Basic за счет полноценного объектного подхода, включающего в себя развитые

механизмы инкапсуляции, наследование и полиморфизм. Последняя версия языка, применяемая в Delphi, по своим возможностям приближается к C++. Из основных механизмов, присущих C++, отсутствует только множественное наследование. (Впрочем, этим красивым и мощным механизмом порождения новых классов пользуется лишь небольшая часть программистов, пишущих на C++).

Плюсы применения языка Pascal очевидны: с одной стороны, в отличие от Visual Basic, основанного на интерпретации промежуточного кода, для него имеется компилятор, генерирующий машинный код, что позволяет получать значительно более быстрые программы. С другой - в отличие от C++ синтаксис языка Pascal способствует построению очень быстрых компиляторов [6].

Среда программирования представляет собой несколько отдельных окон: меню и инструментальные панели, Object Inspector (в котором можно видеть свойства объекта и связанные с ним события), окна визуального построителя интерфейсов (Visual User Interface Builder), Object Browser (позволяющее изучать иерархию классов и просматривать списки их полей, методов и свойств), окна управления проектом (Project Manager) и редактора.

Delphi содержит полноценный текстовый редактор типа Brief, назначения клавиш в котором соответствуют принятым в Windows стандартам, а глубина иерархии операций Undo неограниченна. Как это стало уже обязательным, реализовано цветное выделение различных лексических элементов программы. Процесс построения приложения достаточно прост. Нужно выбрать форму (в понятие формы входят обычные, диалоговые, родительские и дочерние окна MDI), задать ее свойства и включить в нее необходимые компоненты (видимые и, если понадобится, неотображаемые): меню, инструментальные панели, строку состояния и т. п., задать их свойства и далее написать (с помощью редактора исходного кода) обработчики событий. Окна типа Object Browser стали неотъемлемой частью систем программирования на объектно-ориентированных языках. Работа с ними становится возможной сразу после того, как вы скомпилировали приложение.

Project Manager - это отдельное окно, где перечисляются модули и формы, составляющие проект. При каждом модуле указывается маршрут к каталогу, в котором находится исходный текст. Жирным шрифтом выделяются измененные, но еще не сохраненные части проекта. В верхней части окна имеется набор кнопок: добавить, удалить, показать исходный текст, показать форму, задать опции и синхронизировать содержимое окна с текстом файла проекта, т. е. с головной программой на языке Pascal.

Опции, включая режимы компиляции, задаются для всего проекта в целом. В этом отношении традиционные make-файлы, используемые в компиляторах языка C, значительно более гибки.

Visual Component Library (VCL). Богатство палитры объектов для построения пользовательского интерфейса - один из ключевых факторов при выборе инструмента визуального программирования. При этом для пользователя имеет значение как число элементов, включенных

непосредственно в среду, так и доступность элементов соответствующего формата на рынке. [4].

Совместимость с программами, созданными ранее средствами Borland Pascal, сохраняется, несмотря на то, что в язык внесены существенные изменения. Необходимость в некоторых усовершенствованиях давно ощущалась. Самое заметное из них - аппарат исключительных ситуаций, подобный тому, что имеется в C++, был первым реализован в компиляторах корпорации Borland. Не секрет, что при написании объектно-ориентированных программ, активно работающих с динамической памятью и другими ресурсами, немалую трудность представляет аккуратное освобождение этих ресурсов в случае возникновения нештатных ситуаций. Особенно это актуально для среды Windows, где число видов ресурсов довольно велико, а неправильная работа с ними может быстро привести к зависанию всей системы. Предусмотренный в Delphi аппарат исключений максимально упрощает кодирование обработки нештатных ситуаций и освобождения ресурсов.

Объектно-ориентированный подход в новой версии языка получил значительное развитие. Перечислим основные новшества:

- введено понятие класса;

- реализованы методы классов, аналогичные статическим методам C++. Они оперируют не экземпляром класса, а самим классом;

- механизм инкапсуляции во многом усовершенствован. Введены защищенные поля и методы, которые, подобно приватным, не видны извне, но отличаются от них тем, что доступны из методов класса- наследника;

- введена обработка исключительных ситуаций. В Delphi это устроено в стиле C++. Исключения представлены в виде объектов, содержащих специфическую информацию о соответствующей ошибке (тип и местонахождение ошибки). Разработчик может оставить обработку ошибки, существовавшую по умолчанию, или написать свой собственный обработчик. Обработка исключений реализована в виде exception-handling blocks (также еще называется protected blocks), которые устанавливаются ключевыми словами try и end. Существуют два типа таких блоков: try...except и try...finally;

- появилось несколько удобных синтаксических конструкций, в числе которых преобразование типа объекта с контролем корректности (в случае неудачи инициируется исключение) и проверка объекта на принадлежность классу.

Ссылки на классы придают дополнительный уровень гибкости, так, когда вы хотите динамически создавать объекты, чьи типы могут быть известны только во время выполнения кода. К примеру, ссылки на классы используются при формировании пользователем документа из разного типа объектов, где пользователь набирает нужные объекты из меню или палитры. Собственно, эта технология использовалась и при построении Delphi.

Введено средство, известное как механизм делегирования. Под делегированием понимается то, что некий объект может предоставить другому объекту отвечать на некоторые события. Он используется в Delphi для



упрощения программирования событийно-ориентированных частей программ, т. е. пользовательского интерфейса и всевозможных процедур, запускаемых в ответ на манипуляции с базой данных.

После того как Borland внесла перечисленные изменения, получился мощный объектно-ориентированный язык, сопоставимый по своим возможностям с C++. Платой за новые функции стало значительное повышение требований к профессиональной подготовке программиста.

Язык программирования Delphi базируется на Borland Object Pascal.

Кроме того, Delphi поддерживает такие низкоуровневые особенности, как подклассы элементов управления Windows, перекрытие цикла обработки сообщений Windows, использование встроенного ассемблера.[4].

Основной упор этой модели в Delphi делается на максимальном повторном использовании кода. Это позволяет разработчикам строить приложения весьма быстро из заранее подготовленных объектов, а также дает им возможность создавать свои собственные объекты для среды Delphi. Никаких ограничений по типам объектов, которые могут создавать разработчики, не существует. Действительно, все в Delphi написано на нем же, поэтому разработчики имеют доступ к тем же объектам и инструментам, которые использовались для создания среды разработки. В результате нет никакой разницы между объектами, поставляемыми Borland или третьими фирмами, и объектами, которые можно создать самостоятельно.

В стандартную поставку Delphi входят основные объекты, которые образуют удачно подобранную иерархию из 270 базовых классов. На Delphi можно одинаково хорошо писать как приложения к корпоративным базам данных, так и, к примеру, игровые программы. Во многом это объясняется тем, что традиционно в среде Windows было достаточно сложно реализовывать пользовательский интерфейс. Событийная модель в Windows всегда была сложна для понимания и отладки. Но именно разработка интерфейса в Delphi является самой простой задачей для программиста.

Благодаря такой возможности приложения, изготовленные при помощи Delphi, работают надежно и устойчиво. Delphi поддерживает использование уже существующих объектов, включая DLL, написанные на C и C++, OLE сервера, VBX, объекты, созданные при помощи Delphi. Из готовых компонент работающие приложения собираются очень быстро. Кроме того, поскольку Delphi имеет полностью объектную ориентацию, разработчики могут создавать свои повторно используемые объекты для того, чтобы уменьшить затраты на разработку.

Delphi предлагает разработчикам - как в составе команды, так и индивидуальным - открытую архитектуру, позволяющую добавлять компоненты, где бы они ни были изготовлены, и оперировать этими вновь введенными компонентами в визуальном построителе. Разработчики могут добавлять CASE-инструменты, кодовые генераторы, а также авторские help'ы, доступные через меню Delphi. [6].

Мощность и гибкость Delphi при работе с базами данных основана на

низкоуровневом ядре - процессоре баз данных Borland Database Engine (BDE). Его интерфейс с прикладными программами называется Integrated Database Application Programming Interface (IDAPI). В принципе, сейчас не различают эти два названия (BDE и IDAPI) и считают их синонимами. BDE позволяет осуществлять доступ к данным как с использованием традиционного record-ориентированного (навигационного) подхода, так и с использованием set-ориентированного подхода, используемого в SQL-серверах баз данных. Кроме BDE, Delphi позволяет осуществлять доступ к базам данных, используя технологию (и, соответственно, драйверы) Open DataBase Connectivity (ODBC) фирмы Microsoft. Но, как показывает практика, производительность систем с использованием BDE гораздо выше, чем оных при использовании ODBC. ODBC драйвера работают через специальный «ODBC socket», который позволяет встраивать их в BDE.

Все инструментальные средства баз данных Borland - Paradox, dBase, Database Desktop - используют BDE. Все особенности, имеющиеся в Paradox или dBase, «наследуются» BDE, и поэтому этими же особенностями обладает и Delphi.

Библиотека объектов содержит набор визуальных компонентов, которые значительно упрощают разработку приложений для баз данных клиент-сервер. Объекты инкапсуляции более низкий уровень - Borland Database Engine.

Существуют специальные наборы компонент, отвечающий за доступ к данным и отображения данных компонент. Компоненты доступа к данным позволяют соединения с базой данных, выборка, копирования данных и т.д.

Компоненты визуализации данных может отображать данные в виде таблиц, полей, списков. Отображаемые данные могут быть текст, изображение или нестандартного размера.

Таблица хранится в базе данных. Различные базы данных СУБД хранятся в виде нескольких отдельных файлов, представляющих таблицы (как правило, все местные СУБД), а другие состоят из одного файла, содержащего все таблицы и индексы (InterBase). Например, базы данных и Paradox таблицы всегда сохраняются в отдельных файлах на диске. Каталоги, содержащие Dbase. DBF файлы или Paradox. DB файлы, рассматривается как база данных. Другими словами, любые каталоги, содержащие файлы в Paradox, DBase, Delphi рассматривать как единую базу данных. Для переключения на другую базу данных, нужно просто перейти в другой каталог. InterBase сохраняет все таблицы в одном файле с расширением. GDB, поэтому этот файл является базой данных InterBase.

Объекты базы данных в Delphi на основе SQL и включают в себя полную мощь Borland Database Engine. В состав также входит Delphi Borland SQL Link, поэтому доступ к базе данных Oracle, Sybase, Informix и InterBase происходит с высокой эффективностью. Кроме того, Delphi включает в себя локальный сервер Interbase для того, чтобы иметь возможность развивать масштабируемый к любым внешним SQL-серверных приложений в автономном режиме. Среда Разработчик Delphi, проектирование

информационной системы для локальной машине (например, небольшая система учета медицинских записей для одного компьютера) может быть использован для хранения файлов в формате информационных. Dbf (в Dbase или Clipper) или. Дб (Paradox). Если он будет использовать локальный InterBase для Windows 4.0 (это локальный SQL-сервер входит в комплект поставки), его применение без изменений и будет работать как часть более крупной системы с архитектурой клиент-сервер.

Масштабируемость на практике - одно и то же приложение можно использовать как для локального, так и для более серьезного клиент-серверного вариантов.[7].

MS SQL Server 2008 - реляционная система управления базами данных. Следующие версии MS SQL Server 2008:

- Enterprise Edition. Наиболее мощная версия СУБД - сервер издание, установить на крупных предприятиях. Обеспечивает поддержку многопроцессорной конфигурации и различных аппаратных платформах. Этот вариант лучше всего подходит для реализации интернет-решений. Содержит все функции, поддерживаемые в SQL Server, может работать с объемами баз данных, превышающих 2 Гб;

- Developer Edition. Базы данных, ориентированных на разработчиков вариант. Имеет почти полный набор функциональности Enterprise Edition, но установка этой версии допускается только на рабочих станциях;

- Стандарт Edition - сервер СУБД издание. Не все поддерживают серверы и сетевые функции Enterprise Edition, работающий на средних предприятий. Не поддерживает более 4 процессоров и до 4 Гб оперативной памяти;

- Personal Edition. Версия СУБД для реализации небольших баз данных, имеющихся на локальном компьютере клиента. Может быть использован для небольшой рабочей группы имеет ограниченная (до 5) количество клиентских подключений к сети;

- SQL Server CE издание. Версия баз для портативных ПК работаете Windows CE;

- Desktop Edition. Версия SQL Server способен манипулировать данные, извлеченные из базы данных создается, построенный в других версий базы данных, он может работать пользовательских приложений без выполнения каких-либо функции администратора.

Управление MS SQL Server 2008

По инструментов, позволяющих обеспечить простой и эффективное использование MS SQL Server являются:

- Enterprise Manager;
- Service Manager;
- Query Analyzer.

Enterprise Manager - является частью MS Management Console (MMC, консоль управления MS). Enterprise Manager может остановить и запустить сервер; настроить локальную и удаленную сервер; зарегистрировать сервер;

настроить логин, зарегистрируйтесь и добавлять новых пользователей, операторов, системных администраторов; создания и планирования работы; создать базы данных, таблицы, индексы, представления, хранимые процедуры, триггеры; другие услуги по управлению SQL Server. Менеджер MS SQL Server Enterprise, выполнив команду главного меню Пуск | Программы | MS SQL Server | Enterprise Manager. Структура сервере представлена в виде иерархического дерева Enterprise Manager (слева), чьи узлы рухнул в исходном состоянии (разворачивается нажмите на знак +). При выборе список узлов ее компонентов появится в правой панели. Просмотр правильное изменение форточки команды меню View. Режимы просмотра меню Вид следующим образом: Крупные значки, Мелкие значки, Список, Таблица. Существует также еще один режим - Панель задач. Это компоненты базы данных панели, которые перемещаются из одного компонента к другому, нажимая Tab. Доступна с панели задач: общие сведения, сведения о таблицах, мастер. Часть «информационных столов» для этого поле, "Поиск" для поиска объекта базы данных. Компонент «хозяина» дает доступ к ряду инструментов, встроенных в SQL Server.

Утилита Service Manager (Service Control Manager) можно использовать как быстрый и простой в использовании интерфейс для управления сервисами SQL Server. Обычно это работает в фоновом режиме и минимально представлен на экране в правом нижнем углу панели, которая имеет иконки, которые представляют запущенные процессы. Если значок находится в месте, начать диспетчера службы, дважды щелкнув на нем. Если этот значок не отображается, необходимо выполнить команду: Пуск | Программы | MS SQL Server | Service Manager главное меню. В Service Manager, необходимо выбрать сервер, который будет исследовать, выбрать сервис, который вы собираетесь переехать в другой штат: пауза, стоп полностью, а затем запустить снова.

Query Analyzer - другой внешний интерфейс командной строки для взаимодействия с процессором SQL Server. Запуск анализатора запросов в одном из двух способов:

- Из Enterprise Manager. Для этого необходимо выбрать сервер, на который Query Analyzer следует подключить при запуске. Затем выберите пункт меню Tools (Инструменты) и в нем, работать SQL Query Analyzer. С этого момента Query Analyzer работает и готов выполнять команды.

- из меню Пуск. Выполните Пуск | Программы | MS SQL Server | Query Analyzer. Появляется Логин диалоговое окно, которое должны заполнить поле и установите флажок Начать SQL Server, если останавливается (начать SQL Server, если она остановлена). Обратите внимание на точку (.) Указано в SQL Server вместо имени сервера. Эта точка относится к местной SQL Server. Если SQL Server устанавливается на удаленном компьютере, в поле со списком следует указать имя компьютера, который может быть выбран, нажав на многоточие (...) справа.

Вид плиты объектов Object Browser расположенный на левой стороне окна Query Analyzer. Обзор объектов состоит из двух вкладок: объекты

(объект) и шаблонов (шаблоны). Выбор сделан в нижней части панели вкладок. В Object Browser представляет собой список объектов в SQL Server, что связано с, а также функций, которые можно использовать, чтобы сделать запросы или хранимые процедуры, а также их описания. Здесь есть все типы данных определяемые системой в SQL Server. SQL Server включает в себя ряд встроенных баз данных. Четыре из них используются SQL Server, более двух - демонстрации. Объекты, которые составляют базу данных (независимо от того, эта база данных - система, или демо создается пользователем), называются системными объектами. При первом открытии окна Query Analyzer все узлы дерева рухнул системные объекты. Узел развернуты кликов от знаков (+). Цель Объекты, расположенные в узле общий объект - это объекты, которые не принадлежат к любой базе данных. Этот сайт расположен функции, которые могут быть доступны в SQL Server, типы данных, доступные пользователю, справки по всем командам SQL. Шаблоны расположены на шаблонах вкладке - как встроенный в SQL Server, а также нестандартные. Pattern - это файл, содержащий основные операторы некоторых процедур, используемых в Query Analyzer. Панель Запрос находится в окне Object Browser справа. В панели результатов используется для отображения результатов и сообщений от SQL-исполняемый код, работающий с панели запроса. Это всегда под запросы панели. Вывод по умолчанию не выполняется в нем. Для того чтобы это произошло, панель должна быть включена. В состав экспертов вошли запуск кнопки запроса на панели инструментов или команду меню Windows, | Показать Результат панель. Вы можете использовать комбинацию клавиш CTRL + R для включения / выключения панели результатов.

#### Структура базы данных MS SQL Server 2008

Каждая база данных SQL Server состоит из набора файлов операционной системы. Эти файлы могут быть сгруппированы в группы файлов, что делает их легче управлять, помогает в размещении данных и повышения производительности. Файл базы данных может быть либо файл данных или файл журнала. Файлы данных используются для хранения данных и объектов, таких как таблицы, индексы, представления, триггеры и хранимые процедуры.

. Имеются два типа файлов данных: первичные и вторичные. Файлы журналов служат только для хранения информации из журналов транзакций. Место на диске, отводимое для файлов журналов всегда должно администрироваться отдельно от места, отводимого для данных, и никогда не должно быть частью файла данных.

Каждая база данных должна создаваться хотя бы с одним файлом данных и с одним файлом журнала; файлы не могут быть использованы более чем в одной базе данных – т.е., базы данных не могут разделять файлы (или использовать их совместно).

Первичные файлы данных содержат всю информацию для запуска базы данных и ее системных таблиц и объектов. Они указывают на другие файлы, созданные в базе данных. Они могут также содержать таблицы и объекты, задаваемые пользователем, хотя это и необязательно. Каждая база данных

может иметь только один первичный файл. Для этих файлов рекомендуется применять расширение .mdf.

Вторичные файлы данных не являются обязательными. Они могут хранить данные и объекты, которые отсутствуют в первичном файле. База данных может вообще не иметь ни одного вторичного файла (если все ее данные хранятся в первичном файле). Можно иметь ноль, один или несколько вторичных файлов. Для некоторых баз данных требуется иметь несколько вторичных файлов, чтобы размещать данные по нескольким отдельным дискам. Для этих файлов рекомендуется применять расширение .ndf.

Файлы журналов транзакций хранят всю информацию из журнала транзакций, служащую для восстановления базы данных. Каждая база данных должна иметь хотя бы один файл журнала, а может иметь несколько файлов журналов. Для этих файлов рекомендуется применять расширение .ldf.

База данных SQL Server – это контейнер объектов, предназначенных для хранения, записи и извлечения данных в безопасном режиме – содержит следующие компоненты:

- **Таблицы.** Это объекты, внутри которых хранятся данные БД. Информация хранится в таблицах в форме строк и столбцов. Столбцы таблицы определяют простейшие элементы информации, определяют назначение и тип данных, которые в нем хранятся и объем элемента этих данных. Таблица должна состоять минимум из одного столбца. Строка образуется одним элементом информации от каждого столбца. Таблица может содержать любое число строк – оно ограничивается местом свободного пространства на диске. Строка определяет блок информации, состоящий из элементов, хранящихся в столбцах. Другое название строки – запись.

- **Хранимые процедуры.** Программный код, состоящий из одного или нескольких операторов T-SQL, которые уже откомпилированы, хранится в БД и может быть запущена в любой момент.

- **Индексы** можно рассматривать как некие списки с информацией о расположении и упорядочении данных. Используются для ускорения извлечения данных из таблиц БД.

- **Представления.** Особый способ просмотра данных, хранящихся в БД. Среда Delphi была использована для разработки данной программы, так как она самое эффективное средство разработки приложений баз данных, то есть программ, обслуживающих электронные хранилища информации. Эффективность определяется следующими обстоятельствами: высокопроизводительной машиной доступа к данным разного формата (BDE), наличием многочисленных компонентов, ориентированных на эту сферу применения.

Общесистемное программное обеспечение.

В соответствии с техническим заданием разрабатываемый программный комплекс должен работать под управлением операционной системы семейства Windows. На данный момент наиболее распространены операционные системы

Windows XP, Windows 2000, Windows 7. Они удовлетворяют следующим требованиям:

- развитой файловой системе;
- приемлемой стоимости;
- многозадачности;
- удобному интерфейсу;
- широкому распространению в Казахстане;
- поддержке дополнительных аппаратных средств (звуковой карты, DVD-ROM, сканеров, принтеров и т.д.).

В настоящее время наибольшее развитие и внедрение получила операционная система Windows XP. Выбор этой информационной системы обусловлен ее преимуществами, перечисленными ниже.

**Совместимость.** Система имеет привычный интерфейс операционных систем семейства Windows, с некоторыми добавлениями и расширениями. Большинство приложений, написанных под MSDOS, W9x, а также некоторые программы под OS/2 и POSIX запускаются и функционируют без проблем.

**Переносимость.** Система работает на различных процессорах производства Intel и AMD.

**Система безопасности.** Реализована система безопасности на уровне пользователей.

**Распределённая обработка.** Windows XP имеет встроенные в систему сетевые возможности, что обеспечивает возможность связи с различными типами компьютеров благодаря наличию разнообразных транспортных протоколов и технологии "клиент-сервер".

**Надёжность и отказоустойчивость.** Архитектура операционной системы защищает приложения от повреждения друг другом и самой системой. При этом используется отказоустойчивая структурированная обработка особых ситуаций на всех архитектурных уровнях, которая включает восстанавливаемую файловую систему NTFS и обеспечивает защиту с помощью встроенной системы безопасности и усовершенствованных методов управления памятью.

**Локализация.** Программный комплекс предоставляет возможности для работы во многих странах мира на национальных языках, что достигается применением стандарта ISO Unicode.

Драйвер баз данных Microsoft Jet 4.0 OLE DB Provider.

Драйвер Microsoft Jet 4.0 OLE DB Provider используется для обращения к базам данных, созданных с помощью СУБД от Microsoft (MS SQL, MS Access). Драйвер автоматически устанавливается при их инсталляции. Начиная с Windows 2000, он поставляется в комплекте с базовыми драйверами системы.

## **Выбор технических средств реализации системы**

Состав и основные характеристики ПЭВМ, предназначенной для

решения рассматриваемого комплекса поставленных задач приведены в таблице 2.1.

Т а б л и ц а 2 . 1 - Технические характеристики ПЭВМ

Описание характеристики	Характеристики
<b>Конфигурация системного блока</b>	
Процессор	Pentium III
Тактовая частота	Min 800 MHz
Кэш-память	Min 256 Kb
Оперативная память	Min 512 Mb
Емкость жесткого диска	20 Gb
Видеопамять	Min 4 Mb
CD-ROM привод	40x
Floppy Disk	1.44 Mb
<b>Монитор</b>	
Размер	17'
Безопасность	TCO-95-99
Мультимедиа	Нет
<b>Принтер</b>	
Тип	Лазерный, струйный
Формат	A4
Манипулятор-мышь (PS/2)	
Клавиатура (PS/2)	

## 2.7 Анализ информационных потоков

Методика разработки информационной модели предполагает моделирование:

- взаимосвязей входных, промежуточных и результатных информационных потоков и функций предметной области;
- данных информационной базы (диаграмму “сущность-связь” - инфологической модели и диаграмму взаимосвязей файлов - даталогической модели), необходимых для функционирования информационной системы.

Информационная модель изображена на рисунке 2.3.



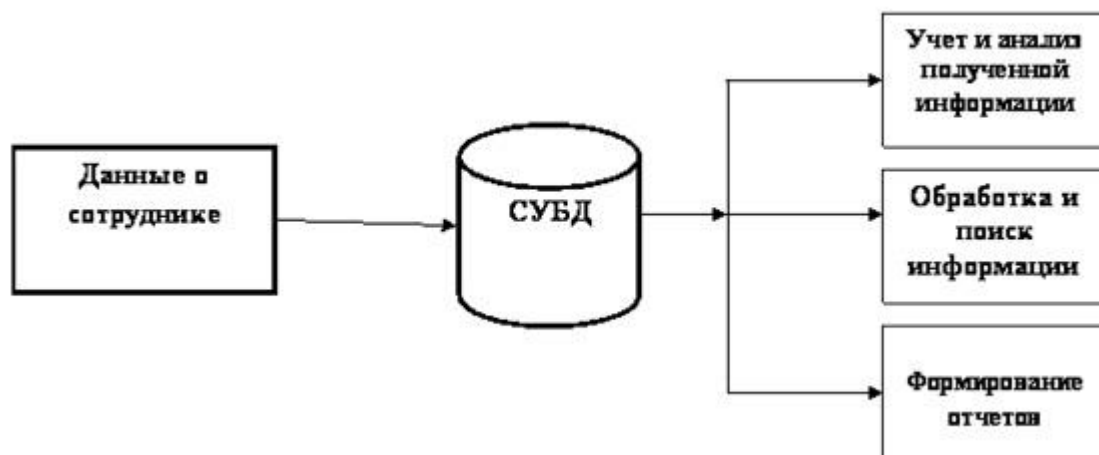


Рисунок 2.3 - Информационная модель задачи "Разработка ПО учета кадрового состава"

При приеме человека на работу после заключения трудового договора специалистом отдела кадров вручную заполняется личная карточка сотрудника и приказ о приеме на работу. После подписания приказа директором, специалист по кадрам регистрирует его в журнале регистрации приказов по личному составу. Приказ о приеме на работу остается в кадровой службе.

Дальнейшие действия специалиста по кадрам следующие:

в трехдневный срок обязан ознакомить работника с приказом. Работник ознакомление с приказом удостоверяет своей личной подписью;

готовит заверенные копии приказа, один экземпляр помещает в личное дело работника, другой передает в бухгалтерию, где работнику присваивается табельный номер и на него открывается лицевой счет;

на основании приказа о приеме на работу заполняет личную карточку работника (форма-Т2);

на основании приказа о приеме на работу вносится запись в трудовую книжку.

Хранение информации о сотрудниках на бумажных носителях включает в себя:

1) первичную информацию о сотруднике (фамилия, имя, отчество, уволен или является в данный момент работником организации, должность, отдел, табельный номер);

2) паспортные данные сотрудника (год рождения, место рождения, пол, адрес, телефон, вид документа, серия, номер, дата выдачи, кем выдан документ);

3) данные о семейном положении сотрудника (семейное положение, члены семьи (ФИО, кем являются, год рождения));

4) данные об образовании сотрудника (учебное заведение, специальность, год поступления, год окончания, серия и номер диплома);

5) дополнительные данные сотрудника.

Приказы об увольнении и о переводе на другую должность заполняются в двух экземплярах, т.к. один из экземпляров каждого документа отправляется в бухгалтерию. Оставшиеся экземпляры подшиваются соответственно в папки «Приказы о прекращении действия трудового договора (контракта) с работником» и «Приказы о переводе сотрудника на другую работу».

Прекращение трудового договора должно оформляться актом работодателя (приказом или распоряжением), за исключением прекращения трудового договора в случае смерти (объявления судом умершим или признания безвестно отсутствующим) работодателя – физического лица и прекращения трудового договора с домашними работниками.

В соответствии с трудовым кодексом РК основаниями прекращения трудового договора являются:

- 1) расторжение трудового договора по соглашению сторон;
- 2) истечение срока трудового договора;
- 3) расторжение трудового договора по инициативе работодателя;
- 4) расторжение трудового договора по инициативе работника;
- 5) обстоятельства, не зависящие от воли сторон;
- 6) отказ работника от продолжения трудовых отношений;
- 7) переход работника на выборную работу (должность) или назначение его на должность, исключающую возможность продолжения трудовых отношений, кроме случаев, предусмотренных законами Республики Казахстан;
- 8) нарушение условий заключения трудового договора;
- 9) основания, предусмотренные в трудовом договоре, заключенном с руководителем исполнительного органа работодателя.

Проект приказа о расторжении трудового договора по инициативе работника составляется специалистом по кадрам (менеджером по персоналу) в текстовой форме на бланке приказа на основании заявления работника.

Текст приказа в этом случае должен начинаться словом «расторгнуть». Далее указываются: фамилия, имя, отчество, должность (профессия) и структурное подразделение, основание прекращения трудового договора статья 51 (с указанием пункта основания расторжения) в соответствии с Трудовым кодексом РК.

Проект приказа о расторжении трудового договора визируют специалист по кадрам (менеджер по персоналу), руководитель структурного подразделения, из которого увольняется работник, юрист, если расторжение производится по инициативе работодателя.

Руководитель организации подписывает приказ, специалист по кадрам (менеджер по персоналу) датирует и регистрирует приказ в журнале регистрации приказов по личному составу.

Подлинник приказа о расторжении трудового договора остается на хранении в кадровой службе. На его основании специалист по кадрам (менеджер по персоналу) готовит копии приказа, один из которых передает в бухгалтерию для выплаты сумм, причитающихся работнику от работодателя, второй, вместе с заявлением о расторжении трудового договора, помещает в

личное дело работника, третий вручает работнику либо направляет ему письмом с уведомлением в трехдневный срок.

Личное дело работника закрывается делопроизводством и перекладывается в картотеку личных дел уволенных сотрудников.

Сведения о расторжении трудового договора специалист по кадрам (менеджер по персоналу) заносит в трудовую книжку и личную карточку работника (форма Т-2).

При прекращении или расторжении с работником трудового договора в трудовую книжку работника вносятся все записи о работе (прием, перевод, прекращение или расторжение трудового договора), награждениях и поощрениях за время работы у работодателя и заверяют подписью работодателя (руководителя организации), печатью организации.

Работодатель выдает работнику трудовую книжку в день прекращения трудового договора с внесенной в нее записью о прекращении трудового договора.

Отчеты, предоставляемые работником отдела кадров руководству предприятия, составляются путем просмотра необходимых документов за определенный период, поиска, подсчета и группировки данных, заполнения принятой на предприятии формы отчетности. Отчеты предоставляются директору предприятия, а их копии хранятся в отделе кадров в папке «Отчеты».

Вся информация о сотрудниках организации хранится на бумажных носителях. При этом трудно осуществить быстрый отбор нужных данных при приеме на работу, уходе в отпуск, увольнении, переходе на другую должность или других перемещениях сотрудника. Так же большое внимание нужно уделить вопросу безопасности хранения данных и конфиденциальности личной информации сотрудника, что трудно обеспечить при хранении на бумажных носителях.

Для решения поставленной задачи необходимо наличие следующих справочников:

1) справочник *Отделы* – предназначен для хранения названий отделов предприятия;

2) справочник *Должности* – хранит информацию о должностях, существующих в организации;

3) справочник *Вид документа* – содержит информацию о виде (полном названии) документа для учета паспортных данных сотрудника;

4) справочник *Вид образования* – содержит информацию о видах образования (среднее общее, среднее профессиональное, высшее);

4) справочник *Виды отпуска* - содержит информацию обо всех видах отпусков, которые существуют (ежегодный, учебный, без сохранения заработной платы и другие).

Информация в систему вносится непосредственно из первичных документов:

1) анкета для приема на работу;

2) удостоверение сотрудника (паспортные данные);

3) служебная записка (от начальника отдела) о перемещении сотрудника с одной должности на другую;

4) заявление о приеме на работу от работника;

5) заявление об увольнении от сотрудника;

6) графики отпусков от начальников отделов.

Для решения поставленной задачи должны быть следующие входные и выходные документы.

Входные документы:

1) личная карточка сотрудника;

2) приказ о приеме на работу;

3) приказ об увольнении;

4) приказ о перемещении;

5) приказ об отпуске.

Выходные документы:

1) личная карточка сотрудника (хранимый документ) – выводит все личные данные сотрудника;

2) приказ о приеме на работу – формируется приказ по внесенным данным сотрудника для подписи руководителя;

3) приказ об увольнении;

4) приказ о перемещении;

5) приказ об отпуске;

6) штатное расписание;

7) ведомость приказов.

Входные документы содержат информацию, которая хранится в БД и необходима в процессе учета кадров и при составлении отчетов. А информация в выходных документах отражает результат процесса учета кадров на предприятии.

Каждому документу соответствует своя экранная форма для ввода переменной и условно-постоянной первичной информации, а также для вывода на экран результатной информации или ответов на запросы. Все формы разработаны в DELPHI. Они имеют поля для заполнения (соответствующие атрибутам документов), а также необходимые кнопки и справочную информацию. Заполнив экранную форму, мы имеем возможность распечатать документ на принтере.

## **2.8 Разработка структуры базы данных**

Основные этапы, на которые разбивается процесс проектирования базы данных информационной системы:

инфологическое проектирование - сбор, анализ и редактирование требований к данным. По окончании данного этапа получаем концептуальную модель, инвариантную к структуре базы данных. Часто она представляется в виде инфологической модели "сущность-связь";

логическое проектирование - преобразование требований к данным в структуры данных. На выходе получаем СУБД-ориентированную структуру базы данных и спецификации прикладных программ. На этом этапе часто моделируют базы данных применительно к различным и конкретным СУБД и проводят сравнительный анализ моделей. Создание даталогической модели;

физическое проектирование - определение особенностей хранения данных, методов доступа и т.д.

Объединяя частные представления о содержимом базы данных, полученные в результате опроса пользователей, и свои представления о данных, которые могут потребоваться в будущих приложениях, проектировщик сначала создает обобщенное неформальное описание создаваемой базы данных. Это описание, выполненное с использованием естественного языка, математических формул, таблиц, графиков и других средств, понятных всем людям, работающим над проектированием базы данных, называют инфологической моделью данных.

Цель инфологического моделирования - это обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком, (естественный язык не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка) [3].

Инфологическая (концептуальная) модель предметной области представляет собой описание структуры и динамики предметной области, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависящих от реализации БД. Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

Сущность – это любой различимый объект (объект, который мы можем отличить от другого объекта), информацию о котором необходимо хранить в базе данных. Сущностями могут быть люди, места, самолеты, рейсы, вкус, цвет и т.д. Необходимо различать такие понятия, как тип сущности и экземпляр сущности. Понятие тип сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое. Экземпляр сущности относится к конкретной вещи в наборе.

Атрибут – это поименованная характеристика сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различного типа сущностей. Атрибуты используются для определения того, какая информация должна быть собрана о сущности. Абсолютное различие между типами сущностей и атрибутами отсутствует. Атрибут является таковым только в связи с типом сущности. В другом контексте атрибут может выступать как самостоятельная сущность. Например, для автомобильного завода цвет – это только атрибут продукта производства, а для лакокрасочной фабрики цвет – тип сущности.

Ключ – это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся.

Связь – это ассоциирование двух или более сущностей. Между двумя сущностями возможно четыре типа связей:

- «один к одному»;
- «один ко многим»;
- «многие ко многим»;
- «многие к одному».

Существует три основных класса сущностей: стержневые, ассоциативные и характеристические, а также подкласс ассоциативных сущностей – обозначение.

Стержневая сущность (стержень) – это независимая сущность.

Ассоциативная сущность (ассоциация) – это связь вида "многие-ко-многим" ("-ко-многим" и т.д.) между двумя или более сущностями или экземплярами сущности. Ассоциации рассматриваются как полноправные сущности: они могут участвовать в других ассоциациях и обозначениях точно так же, как стержневые сущности; могут обладать свойствами, т.е. иметь не только набор ключевых атрибутов, необходимых для указания связей, но и любое число других атрибутов, характеризующих связь

Характеристическая сущность (характеристика) – это связь вида "многие-к-одной" или "одна-к-одной" между двумя сущностями (частный случай ассоциации). Единственная цель характеристики в рамках рассматриваемой предметной области состоит в описании или уточнении некоторой другой сущности. Необходимость в них возникает в связи с тем, что сущности реального мира имеют иногда многозначные свойства.

Обозначающая сущность или обозначение – это связь вида "многие-к-одной" или "одна-к-одной" между двумя сущностями и отличается от характеристики тем, что не зависит от обозначаемой сущности [5].

ER-диаграмма изображена на рисунке 2.4.

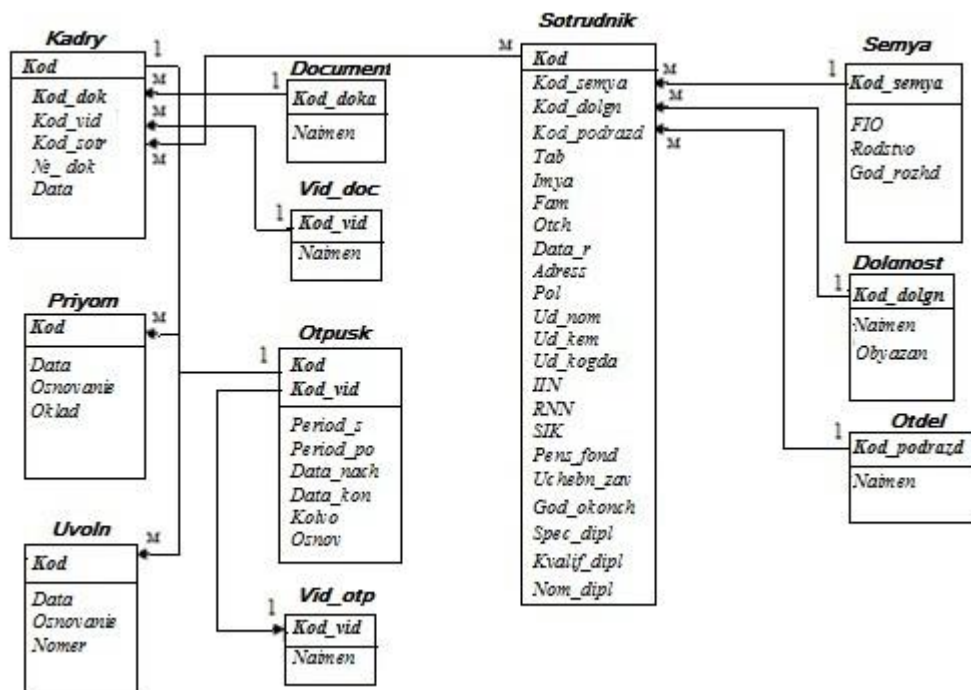


Рисунок 2.4 - ER-диаграмма задачи автоматизации рабочего места сотрудника отдела кадров

На этом рисунке сущности показаны вместе с атрибутами. Физически каждая сущность представляет собой таблицу БД, а атрибуты – поля в них. Опишем поля для таблиц БД, которые получаются из физической ER – модели.

Личная карточка сотрудника (таблица 2.2) соответствует сущности Kadry, и поля этой таблицы соответствуют атрибутам этой сущности. Сущность Kadry здесь будет соответствовать личному делу, заводимому на сотрудника.

Т а б л и ц а 2 . 2 - Личная карточка сотрудника

Поле	Расшифровка	Тип данных	Размер
Kod	Номер записи о сотруднике	Автоинкрементное (Autoincrement)	
Kod_dok	Идентификатор вида документа	Целое (Short)	Соответствует значениям связанной таблицы
Kod_semya	Идентификатор семейного положения	Целое (Short)	Соответствует значениям связанной таблицы
Kod_dolg	Идентификатор должности	Целое (Short)	Соответствует значениям связанной таблицы
Tab	Табельный номер	Текстовое (Alpha)	6

Личная карточка сотрудника  
таблицы 2.2

Продолжение

Imya	Имя	Текстовое(Alpha)	20
Fam	Фамилия	Текстовое(Alpha)	20
Otch	Отчество	Текстовое(Alpha)	30
Data_r	Дата рождения	Дата (Date)	
Adress	Адрес	Текстовое(Alpha)	3
Pol	Пол	Символ(Char)	
Ud_nom	Номер удостоверения	Целое(Short)	6
Ud_kem	Кем выдан	Текстовое(Alpha)	40
Ud_kogda	Когда выдан	Дата (Date)	
IIN	ИИН	Целое(Short)	15
RNN	РНН	Целое(Short)	15
SIK	СИК	Текстовое(Alpha)	30
Pens_fond	Пенсионный фонд	Текстовое(Alpha)	30

Атрибуты сущности Document соответствуют полям таблицы справочник видов документов (таблица 2.3). Сущность Document имеет связь с Kadru как один ко многим.

Т а б л и ц а 2 . 3 - Справочник Документ

Поле	Расшифровка	Тип данных	Размер
Kod_doka	Идентификатор документа	Автоинкрементное (Autoincrement)	
Naimen	Вид документа	Текстовое(Alpha)	40



Сущность Vid\_obrazov организована в таблицу справочник виды образований (таблица 2.4). Так же как и другие справочники имеет связь один ко многим с той же сущностью Kadry.

Т а б л и ц а 2 . 4 - Справочник Вид образования

Поле	Расшифровка	Тип данных	Размер
Kod_vid	Идентификатор вида образования	Автоинкрементное (Autoincrement)	
Naimen	Вид образования	Текстовое (Alpha)	30

Сущность Vid\_otp организована в таблицу справочник Виды отпусков (таблица 2.5). Имеет связь один ко многим с той же сущностью Kadry.

Т а б л и ц а 2.5-Справочник Вид отпуска

Поле	Расшифровка	Тип данных	Размер
Kod_vid	Идентификатор вида отпуска	Автоинкрементное (Autoincrement)	
Naim	Вид отпуска	Текстовое (Alpha)	20

Таблица справочник должностей (таблица 2.6) соответствует сущности Dolgn имеет так же связь один ко многим с сущностью Kadry.

Т а б л и ц а 2 . 6 - Справочник Должности

Поле	Расшифровка	Тип данных	Размер
Kod_dolgn	Идентификатор должности	Автоинкрементное (Autoincrement)	
Naimen	Должность	Текстовое (Alpha)	20
Obyazan	Должностные обязанности	Текстовое (Memo)	200

Сущность Semya, соответствующая таблице с названием «члены семьи»

(таблица 2.7) имеет так же связь один ко многим с сущностью kadry.

Т а б л и ц а 2 . 7 - Справочник Семья

Поле	Расшифровка	Тип данных	Размер
Kod_sem ya	Идентификатор Записи о сотруднике	Автоинкрементное (Autoincrement)	
FIO	Фамилия Имя Отчество члена семьи	Текстовое(Alpha)	60
Rodstvo	Степень родства члена семьи	Текстовое (Alpha)	10
God_rozh d	Дата Рождения	Дата (Date)	

Информация о приеме сотрудника на работу содержится в таблице «Priyom» (таблица 2.8).

Т а б л и ц а 2 . 8 - Справочник Прием на работу

Поле	Расшифровка	Тип данных	Размер
Kod	Идентификатор Записи о сотруднике	Целое (Short)	
Dat	Дата приема сотрудника на работу	Дата (Date)	
Osn ovanie	Основание	Текстовое (Alpha)	30
Okl ad	Оклад работника	Текстовое (Alpha)	10

Информация об отпуске сотрудника содержится в таблице «Otputsk» (таблица 2.9).

Т а б л и ц а 2 . 9 - Справочник Отпуск

Поле	Расшифровка	Тип данных	Размер

	Kod	Идентификатор Записи о сотруднике	Целое (Short)	
_vid	Kod	Идентификатор вида отпуска	Целое (Short)	Соответс тствует значениям связанной таблицы
od_s	Peri	Период работы с	Дата (Date)	
od_po	Peri	Период работы по	Дата (Date)	
a_nach	Dat	Дата начала отпуска	Дата (Date)	
a_kon	Dat	Дата окончания отпуска	Дата (Date)	
vo	Kol	Количество дней отпуска	Текстовое (Alpha)	10
ov	Osn	Основание	Текстовое (Alpha)	30

Информация об увольнении сотрудников содержится в таблице «Uvoln» (таблица 2.10).

Т а б л и ц а 2 . 1 0 - Справочник Увольнение

Поле	Расшифров ка	Тип данных	Размер
Kod	Идентифик атор записи о сотруднике	Целое (Short)	
Data	Дата увольнения	Дата (Date)	
Osnovani e	Основание	Текстовое (Alpha)	15
Nomer	Номер приказа	Текстовое (Alpha)	10

Сущность Otdel организована в таблицу справочник отделов (таблица 2.11). Имеет связь один ко многим с сущностью priuom.

Т а б л и ц а 2 . 1 1 - Справочник Отделы

Поле	Расшифров ка	Тип данных	Размер
Kod_pod razd	Идентифик атор отдела	Автоинкрем ентное	

		(Autoincrement)	
Naimen	Наименова ние отдела	Текстовое (Alpha)	25

## 3 Реализация программного обеспечения

### 3.1 Пользовательское меню

При запуске исполняющего файла программного продукта пользователю предоставляется возможность самостоятельного использования меню программы: ввод и редактирование, удаление и просмотр записей базы данных, получение необходимой информации посредством ввода данных во входные формы и просмотра выходных форм.

Основное меню программного продукта состоит из следующих вкладок:

- действия;
- справочники;
- отчеты;
- справка.

Пункт меню Сотрудники вызывает главный журнал сотрудников.

Справочники в программе предназначены для ввода и хранения информации, постоянно используемой при вводе исходных данных. Справочники вызываются при вводе в поле, с которым они связаны. В программе используются следующие справочники:

- справочник «отделы»;
- справочник «должности»;
- справочник «вид отпуска»;
- справочник «вид документа»;
- справочник «вид образования».

Во всех локальных справочниках используется порядковая система кодирования. Обосновывается это тем, что набор значений атрибутов очень ограничен и практически не меняется. Порядковый номер присваивается автоинкрементно, т.е. автоматически и последовательно. Информационная база представлена в виде совокупности локальных файлов.

Через пункт меню «Отчеты» доступны стандартные и статистические отчеты.

Стандартные отчеты:

- штатное расписание;
- список сотрудников по отделам;
- список сотрудников с датой приема;
- уволенные по подразделениям;
- количество сотрудников.

Так как данный проект предполагает хранение приказов по сотрудникам, то в приложении должна обеспечиваться генерация печатных форм приказов на основе имеющейся информации в базе данных. Для обеспечения возможности вносить изменения в сгенерированные печатные формы они, при нажатии на форме ввода либо на форме редактирования приказа кнопки «Печать» отправляются в приложение Microsoft Word, где существует больше

возможностей для корректировки содержимого документов. Для каждого приказа существуют шаблоны, в которых поля типа -Имя Поле БД - заполняются соответствующими полями из БД.

По шаблонам формируются следующие документы:

- 1) приказ о приеме на работу;
- 2) трудовой договор;
- 3) приказ об увольнении.

### 3.2 Формы как средство добавления, удаления, просмотра, изменений параметров базы данных

Как правило, работа с любой задачей начинается с заполнения справочников. В дальнейшем по мере работы с программой справочники также пополняются и изменяются. В проекте присутствуют следующие справочники:

Справочник «Должности» (рисунок 3.1).

Код	Наименование
1	директор
2	бухгалтер
3	программист
4	менеджер
5	старший менеджер
6	продавец консультант
7	охранник
8	юрист
9	экономист
10	кассир
11	делопроизводитель

Рисунок 3.1 - Экранная форма справочника «Должности»

Справочник «Подразделения» содержит информацию о подразделениях предприятия (рисунок 3.2).

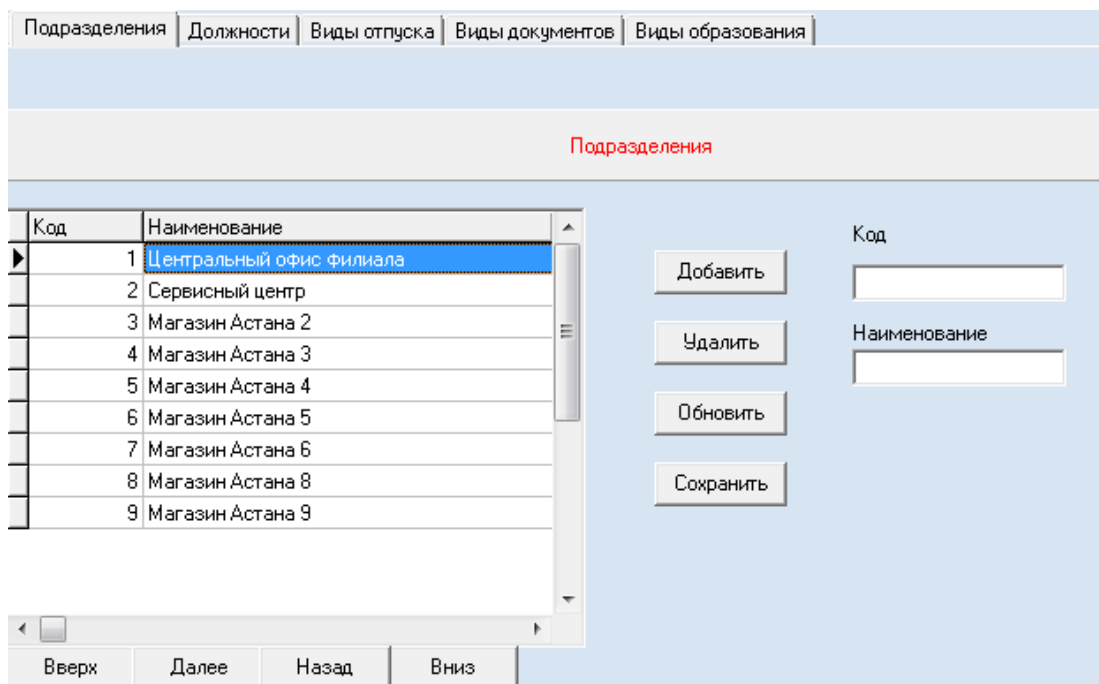


Рисунок 3.2 - Экранная форма справочника «Подразделения»

На рисунке 3.3 приведена экранная форма справочника «Виды образования».

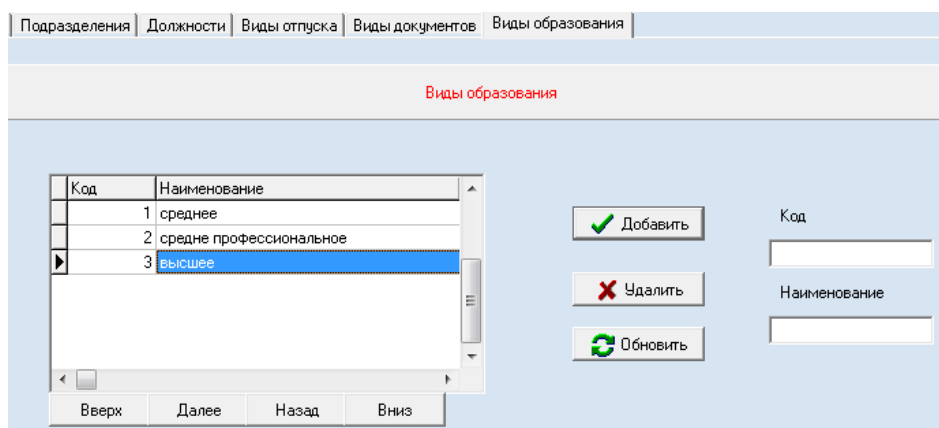


Рисунок 3.3 - Экранная форма справочника «Виды образования»

На рисунке 3.4 приведена экранная форма справочника «Виды отпусков».

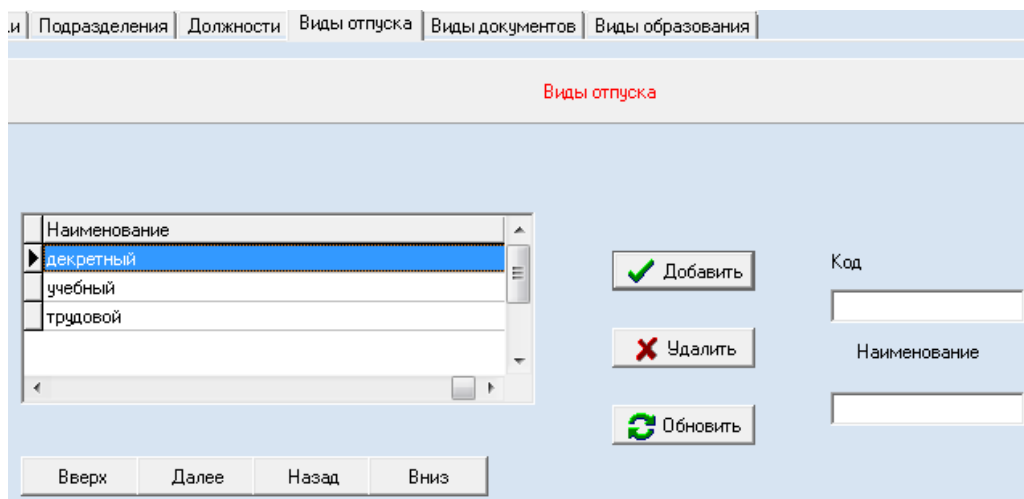


Рисунок 3.4 - Экранная форма справочника «Виды отпусков»

На рисунке 3.5 приведена экранная форма справочника «Виды документов».

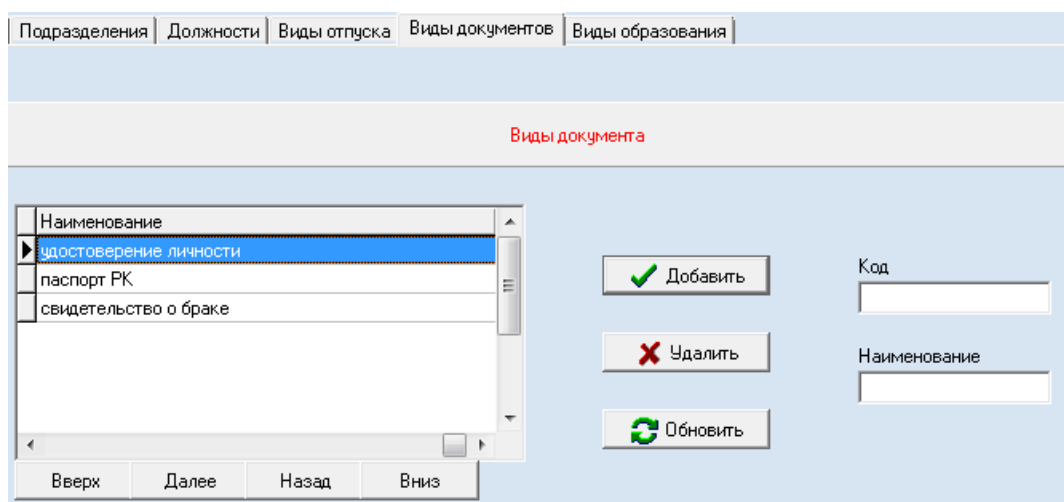


Рисунок 3.5 - Экранная форма справочника «Виды документов»

Приведенные выше формы, кроме перехода на следующую и предыдущую запись, позволят также перейти сразу непосредственно к первой или последней записи, создать новую запись, отредактировать существующую запись, сохранить изменения, удалить запись, а так же выйти из формы.

При выборе ссылки "Сотрудники" на главной форме открывается список действующих сотрудников (рисунок 3.6). На данной странице реализованы функции добавления, редактирования, удаления и просмотра всех данных сотрудника.



Отдел кадров

Действия Справочники Отчеты Справка

Сотрудники Подразделения Должности Виды отпуска Виды документов Виды образования

Сотрудники

Табел	ФИО	Пол	Дата рождения	Место рождения	Адрес	Дом. телефон	Мобильный	Эл. почта	ИНН	Уд. личн. кем
1167	Николаев Борис Анатоль	муж	1975-10-23	Карагандинская обл.	ул. Дружбы д. 73 кв. 27	413964	87024625093	focus@m91023846		МВД РК
1624	Шульженко Ада Вячеслав	жен	1982-12-03	Алматинская обл.	пер. Районный д. 12 кв. 25	301835	87753334934	ada_sh@21203456		МВД РК
1427	Коробейников Сергей Вик	муж	1979-04-05	Карагандинская обл.	ул. Абылай хана д. 48 кв. 11	302009	87013546724	sergey_790403657		МВД РК
1123	Аймужамбетов Нуржан М.	муж	1986-04-17	Восточно-Казах	ул. Лободы д. 4 кв. 37	560691	87714526476	nurzhan-060417563		МВД РК
1127	Тусибиянова Еркежан М.	жен	1988-09-05	Восточно-Казах	5 микр. д. 3 кв. 11	520198	87028364646	erke-511#80905827		МВД РК
1455	Баймурзин Ринат руслан	муж	1985-04-23	Акмолинская обл.	ул. Муканова д. 17 кв. 45	467365	87756241673	rika_84@40294848		МВД РК
2413	Акласов Дастан Ериков	муж	1990-01-23	Акмолинская обл.	ул. Муканова д. 24 кв. 87	462727	87057352783	dosik90@00123948		МВД РК

Добавить <Вверх Далее> <<Назад Вниз>

Рисунок 3.6 - Экранная форма «Сотрудники»

Чтобы занести информацию о новом сотруднике нужно выбрать кнопку «Добавить». Появляется форма ввода информации о новом сотруднике с несколькими закладками.

3.7. Внешний вид экранной формы закладки «Общие» приведен на рисунке

Общие | Документы | Образование | Семейное положение | Прием на работу

Табельный №  ФИО  Пол

Дата рождения  Адрес

Место рождения  Дом. телефон

Электронная почта  Мобильный

Рисунок 3.7 - Экранная форма «Личная карточка» (Закладка «Общие»)

Вносятся текстовые поля «Фамилия, имя, отчество», «Адрес», выбирается пол, дата рождения и другая информация.

Внешний вид экранной формы закладки «Образование» приведен на рисунке 3.8.

Рисунок 3.8 - Экранная форма «Личная карточка» (Закладка «Образование»)

При щелчке на кнопку «Семейное положение» появляется форма с возможными значениями данного поля. В поле «ФИО» вносится соответственно имя, фамилия и отчество члена семьи, степень родства работника с членом семьи, а также год рождения члена семьи. Внешний вид экранной формы «Семейное положение» приведен на рисунке 3.9.

Рисунок 3.9 - Экранная форма «Семейное положение»

На вкладке «Прием на работу» (рисунок 3.10) вносятся сведения о номерах и датах трудового договора, приказов о приеме на работу, переводе и увольнении.

Общие | Документы | Образование | Семейное положение | Прием на работу

Приказ о приеме на работу      Номер: 125с      Дата: 04.11.2013

Должность: старший менеджер      Отдел: отдел продаж

Трудовой договор      Номер: 52      Дата: 04.11.2013

Приказ об увольнении      Номер:      Дата:

Добавить | Удалить | Печать

Рисунок 3.10 - Экранная форма «Личная карточка» (Закладка «Прием на работу»)

При нажатии на кнопку «Печать» происходит формирование соответствующего приказа и трудового договора в виде документа Microsoft Word.

### 3.3 Формирование отчетов

Чтобы сформировать отчеты, нужно выбрать пункт меню «Отчеты», выбрать вид отчета, указать временной интервал и нажать кнопку «Вывести отчет». После просмотра отчета его можно распечатать. На рисунке 3.11 изображено главное окно программы с выпадающим списком формируемых отчетов.

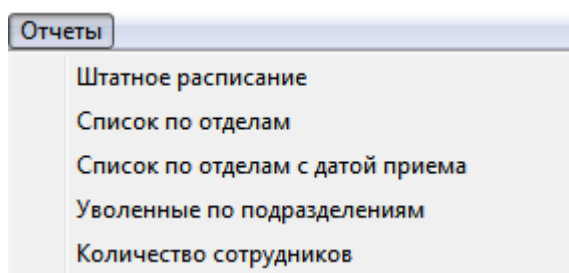


Рисунок 3.11 - Выбор вида отчета

Для формирования списка сотрудников по отделам необходимо выбрать название отдела из ниспадающего списка (рисунок 3.12) и нажать кнопку «Вывести отчет». Сформированный отчет можно экспортировать в Excel для дальнейшего редактирования и вывода на печать (рисунок 3.13).

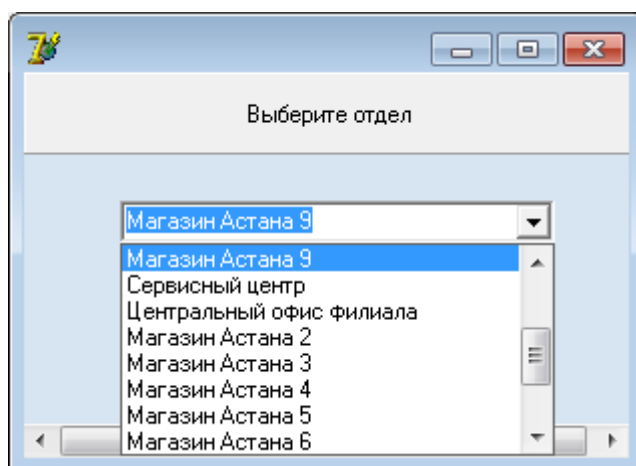


Рисунок 3.12 - Выбор отдела

ФИО сотрудника	Отдел	Должность
Аймухамбетов Нуржан Манарбекович	Магазин Астана 9	менеджер
Тусибиянова Еркежан Манарбековна	Магазин Астана 9	старший менеджер
Баймурзин Ринат русланович	Магазин Астана 9	продавец консультант
Акласов Дастан Ерикович	Магазин Астана 9	менеджер
Ким Иван	Магазин Астана 9	старший менеджер
Пак Юлия	Магазин Астана 9	продавец консультант
Касенова Маржан Сайлауовна	Магазин Астана 9	кассир
Каримов Сырым Асхатович	Магазин Астана 9	бухгалтер
Сатиева Акмарал Сиезовна	Магазин Астана 9	кассир
Уралжанов Азамат Мамырханович	Магазин Астана 9	охранник

Рисунок 3.13 - Внешний вид отчета «Список по отделам»

Для просмотра сведений о датах приема на работу сотрудников и номеров приказов используется отчет «Список по отделам с датой приема». Для формирования данного отчета необходимо выбрать название отдела из ниспадающего списка и нажать кнопку «Вывести отчет». Сформированный отчет можно экспортировать в Excel для дальнейшего редактирования и вывода на печать (рисунок 3.14).

Список по отделам с датой приема

ФИО сотрудника	Отдел	Номер приказа о приеме	Дата приема
Аймұхамбетов Нуржан Манарбекович	Магазин Астана 9	4	2008-02-04
Тусибиянова Еркежан Манарбековна	Магазин Астана 9	5	2010-03-12
Баймурзин Ринат Русланович	Магазин Астана 9	6	2005-02-04
Ақласов Дастан Ерикович	Магазин Астана 9	7	2009-01-13
Ким Иван Анатольевич	Магазин Астана 9	8	2007-09-23
Пак Юлия Вячеславовна	Магазин Астана 9	13	2012-04-23
Касенова Маржан Сайлауовна	Магазин Астана 9	14	2011-03-20
Каримов Сырым Асхатович	Магазин Астана 9	15	2012-04-20
Сатиева Акмарал Сиезовна	Магазин Астана 9	16	2011-03-26
Уралжанов Азамат Мамырханович	Магазин Астана 9	17	2012-06-12

Экспорт в Excel    Зкрыть

Рисунок 3.14 - Внешний вид отчета «Список по отделам с датой приема»

На рисунке 3.15 приведен внешний вид формы для просмотра сведений об уволенных сотрудниках.

Уволенные по подразделениям

ФИО сотрудника	Отдел	Номер приказа об увольнении	Дата увольнения
Шульженко Ада Вячеславовна	Магазин Астана 2	4	2013-09-20

Экспорт в Excel    Зкрыть

Рисунок 3.15 - Внешний вид отчета «Список уволенных сотрудников по подразделениям»

При выборе пункта «Штатное расписание» появляется форма, позволяющая добавлять, удалять, редактировать позиции штатного расписания. Возможен вывод на печать и экспорт в Microsoft Excel (рисунки 3.16, 3.17).

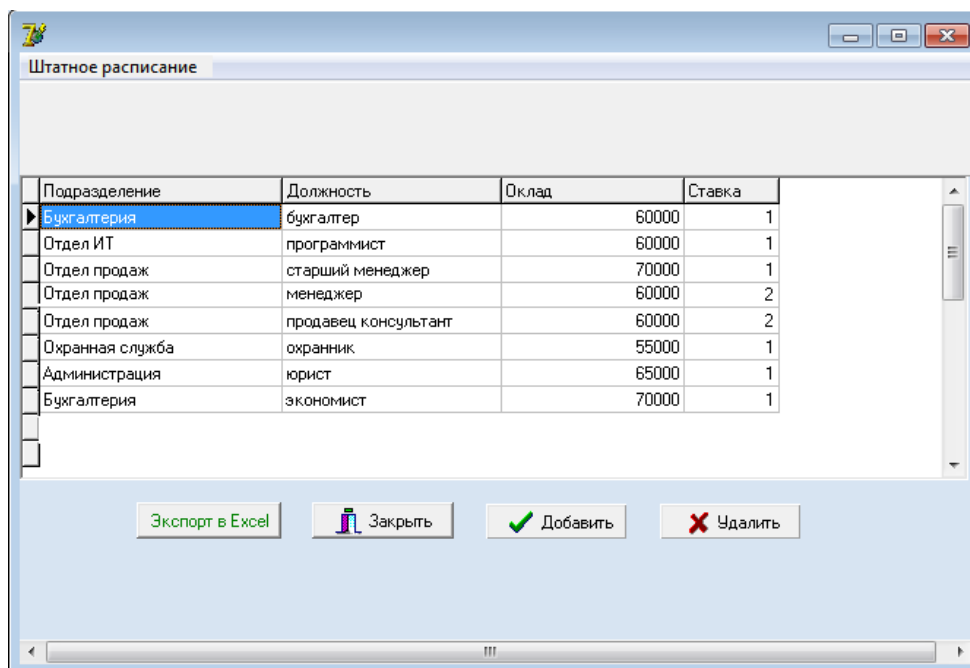


Рисунок 3.16 - Экранная форма «Штатное расписание»

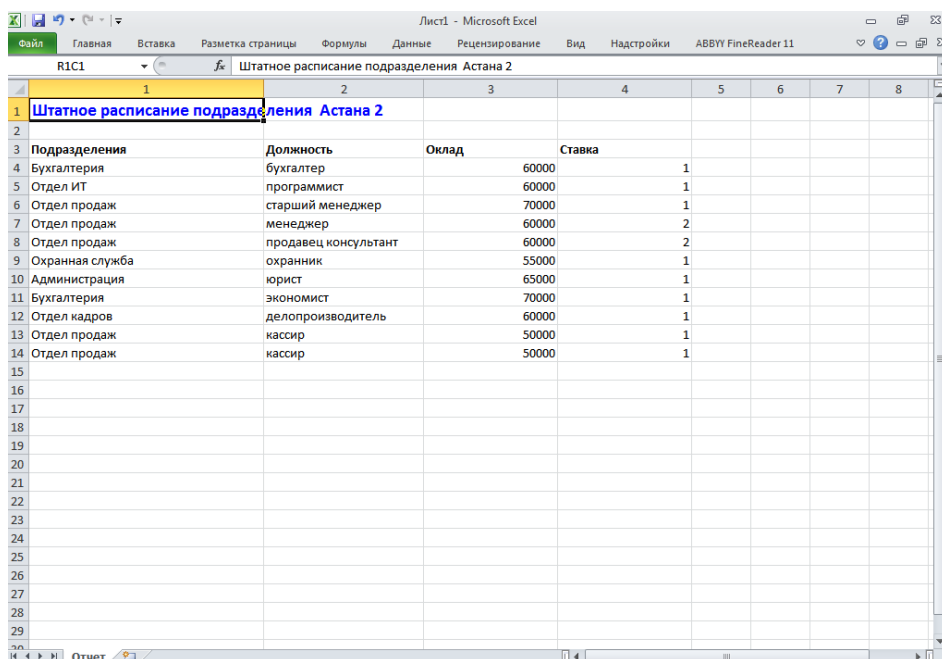


Рисунок 3.17 - Экспорт в MS Excel

На рисунках 3.18 и 3.19 изображен экспорт в MS Word приказа о приеме на работу и трудового договора.

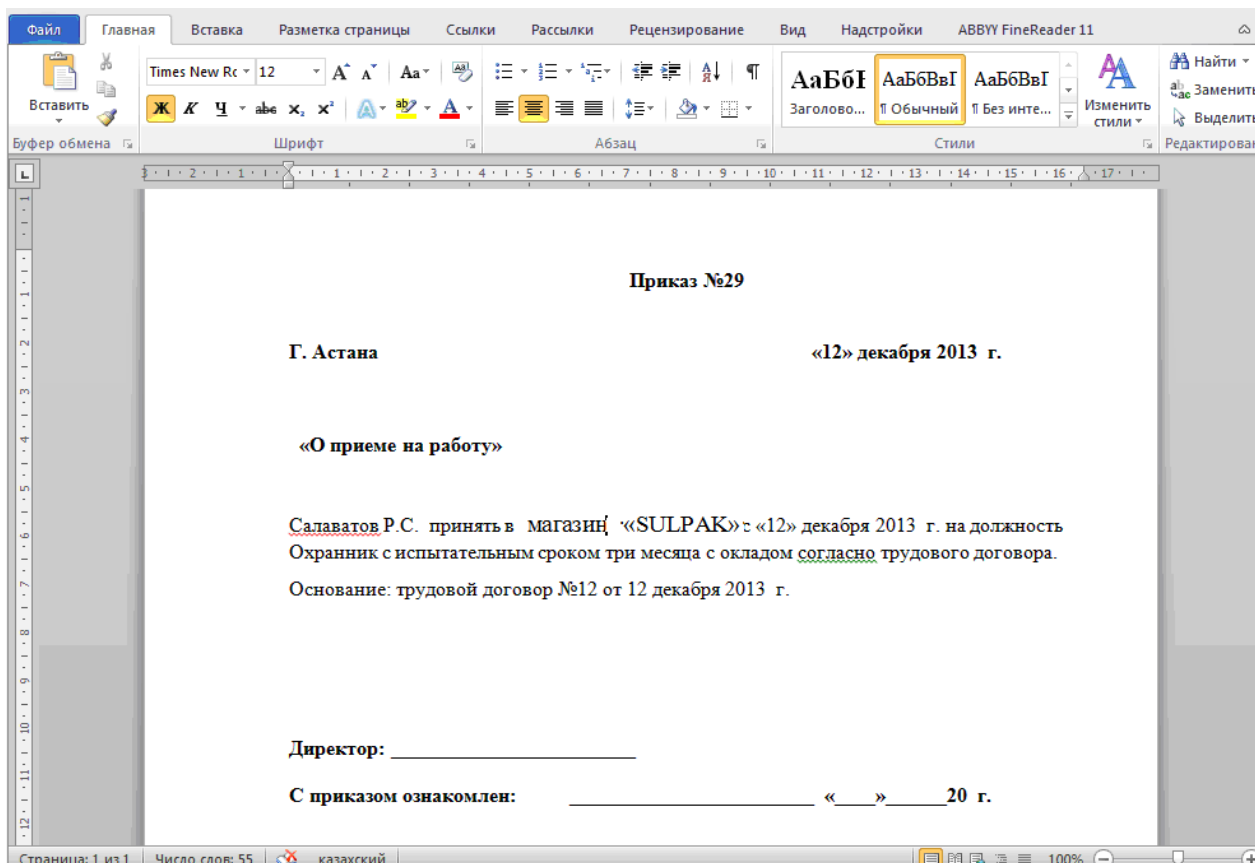


Рисунок 3.18 - Экспорт в MS Word приказа о приеме на работу

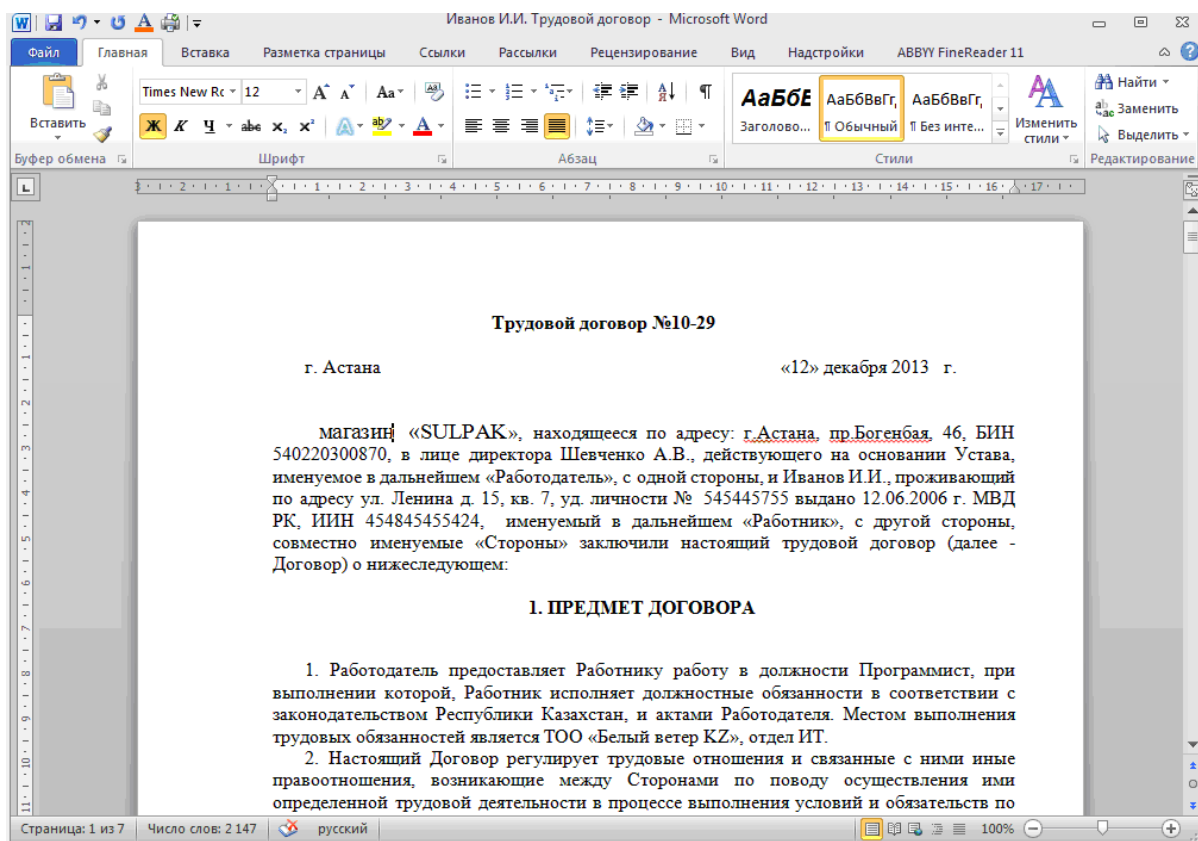


Рисунок 3.19 - Экспорт в MS Word трудового договора

#### 4 Технико-экономическое обоснование

В современных условиях развития отечественных предприятий применение автоматизации на всех уровнях управления и производства имеет огромное значение, которое в первую очередь является неотъемлемым признаком развития предприятий.

При использовании традиционных расчетных методов на этапе проектирования время, необходимое для расчета, оказывается соизмеримым со временем проектирования всей конструкции в целом. Если к тому же на начальном этапе проектирования требуется сравнить несколько вариантов схем в целях выбора оптимальной, то необходимость применения программных комплексов для расчета на ЭВМ становится очевидной [16].

Одной из актуальных сегодня задач является автоматизация производства. Автоматизация проникает во все сферы деятельности предприятия. В том числе и в такую область работы, как процесс управления документацией предприятия

Преимуществами автоматизированной системы перед традиционным методом управления документооборотом является:

- сокращения количества бумаги;
- расходных материалов для принтеров, копировально-множительной техники;
- уменьшения площадей, отводимых для хранения документов;
- уменьшение простоев при обработке документов и времени их поиска.
- улучшение качества работы организации.

Целью технико-экономического обоснования разработки данного программного комплекса является проведение качественного и количественного анализа экономической целесообразности внедрения системы, а также определение организационно-экономических условий ее эффективного функционирования.

Расчет себестоимости разработки программного комплекса

Себестоимость создания автоматизированной системы определяется по формуле:

$$C_{\text{созд}} = M_3 + \Phi_{\text{от}} + Z_{\text{эл}} + A_{\text{ктс}} + \text{НР}, \quad (4.1)$$

где  $M_3$  - материальные затраты;

$\Phi_{\text{от}}$  - фонд оплаты труда;

$Z_{\text{эл}}$  - расходы на электроэнергию, отопление, освещение;

$A_{\text{ктс}}$  - амортизационные отчисления на покупное оборудование и программное обеспечение;

НР - накладные расходы.

Все величины измеряются в тенге.

Расчет материальных затрат

Материальные затраты рассчитываются по формуле:



$$M_3 = Z_6 + Z_T + Z_{др}, \quad (4.2)$$

где  $Z_6$  – затраты на покупку бумаги,  
 $Z_T$  – затраты на покупку тонера для принтера;  
 $Z_{др}$  – расходы на другие канц. товары  
 Затраты на приобретение бумаги вычисляются по формуле:

$$Z_6 = C_6 * K_6, \quad (4.3)$$

где  $C_6$  – стоимость 1-го листа бумаги, тенге;  
 $K_6$  – количество листов бумаги, листов;  
 Принимает  $C_6$  равной 1 тенге,  $K_6 = 500$  листов.  
 Таким образом, согласно формуле 3

$$Z_6 = 1 * 500 = 500 \text{ тенге}$$

Затраты на покупку тонера определяются по формуле:

$$Z_k = C_T * K_T, \quad (4.4)$$

где  $C_T$  – стоимость 1-ой заправки принтера тонером, тенге;  
 $K_T$  – количество заправок в процессе разработки, заправка;  
 Принимает  $C_T$  равной 800 тенге (принтер Samsung ML-1250),  $K_T = 1$   
 (одной заправки хватает для печати 2000 страниц, что в нашем случае является достаточным количеством.

Таким образом, по формуле (4.4),

$$Z_k = 1 * 800 = 800 \text{ тенге}$$

Затраты на закупку других канцелярских товаров (дискеты, папки, ручки и т.д.) принимаем равной  $Z_{др} = 1500$  тенге

В конечном итоге сумма вычисляется по формуле (4.2) и составляет:

$$M_3 = 500 + 800 + 1500 = 2800 \text{ тенге}$$

Расчет фонда оплаты труда

Фонд оплаты труда рассчитываем по формуле:

$$\Phi_{от} = 1,21 * Z_{п} * T_3 * n, \quad (4.5)$$

где  $Z_{п}$  - ежемесячная заработная плата разработчика с учетом социального налога (21%);

$T_3$  - время на разработку задачи, месяц;

n – количество разработчиков.

По формуле (5) получим

$$\Phi_{от} = 1,21 * 20000 * 2 * 1 = 48400 \text{ тенге}$$

Расчет затрат на электроэнергию, освещение, отопление

Затраты рассчитываем по формуле

$$З_{эл} = C_{эл} + C_{ос} + C_{от}, \quad (4.6)$$

где  $C_{эл}$  – затраты на электроэнергию, потребляемую фиксированными активами;

$C_{ос}$  – затраты на освещение;

$C_{от}$  – затраты на отопление.

Затраты на электроэнергию можно определить по формуле

$$C_{эл} = [M_k * t_k + M_p * t_p] * T_3 * c, \quad (4.7)$$

где  $M_k$  - потребляемая мощность компьютера, кВт;

$t_k$  - время работы компьютера в день, час;

$M_p$  – потребляемая мощность принтера, кВт;

$t_p$  – время работы принтера в день, час;

$T_3$  – количество рабочих дней, затраченных на разработку, день;

$c$  – стоимость 1 кВт\*ч, тенге.

$M_k$  принимаем равной 0,45 кВт/час

$t_k$  принимаем равным продолжительности рабочего дня – 8 часов

$M_p$  по техническим данным составляет 0,06 кВт/час.

$t_p$  – 2 часа в день

$T_3 = 45$ . Т.к. на разработку системы необходимо 2 месяца, то количество дней, с учетом выходных, примерно составит 45.

Стоимость одного кВт\*ч составляет 7,13 тенге.

Вычисляем по формуле (4.7) и получаем сумму затрат на электроэнергию:

$$C_{эл} = [0,45 * 8 + 0,06 * 2] * 45 * 7,13 \approx 2093 \text{ тенге}$$

Затраты на освещение вычисляются по формуле:

$$C_{ос} = C_{лх} * t_n * c * T_3, \quad (4.8)$$

где  $C_{лх}$  - мощность светустановки, кВт;

$t$  - количество часов освещения в день, час;

$c$  - стоимость 1 кВт\*ч, тенге;

$T_3$  - количество рабочих дней, затраченных на создание системы, день.

Принимаем следующие значения

$C_{л} = 0,25$  кВт;

$t = 4$  часа, т.к. разработка производится в зимнее время;

$c = 12,5$  тенге;

$T_3 = 45$

Затраты на освещение составят

$$C_{oc} = 0,25 * 4 * 12,5 * 45 = 563 \text{ тенге}$$

Затраты на отопление вычисляются по формуле

$$C_{от} = C_{от} * P_{л} * M_{от}, \quad (4.9)$$

где  $C_{от}$  – цена отопления за 1 м<sup>2</sup>, тенге.;

$P_{л}$  - площадь рабочего помещения, м<sup>2</sup>;

$M_{от}$  – количество отопительных месяцев совпадающих с проведением работ.

Принимаем следующие значения

$C_{от} = 84,8$  тенге/м<sup>2</sup>

$P_{л} = 29,2$  м<sup>2</sup>

$M_{от} = 2$  месяца, т.к. время разработки приходится на зимний период

Затраты на отопление составят:

$$C_{от} = 84,8 * 29,2 * 2 = 4952 \text{ тенге}$$

Общие затраты на электроэнергию, освещение, отопление вычисляем по формуле (4.6) и они составят:

$$Z_{эл} = 2093 + 563 + 4952 = 7608 \text{ тенге}$$

Расчет затрат на амортизационные отчисления

Затраты на амортизационные отчисления определяются по формуле

$$A_{кТС} = ((C_{фа} * N_a) / 12) * T_3, \quad (4.10)$$

где  $C_{фа}$  - стоимость фиксированных активов, тенге;

$N_a$  – норма годовых амортизационных отчислений фиксированных активов;

$T_3$  - количество рабочих дней, затраченных на создание системы, месяц.

Стоимость фиксированных активов вычисляется по формуле

$$C_{\text{фа}} = C_{\text{комп}} + C_{\text{прин}} + C_{\text{по}} * n \quad (4.11)$$

где  $C_{\text{комп}}$  – стоимость компьютера, тенге;

$C_{\text{прин}}$  – стоимость принтера, тенге;

$C_{\text{по}}$  – стоимость пакета программного обеспечения;

$n$  – количество компьютеров;

Принимаем следующие значения:

$C_{\text{комп}} = 70\,000$  тенге;

$C_{\text{прин}} = 30\,000$  тенге (принтер Samsung ML-1250);

$C_{\text{по}} = 63\,000$  тенге на 1 компьютер

$n = 1$ , т.к. для разработки автоматизированной системы достаточно установить пакет программного обеспечения на 1 компьютер

$H_a = 0,25$ ;

$T_3 = 2$  месяца.

Стоимость фиксированных активов составляет

$$C_{\text{фа}} = 30\,000 + 70\,000 + 63\,000 \times 1 = 163\,000 \text{ тенге}$$

Затраты на амортизационные отчисления составят

$$A_{\text{ктс}} = ((163\,000 \times 0,25)/12) \times 2 = 6792 \text{ тенге}$$

Накладные – это расходы, связанные с организацией производства и содержанием производственного оборудования. Они рассчитываются по формуле

$$НР = K_3 \times C_{\text{фа}}, \quad (4.12)$$

где  $K_3 = 0,05$  – условный коэффициент затрат на накладные расходы от стоимости фиксированных активов.

$C_{\text{фа}}$  – стоимость фиксированных активов.

Накладные расходы составят

$$НР = 0,05 \times 163\,000 = 8150 \text{ тенге}$$

Вычислим по формуле (4.1), себестоимость создания автоматизированной системы

$$C_{\text{созд}} = 2800 + 48400 + 7608 + 6792 + 8150 = 73750 \text{ тенге}$$

Сводные данные по капитальным затратам создания программного комплекса представлены в таблице 4.1.

Т а б л и ц а 4 . 1 - Капитальные вложения

Наименование	Сокращенное обозначение	Сумма
Материальные затраты	$M_3$	2800 тенге
Фонд оплаты труда	$\Phi_{от}$	48400 тенге
Затраты на электроэнергию, освещение, отопление	$Z_{эл}$	7608 тенге
Затраты на амортизационные отчисления	$A_{кгс}$	6792 тенге
Накладные расходы	НР	8150 тенге
Себестоимость создания ПО	$C_{созд}$	73750 тенге

Внедрение программного обеспечения для учета на предприятии отражается в первую очередь на потреблении «расходных» материалов (бумага, тонер для принтера, копировально-множительного аппарата, канц.товары). Произведем расчет материальных затрат в течение отчетного периода (1 года) по двум методам управления документацией на предприятии.

$$M_3 = Z_6 + Z_п + Z_к + Z_{др} \quad (4.13)$$

где  $Z_6$  – затраты на покупку бумаги,  
 $Z_п$  – затраты на покупку тонера для принтера;  
 $Z_к$  – затраты на заправку картриджа для ксерокса;  
 $Z_{др}$  – расходы на другие канц.товары

Расчет материальных затрат (обычный способ работы):

$$Z_6 = 1 * (3000 + 4000 + 1000 + 250 + 250 + 120 + 960 + 50 + 100 + 2000) = 11730 \text{ тенге}$$

$$Z_п = (800 * 4 + 2000) = 3200 + 2000 = 5200 \text{ тенге}$$

$$Z_к = 15000 * 2 + 3000 * 6 = 30000 + 18000 = 48000 \text{ тенге}$$

$$Z_{др} = 9000 \text{ тенге}$$

Суммарный объем материальных затрат по формуле (4.13) составит:

$$M_3 = 11730 + 5200 + 48000 + 9000 = 73930 \text{ тенге}$$

Расчет материальных затрат (АСУД)

$$Z_6 = 1 * (1500 + 1000 + 500 + 100 + 100 + 60 + 192 + 0 + 50 + 144) = 3646 \text{ тенге}$$

$$Z_п = 800 * 4 + 2000 = 5200 \text{ тенге}$$

$$Z_к = 15000 + 3000 * 3 = 24000 \text{ тенге}$$

$$Z_{др} = 6000 \text{ тенге}$$

Суммарный объем материальных затрат по формуле (4.13) составит

$$M_з = 3646 + 5200 + 24000 + 6000 = 38846 \text{ тенге}$$

Данные сравнения материальных затрат:

- традиционный метод – 73930 тенге;

- автоматизированный метод – 38846 тенге

Экономия по виду затрат – 35084 тенге в год, то есть затраты компании уменьшились в 2 раза.

Итак, создание и внедрение ПО учета кадрового состава предприятия с точки зрения экономики позволяет:

повысить эффективность работы персонала с поступающим потоком ежедневной информации, автоматизировать документооборот предприятия;

увеличить производительность труда за счет экономии времени на регистрацию, оформление, поиск и анализ информации;

улучшение значений показателей качества обработки информации (повышение степени достоверности обработки информации, степени ее защищенности, повышение степени автоматизации получения первичной информации).

## **5 Безопасность жизнедеятельности**

### **5.1 Анализ опасных и вредных факторов, воздействующих на программиста**

Эргономика - наука, изучающая функциональные возможности человека в трудовых процессах с целью создания для него оптимальных условий труда, т.е. таких условий, которые делают труд высоко производительным и надежным, в то же время обеспечивают человеку необходимые удобства и сохраняют его силы, здоровье и работоспособность. Эргономика изучает проблемы оптимального распределения и согласования функций между человеком и машиной, проектирует процесс деятельности человека, обосновывает оптимальные требования к средствам и условиям деятельности и разрабатывает методы их учета при создании и эксплуатации техники, управляемой и обслуживаемой человеком.

Работа с ЭВМ в РК регламентируется санитарными правилами и нормы СанПиН N 1.01.004.01 «Гигиенические требования к организации и условиям работы с видеодисплейными терминалами и персональными электронно-вычислительными машинами» [20].

В дипломном проекте разрабатывается программное обеспечение учета кадрового состава предприятия.

Рабочий кабинет, располагается на втором этаже здания фирмы. Длина комнаты – 8 м., ширина – 7 м., высота - 3,5 м. Естественное освещение кабинета - боковое, осуществляется посредством двух окон размерами каждого 1,7 x 2,4 м. Дверь – деревянная, двухстворчатая, покрытая лаком коричнево-красного цвета. Высота двери – 2 м., ширина каждой створки двери - 0,7 м.

Площадь на одно рабочее место с ВДТ и ПЭВМ в офисах, административно-производственных помещениях и других учреждениях при периметральном расположении должна быть не менее 4,0 кв.м. (п. 3.4 [20]). Для данного помещения норма выполняется.

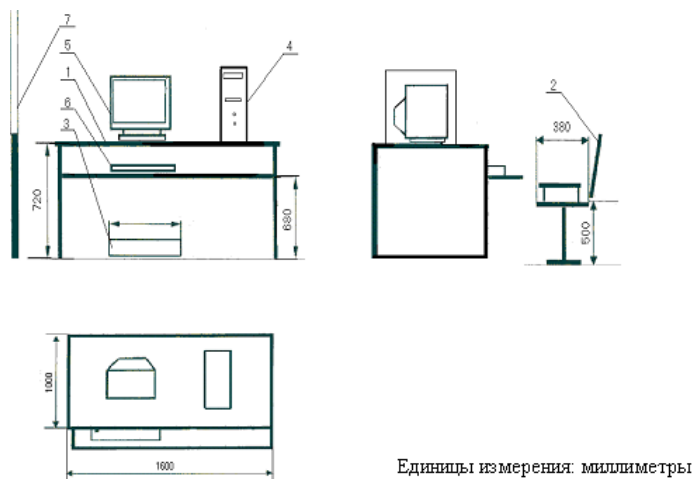
Помещение оборудовано 3 персональными компьютерами с цветными жидкокристаллическими мониторами.

В соответствии с санитарными правилами и нормами дизайн должен предусматривать окраску корпуса ПЭВМ в спокойные мягкие тона с диффузным рассеиванием света, одноцветность окраски всех блоков, матовую их поверхность с коэффициентом отражения 0.4-0.6 (В соответствии с п. 2.1 [20]). В помещении установлены компьютеры черного и серого цвета, поверхность корпуса матовая, не имеет бликующих деталей.

Оконный проем в помещении оборудован занавесями (тюль) для исключения попадания прямых солнечных лучей и регулировки уровней освещенности на рабочих местах. Расстояние между боковыми поверхностями мониторов составляет 1,2 м., что соответствует требованиям санитарных норм и правил. Расстояние между стеной с оконными проемами и столами составляет

0,3 м. Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и особенностей (размер монитора, процессорного блока, клавиатуры и т.п.), характера выполняемой работы, соответствовать эргономическим требованиям. [20]

Схема рабочего места представлена на рисунке 5.1.



1- стол; 2 – стул; 3 – подставка для ног; 4 - системный блок; 5 - монитор; 6 – клавиатура; 7 – окно.

Рисунок 5.1 – Схема рабочего места

Необходимо оборудовать рабочие места компьютерными столами, соответствующим требованиям [20]. Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и особенностей (размер монитора, процессорного блока, клавиатуры и т.п.), характера выполняемой работы, соответствовать эргономическим требованиям. Высота рабочей поверхности стола должна регулироваться в пределах 68-80 см; при отсутствии такой возможности высота рабочей поверхности стола должна составлять не менее 72,5 см. ([18]). Рабочий стол должен иметь пространство для ног высотой не менее 60 см, шириной - не менее 50 см глубиной на уровне колес - не менее 45 см и на уровне вытянутых ног - не менее 65см ([20]).

Рабочее место рекомендуется оборудовать подставкой для ног. Рабочие места необходимо оборудовать стульями, соответствующими эргономическим требованиям СанПиН. Рабочий стул (кресло) должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки.

### 5.1.1 Мероприятия по снижению опасных и вредных факторов

Согласно пункту 4.1.9 [20], санитарно-химические исследования в



натурных условиях (в жилых, учебных, административных и общественных помещениях) должны проводиться при осуществлении текущего санитарного надзора за состоянием воздушной среды и условиями применения ВДТ, ПЭВМ, ПК и ЭС в процессе их эксплуатации.

В среднестатистическом 17-дюймовом ЭЛТ - мониторе содержится 1-2 килограмма свинца, большая часть которого сосредоточена в трубке. Свинец при воздействии на организм человека поражает человеческие почки, кровь, центральную нервную систему и вызывает самые разнообразные врождённые уродства.

Хром, выступающий во многих случаях в качестве защиты от коррозии, способен вызывать целый спектр аллергических реакций и даже приводить к повреждению ДНК.

Аббревиатура TCO, за которой скрывается Шведская федерация профсоюзов, известна сегодня во всём мире, поскольку именно эта организация за последние 15 лет разработала массу различных сертификационных требований к эргономике и экологической чистоте электроники и прочей высокотехнологичной техники. Нередко требования TCO оказываются строже национальных стандартов, поэтому крупные производители стараются получить именно сертификацию TCO, которая гарантирует соответствие всем самым высоким требованиям [21].

Самые "зелёные" системные платы выпускает, вероятно, компания Fujitsu-Siemens. По крайней мере, они при всяком удобном случае подчёркивают экологическую чистоту технологического процесса. Fujitsu-Siemens даже выпускает специальную серию плат, не содержащих свинца и галогена, причём эти элементы не применяются даже в процессе производства, в частности, при пайке. Системные платы Intel хорошо знакомы любителям максимально надёжных ПК, при этом на большей части современных моделей можно прочесть слова "Lead Free", то есть "не содержит свинца".

Что касается центральных процессоров, то сегодня развернулась настоящая борьба за снижение энергопотребления, поэтому и AMD, и Intel, и VIA, и Transmeta предлагают продукцию с максимально низким потреблением энергии, которого им только удалось добиться.

При производстве оперативной памяти используются те же технологии, что и при выпуске процессоров или системных плат, поэтому с "экологичностью" этих комплектующих всё в полном порядке.

Набирают популярность компьютерные блоки питания с системой активной коррекцией коэффициента мощности - она обозначается как A-PFC (Active Power Factor Correction). Эта система снижает потери энергии за счёт устранения перекаса фаз и учёта реактивного тока, повышает энергоёмкость блока питания и стабилизирует выходное напряжение. [21].

Жёсткие диски всех ведущих производителей, в принципе, мало отличаются друг от друга, при этом порядка 80 процентов компонентов винчестеров, поддаются вторичной переработке. Последние модели любых марок соответствуют требованиям RoHS и отличаются низким

энергопотреблением - порядка 12-14 Вт в режиме поиска и около 7-8 Вт в режиме ожидания.

Повсеместный переход на использование жидкокристаллических мониторов сделал эти устройства менее вредными для окружающей среды, хотя при нагреве различные типы пластиков, применяемых для производства корпусов, могут испускать вредные вещества. Энергопотребление современного ЖК - монитора примерно втрое ниже, чем у ЭЛТ - дисплея, так что в данном случае прогресс идёт рука об руку с улучшением экологических характеристики устройств.

Электромагнитные излучения (ЭМИ) искусственных источников («электромагнитное загрязнение») имеют ряд особенностей:

- возможно их воздействие на большие контингенты людей, включая детей, стариков и больных;
- вполне вероятно долговременное воздействие ЭМП (круглосуточно и даже на протяжении ряда лет);
- практически невозможна «очистка» эфира от нежелательных ЭМИ;
- учитывая специфику работы искусственных источников ЭМИ, практически невозможно ограничение их «выброса».

Под электромагнитным загрязнением среды понимается состояние электромагнитной обстановки, характеризуемое наличием в атмосфере ЭМП повышенной интенсивности, создаваемых техногенными и природными источниками излучения неионизирующей части электромагнитного спектра [22]. Указанный спектр приведен на рисунке 5.2



Рисунок 5.2 – Электромагнитный спектр

## 5.2 Микроклимат и вентиляция

В таблице 5.1 указаны нормы микроклимата для помещений с ПЭВМ

Т а б л и ц а 5 . 1 -Нормы микроклимата

Период	Категория	Температура	Относительная влажность	Скорость движения
Года	работ	воздуха, гр.с, не более	воздуха	воздуха, м/с

Хол одный	Лег кая 1а Легкая 16	22- 24 23-21	40-60 40-60	0, 1 0,1
Теп лый	Лег кая 1а Легкая 16	23- 25 22-24	40-60 40-60	0,1 0,2

В помещении обеспечиваются нормальные условия микроклимата. Увлажнители воздуха не используются.

Постоянство температуры в помещении в зимний период обеспечивается центральной системой отопления. В кабинете установлен чугунный радиатор, состоящий из 14 секций. Однако радиаторы не закрыты деревянными решетками. Рекомендуется закрыть радиаторы деревянными решетками, подходящими по цветовому решению общему интерьеру кабинета.

Кондиционирование воздуха – это создание и автоматическое поддержание в помещениях независимо от наружных условий постоянных или изменяющихся по определенной программе температуры, влажности, чистоты и скорость движения воздуха, наиболее благоприятных для людей или требуемых для нормального протекания технологического процесса. Система кондиционирования в рассматриваемом помещении отсутствует. Обеспечивается естественное проветривание помещений. Проветривание осуществляется перед началом работы, в обеденное время и после окончания работы.

### 5.2.1 Освещение

Одним из основных вопросов охраны труда является организация рационального освещения производственных помещений и рабочих мест.

Правильно спроектированное и выполненное освещение обеспечивает высокий уровень работоспособности, оказывает положительное психологическое воздействие на работающих, способствует повышению производительности труда.

Для освещения помещения, в котором работает оператор, используется смешанное освещение, т.е. сочетание естественного и искусственного освещения.

Естественное освещение – осуществляется через окна в наружных стенах здания. Для регулирования уровня естественной освещенности на окнах установлены жалюзи, что удовлетворяет требованиям пункта о санитарных нормах. Искусственное освещение – используется при недостаточном естественном освещении и осуществляется с помощью двух систем: общего и местного освещения. Общим называют освещение, светильники которого освещают всю площадь помещения. Местным называют освещение, предназначенное для определённого рабочего места.

В таблице 5.2 указаны нормы освещенности рабочего места, согласно принятым санитарным нормам.

Т а б л и ц а 5 . 2 - Уровни освещенности на рабочих местах

Точка контроля	Освещенность, {естеств., искусств), лк	Освещенность, КЕО, %	Освещенность комбинир., лк
Стол, клавиатура	400	1,5	500
Экран	300		300

Для искусственного освещения следует использовать главным образом люминесцентные лампы, у которых высокая световая отдача (до 75 лм/Вт и более), продолжительный срок службы (до 10000 ч), малая яркость светящейся поверхности.

Для исключения засветки экрана дисплея прямыми световыми потоками светильники общего освещения располагают сбоку от рабочего места, параллельно линии зрения оператора и стене с окнами. Такое размещение светильников позволяет производить их последовательное включение в зависимости от величины естественной освещенности и исключает раздражение глаз чередующимися полосами света и тени, возникающее при поперечном расположении светильников.

Искусственное освещение осуществляется двумя потолочными светильниками серии «Амстронг» 418 ARS\R (600x600 мм) с зеркализированными решетками. В каждом светильнике расположены четыре люминесцентные лампы мощностью 18 Вт каждая.

### 5.2.2 Защита от излучений и электромагнитных полей

Электромагнитное излучение (электромагнитные волны) — распространяющееся в пространстве возмущение электрических и магнитных полей [22].

ПЭВМ являются источниками таких излучений как:

- мягкого рентгеновского;
- ультрафиолетового 200-400 нм;
- видимого 400-700 нм;
- ближнего инфракрасного 700-1050 нм;
- радиочастотного 3 кГц-30 МГц;

электростатических полей.

Действие электромагнитного излучения на организм человека в основном определяется поглощенной в нем энергией. Известно, что излучение, попадающее на тело человека, частично отражается и частично поглощается в нем. Поглощенная часть энергии электромагнитного поля превращается в тепловую энергию. Эта часть излучения проходит через кожу и распространяется в организме человека в зависимости от электрических свойств тканей (абсолютной диэлектрической проницаемости, абсолютной магнитной проницаемости, удельной проводимости) и частоты колебаний электромагнитного поля.

При проведении наладочных и профилактических работ, а также в процессе эксплуатации оборудования Вычислительного Центра (ВЦ), человек может прикоснуться к находящимся под напряжением проводниками электрического тока. В этом случае через тело человека будет протекать ток, который может вызвать ожог и другие нарушения организма вплоть до потери сознания

Нормы допустимого облучения устанавливаются для обеспечения безопасных условий труда обслуживающего персонала.

Напряженность электромагнитных полей на рабочем месте не должна превышать:

а) по электрической составляющей:

в диапазоне частот 60 кГц — 3 МГц — 50 В/м;

3 — 30 МГц — 20 В/м;

30—50 МГц — 10 В/м;

50—300 МГц — 5 В/м;

б) по магнитной составляющей:

в диапазоне частот 60 кГц — 1, 5 МГц — 5 А/м;

30 МГц—50 МГц — 0, 3 А/м.

Предельно допустимая плотность потока энергии электромагнитных полей в диапазоне частот 300 МГц — 300 ГГц и время пребывания на рабочем месте и в местах возможного нахождения персонала, связанного профессионально с воздействием полей взаимосвязаны следующим образом: пребывание в течение рабочего дня — до 0, 1 Вт/м<sup>2</sup>; пребывание не более 2ч - 0,1—1 Вт/м<sup>2</sup>, в остальное рабочее время плотность потока энергии не должна превышать 0, 1 Вт/м<sup>2</sup>. [23]

### 5.2.3 Электробезопасность

ЭВМ и большая часть периферийного оборудования являются электрическими установками и работают от сети переменного тока. В случае повреждения проводки или изоляции корпуса различных устройств могут находиться под напряжением. Воздействие тока на человеческое тело может привести к электрической травме. Согласно [23] все помещения,

предназначенные к эксплуатации ВДТ и ЭВМ обязательно должны оснащаться отдельным контуром заземления.

При прикосновении обслуживающего персонала к любому из элементов ЭВМ могут возникать разрядные токи статического электричества. Для снижения величин возникающих зарядов статического электричества в помещениях с установленными ЭВМ покрытие пола следует выполнять из однослойного поливинилхлоридного антистатического линолеума. Аппараты, машины и устройства, являющиеся источниками интенсивного возникновения зарядов статического электричества, следует выделять и заземлять независимо от заземления всей технологической цепи [23]. Рабочее помещение оборудовано заземляющим контуром и пол покрыт антистатическим линолеумом.

Одним из основных мероприятий по обеспечению электробезопасности является защитное заземление всего оборудования ЭВМ. Назначение защитного заземления заключается в том, чтобы создать между корпусом защищаемого устройства и землей электрическое соединение с достаточно малым сопротивлением для того, чтобы в случае замыкания на корпус этого устройства прикосновение к этому корпусу человека (параллельное присоединение) не могло вызвать прохождение через его тело тока такой величины, которая угрожала бы жизни или здоровью. В целях обеспечения надежности заземления требуется периодически проверять сопротивление заземлителя  $R_3$ , которое должно быть не более 10 Ом в сети с напряжением до 1000 В. Также необходимо проводить плановые мероприятия по ремонту и обслуживанию оборудования, в ходе которых следует проверять электробезопасность всех устройств ЭВМ.

Для обеспечения безопасности поражения людей током необходимо обеспечить изоляцию токонесущих частей оборудования, для чего рекомендуется проведение профилактических осмотров кабелей и всей электропроводки. Кроме того, необходимо обеспечить надежное заземление. Эффективным заземлением является заземление трубчатого типа с толщиной стенки 3,5 мм. Длина трубы обычно составляет 250 см, диаметр 5 см. Заземлители располагаются по четырехугольному контуру, с глубиной заложения около 80 см, причем сопротивление заземлителя не должно превышать 10 Ом.

### **5.3 Меры пожарной безопасности**

Пожарная безопасность предусматривает комплекс организационных и технических мероприятий, направленных на обеспечение безопасности людей, предотвращения пожара, ограничение его распространения, а также создание условия для успешного тушения пожара. Пожарная безопасность объединяет мероприятия, осуществляемые в процессе проектирования, строительства, эксплуатации предприятия. Пожар на предприятии наносит большой

материальный ущерб и часто сопровождается несчастными случаями с людьми. Основными причинами воспламенения материалов и возникновения пожаров является неисправность электрооборудования, освещения, неправильная их эксплуатация, наличие статического электричества, неосторожное обращение с огнем, неудовлетворительный надзор за пожарными устройствами и производственным оборудованием.

В соответствии со статьей 15 Закона Республики Казахстан «О пожарной безопасности» разработка и реализация мер пожарной безопасности для предприятий, зданий, сооружений и других объектов, в том числе при их проектировании, должны в обязательном порядке предусматривать решения, обеспечивающие безопасную эвакуацию людей при пожарах.[24]

Здание, в котором расположено рассматриваемое помещение, относится к классу Ф 4.2 функциональной пожарной опасности.

В комнате оператора необходимо расположить ручной углекислотный огнетушитель типа ОУ (ОУ-2, ОУ-5, ОУ-8). Он представляет собой стальной баллон с запорным вентилем. Баллон заполнен сжиженным диоксидом углерода (углекислотой) под давлением 7 Мпа. При открывании вентиля сжиженная углекислота выбрасывается из баллона по сифонной трубке, испаряется, сильно охлаждается в раструбе - снегообразователе и поступает наружу в виде хлопьев снега. Длина струи 2,3 метра, время действия 30-40 секунд. Допускается применение в электроустановках, находящихся под напряжением.

Во всех служебных помещениях обязательно должен быть «План эвакуации людей при пожаре», регламентирующий действия персонала в случае возникновения очага возгорания и указывающий места расположения пожарной техники. Коридоры и проходы, предназначенные для эвакуации, должны иметь, возможно меньшую длину и минимальное количество поворотов. На всем протяжении прохода не должно быть порогов или промежуточных ступеней. Лестницы должны иметь надежные ограждения.

Целесообразно поручень при спуске по лестнице располагать с правой стороны. Дверные проемы на путях эвакуации следует располагать по оси прохода или лестничной клетки. Все виды путей эвакуации должны иметь естественное или искусственное освещение, работающее как, от обычной электросети, так и от сети аварийного освещения. Важную роль в обеспечении безопасного выхода людей играет противодымная защита эвакуационных путей. Лестницы, как правило, размещают у наружных стен с обязательным устройством оконных проемов, которые выполняют роль дымовых люков и обеспечивают лучшую ориентировку эвакуирующихся при движении. План эвакуации представлен на рисунке 5.3.

В рассматриваемом помещении горению подвержены: пол и напольное покрытие (линолеум), занавеси, деревянные конструкции окон и дверей, электронное оборудование ЭВМ и дополнительных устройств, проводка, мебель, пластиковые элементы приборов. При этом возможно выделение опасных веществ: оксид углерода, циан водорода, хлористый водород,

акролеин, окислы азота, формальдегид, фенол , различные алифатические и ароматические углеводороды и др.

Комната не оборудована пожарной сигнализацией, первичные средства пожаротушения отсутствуют, в коридоре расположен пожарный кран, оборудованный пожарным рукавом.

Для повышения уровня пожарной безопасности, необходимо:

оснастить помещение пожарной сигнализацией. Рекомендуемый тип: система на основе оптических пожарных извещателей ДИП-3СУ и пульта Сигнал-20П SMD.C-2000;

оборудовать помещение первичными средствами пожаротушения: углекислотным огнетушителем типа ОУ-2 или ОУ-5;

оборудовать помещение автоматической установкой объемного газового пожаротушения: система азотного пожаротушения «Гарсис»;

обеспечивать регулярные проверки пожарной сигнализации, первичных средств пожаротушения. Проводить инструктаж и тренировки по действиям в случае пожара;

не загромождать и не блокировать эвакуационные выходы;

выполнять требования правил технической эксплуатации и правил техники безопасности при эксплуатации электроустановок.

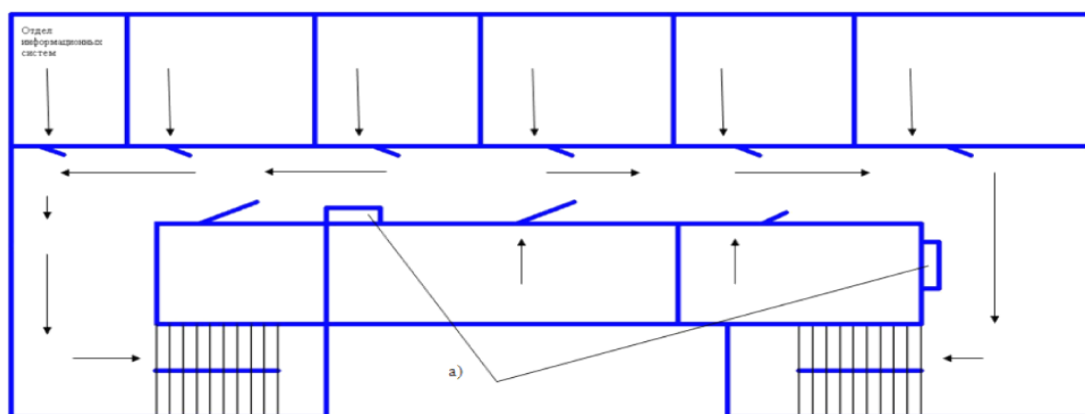


Рисунок 5.3 – План эвакуации в случае возникновения пожара



## **Заключение**

В любой организации, как большой, так и маленькой, возникает проблема такой организации управления данными, которая обеспечила бы наиболее эффективную работу. Ранее организации использовали для этого шкафы с папками, однако теперь предприятия используют компьютеризированные системы автоматизации, позволяющие эффективно хранить, извлекать информацию и управлять большими объемами данных.

В результате проделанной работы было автоматизирован учет кадрового состава магазина «SULPAK»

Реализация данного проекта была проведена без привлечения мощных средств работы с базами данных, которые очень громоздки, поскольку носят универсальный характер и к тому же требуют необходимую базу знаний по теории баз данных.

Разработка программной модели производилась в интегрированной среде визуального программирования Delphi 7.0 на языке программирования Object Pascal.

Использование мощных средств Delphi 7 по созданию приложений работающих в операционной системе Windows и в частности приложений баз данных, позволило создать программный продукт максимально ориентированный на конечного пользователя, который не искушен в вопросах теории баз данных.

Вся необходимая работа по осуществлению методов доступа к информации хранимой в базе данных, её модификации, поддержании базы данных в целостном виде скрыта внутри и пользователю нет необходимости знать о ней, чтобы успешно решать весь круг возникающих задач связанных с использованием информации хранимой базе данных. Более того, программный интерфейс максимально облегчает работу по обращению с базой данных.

Разработанное программное обеспечение позволяет быстро и эффективно производить учет работников на предприятии. Избавляет от рутинных работ, связанных с заполнением данных работника (личные данные, сведения об образовании, паспортные данные, сведения трудовой книжки) в различные книги регистраций. Информационная система обеспечивает однократный ввод сведений, соответствующих личному листку по учёту кадров, которые многократно используются впоследствии в разных документах. Система проста в эксплуатации, удобна и эргономична, не требует особых знаний от пользователя в области программирования и теории баз данных.

## Список использованной литературы

- 1) Гэри Хансен, Джеймс Хансен, «Базы данных разработка и управление» ЗАО «Издательство Бином», 2009г., Перевод с англ. под редакций С. Каратыгина, М.
- 2) Сайт: [www.ci.ru/inform11\\_01/p089pers.htm](http://www.ci.ru/inform11_01/p089pers.htm).
- 3) Сайт: [www.rusnauka.com/14\\_NPRT\\_2011/Economics/10\\_86879.doc.htm](http://www.rusnauka.com/14_NPRT_2011/Economics/10_86879.doc.htm).
- 4) Когловский М.Р. «Технология баз данных на персональных ЭВМ» - М.: Финансы и статистика, 2002г.
- 5) Сайт: [www.wenses.ru/article118.html](http://www.wenses.ru/article118.html).
- 6) Фаронов В.В., Шумаков П.В. Delphi 4. «Руководство разработчика баз данных» - М.: «Нолидж», 2009г.-560с.
- 7) Архангельский А.Я. «Разработка прикладных программ для Windows в Delphi 5» - М.: ЗАО «Издательство БИНОМ», 2009г. – 256 с.
- 8) Дж.Ульман, «Основы систем баз данных», М.: Финансы и статистика, 2003г.
- 9) Дейт К., «Введение в системы баз данных», М.: Наука, 2002г.
- 10) А.Н.Наумов, А.М.Вендров и др., «Системы управления базами данных и знаний», М.: Финансы и статистика, 2001г.
- 11) Брябрин В.М., «Программное обеспечение персональных ЭВМ», М.: Наука, 2009г.
- 12) Дж. Мартин, «Организация баз данных в вычислительных системах» М: Мир 2003г.
- 13) С.М.Диго «Проектирование и использование баз данных» М: Финансы и статистика 2005г.
- 14) Кириллов В.В. «Структурированный язык запросов (SQL)» – СПб.: ИТМО, 2004г. – 80 с.
- 15) Б.Г. Голованов «Введение в программирование в сетях Nowell NetWare» С-П.: Питер, 2000г.
- 16) «Экономика предприятия» Под ред. В.П.Грузинова. -М.: «Банки и биржи» 1998г.
- 17) Шмален Г. «Основы и проблемы экономики предприятия» -М.: Финансы и статистика, 2006г.
- 18) «Экономическая стратегия фирмы». Учебное пособие/колл. авт. под ред. А.П.Градова - М.: Специальная литература, 2005г.
- 19) Трудовой Кодекс Республики Казахстан от от 15 мая 2007 г.
- 20) Санитарные правила и нормы СанПиН N 1.01.004.01. Гигиенические требования к организации и условиям работы с видеодисплейными терминалами и персональными электронно-вычислительными машинами от 17 декабря 2001 г.
- 21) Денисенко Г.Ф. «Охрана труда». Учебное пособие для инженерно-технических вузов. М. Высшая школа. 1990г.
- 22) Гордон Г.Ю., Вайнштейн Л.И. «Электротравматизм и его

- предупреждение». М. Энергоатомиздат. 1986г.
- 23) Закон Республики Казахстан «Об охране окружающей природной среды», Алматы, 1997г.
- 24) Графкина М.В., Перехожих, Мельников. Электромагнитное излучение. 1999г

## Приложение А

```
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Menus, ActnList, StdActns, ShellAPI, ComCtrls;
type
TForm1 = class(TForm) MainMenu1: TMainMenu; PopupMenu1:
TPopupMenu;
  ActionList1: TActionList; Open: TAction; N3: TMenuItem; N4: TMenuItem;
  N5: TMenuItem; N6: TMenuItem; N7: TMenuItem; N12: TMenuItem;
  Passport: TAction; Obrazov: TAction; Rabota: TAction; VSY: TAction;
  Poisc: TAction; N1: TMenuItem; N2: TMenuItem; N8: TMenuItem;
  StatusBar1: TStatusBar; procedure N4Click(Sender: TObject); procedure
  OpenExecute(Sender: TObject); procedure PoiscExecute(Sender: TObject);
  procedure N2Click(Sender: TObject); procedure N8Click(Sender: TObject);
  procedure FormCreate(Sender: TObject); procedure N1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
Form1: TForm1;
implementation
uses Unit2, Unit3, Unit4, Unit5, Unit6, Unit7, Unit8;
{$R *.dfm}
procedure TForm1. N4Click(Sender: TObject);
begin
  Close; // закрывает программу
end;
procedure TForm1. OpenExecute(Sender: TObject);
begin
  form2: =tform2. Create(self); // открывает форму "Сведения о
сотрудниках"
  form2. showmodal;
end;
procedure TForm1. PoiscExecute(Sender: TObject);
begin
  form7: =tform7. Create(self); // открывает форму "Поиск"
  form7. showmodal;
end;
```

```

procedure TForm1. N2Click(Sender: TObject);
begin
form8: =tform8. Create(self); // открывает форму "Редактировать"
form8. showmodal;
end;
procedure TForm1. N8Click(Sender: TObject);
begin
end;
procedure TForm1. FormCreate(Sender: TObject);
begin
statusbar1. Panels. Items [1]. Text: =datetostr(date);
end;
procedure TForm1. N1Click(Sender: TObject);
begin
aAPPLICATION. HelpFile: ='HELP. HLP';
APPLICATION. HelpCommand(HELP_FINDER,0);
end;
end.
unit Unit2;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ADODB, Grids, DBGrids, ExtCtrls, DBCtrls, StdCtrls, Mask,
ComCtrls;
type
TForm2 = class(TForm)
Label1: TLabel; Label2: TLabel; Label3: TLabel; Label4: TLabel; Label5:
TLabel;
Label6: TLabel; Label7: TLabel; Label8: TLabel; Label10: TLabel; Label11:
TLabel; Label12: TLabel; DBEdit1: TDBEdit; DBEdit2: TDBEdit; DBEdit3:
TDBEdit; DBEdit4: TDBEdit; DBEdit5: TDBEdit; DBEdit9: TDBEdit; DBEdit10:
TDBEdit;
Button1: TButton; Button2: TButton; Button3: TButton; Button4: TButton;
Button5: TButton; Button6: TButton; DataSource1: TDataSource;
Table1: TADOTable; DataSource2: TDataSource; Table6: TADOTable;
DBLookupComboBox1: TDBLookupComboBox; Table7: TADOTable;
DataSource3: TDataSource; DBLookupComboBox2: TDBLookupComboBox;
DateTimePicker1: TDateTimePicker; MaskEdit1: TMaskEdit; procedure
Button6Click(Sender: TObject); procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject); procedure Button3Click(Sender:
TObject); procedure Button4Click(Sender: TObject); procedure
Button5Click(Sender: TObject);
procedure FormCreate(Sender: TObject); procedure dbedit1KeyPress(Sender:
TObject; var Key: Char); procedure dbedit2KeyPress(Sender: TObject; var Key:

```

```

Char); procedure DBEdit3KeyPress(Sender: TObject; var Key: Char);
      procedure DBEdit4KeyPress(Sender: TObject; var Key: Char);
      private
      { Private declarations }
      public
      { Public declarations }
      end; var
      Form2: TForm2;
      implementation
      uses Unit3, Unit4, Unit5, Unit6;
      {$R *.dfm}
      procedure TForm2. Button6Click(Sender: TObject);
      begin
      Close; // закрывает форму "Сведения о сотруднике"
      end;
      procedure TForm2. Button1Click(Sender: TObject);
      begin
      form3: =tform3. Create(self); // запускает форму "Паспортные данные"
      form3. showmodal;
      end;
      procedure TForm2. Button2Click(Sender: TObject);
      begin
      unit4. Form4. Edit1. Text: =table1. Fields. Fields [0]. AsString;
      // form4. edit1. text: =table1. fieldbyname('код_с'). asstring;
      // form4: =tform4. Create(self); // запускает форму "Образование"
      form4. showmodal;
      end;
      procedure TForm2. Button3Click(Sender: TObject);
      begin
      form5: =tform5. Create(self); // запускает форму "Прием на работу и
      переводы"
      form5. showmodal;
      end;
      procedure TForm2. Button4Click(Sender: TObject);
      begin
      form6: =tform6. Create(self);
      form6. showmodal;
      end;
      procedure TForm2. Button5Click(Sender: TObject);
      var
      spisok: string;
      begin
      if (Dbedit1. text=") or (Dbedit2. text=") or (Dbedit3. text=") or
      (DblookupCombobox1. text=") or (DblookupCombobox2. text=") or (Dbedit5.

```

```

text=") or (MaskEdit1. Text=") or (Dbedit9. text=") or (Dbedit10. text=") then Begin
  if dbedit1. text=" then spisok: =spisok+'Фамилия, ';
  if dbedit2. text=" then spisok: =spisok+'Имя, ';
  if dbedit3. text=" then spisok: =spisok+'Отчество, ';
  if DblookupCombobox1. text=" then spisok: =spisok+'Семейное положение,
';
  if DblookupCombobox2. text=" then spisok: =spisok+'Пол, ';
  if dbedit5. text=" then spisok: =spisok+'Номер удостоверения, ';
  if MaskEdit1. Text=" then spisok: =spisok+'ИИН ';
  AsDateTime: =DateTimePicker1. DateTime;
  table1. post;
  end; button1. Enabled: =true; button2. Enabled: =true; button3. Enabled:
=true;
  button4. Enabled: =true;
  end;
  procedure TForm2. FormCreate(Sender: TObject);
  begin
    table1. ConnectionString: ='Provider=Microsoft. Jet. OLEDB.4.0; Data
Source='+getcurrentdir+'\Database. mdb'; table1. Active: =true;
    table6. ConnectionString: ='Provider=Microsoft. Jet. OLEDB.4.0; Data
Source='+getcurrentdir+'\Database. mdb'; table6. Active: =true;
    Table7. ConnectionString: ='Provider=Microsoft. Jet. OLEDB.4.0; Data
Source='+getcurrentdir+'\Database. mdb '; table7. Active: =true;
    table1. insert;
  end;
  procedure TForm2. dbedit1KeyPress(Sender: TObject; var Key: Char);
  begin
    case Key of
      'a'. 'я'; 'A'. 'Я'; #8: else Key: = #0;
    end; end;
  procedure TForm2. dbedit2KeyPress(Sender: TObject; var Key: Char);
  begin
    case Key of 'a'. 'я'; 'A'. 'Я'; #8:
      else Key: = #0;
    end; end;
  procedure TForm2. DBEdit3KeyPress(Sender: TObject; var Key: Char);
  begin
    case Key of 'a'. 'я'; 'A'. 'Я'; #8:
      else Key: = #0;
    end; end;
  procedure TForm2. DBEdit4KeyPress(Sender: TObject; var Key: Char);
  begin
    case Key of 'a'. 'я'; 'A'. 'Я'; #8:
      else Key: = #0;

```

```

end; end; end.
unit Unit3;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Grids, DBGrids, DB, ADODB, ExtCtrls, DBCtrls, StdCtrls, Mask,
ComCtrls;
Type
TForm3 = class(TForm) Label2: TLabel; Label3: TLabel; Label4: TLabel;
Label5: TLabel; Label6: TLabel; Label7: TLabel; Label8: TLabel; Label9: TLabel;
Label10: TLabel; Label11: TLabel; DBEdit4: TDBEdit; DBEdit5: TDBEdit;
DBEdit8: TDBEdit; DBEdit9: TDBEdit; DataSource1: TDataSource; Table2:
TADOTable;
    Button1: TButton; Button2: TButton; DBLookupComboBox1:
TDBLookupComboBox; DataSource2: TDataSource; Table8: TADOTable;
    DateTimePicker1: TDateTimePicker; MaskEdit1: TMaskEdit; MaskEdit2:
TMaskEdit; MaskEdit3: TMaskEdit; DBEdit1: TDBEdit; procedure
Button1Click(Sender: TObject); procedure Button2Click(Sender: TObject);
procedure FormCreate(Sender: TObject); procedure DBEdit1KeyPress(Sender:
TObject; var Key: Char);
    private
    { Private declarations }
    public
    { Public declarations }
end;
var
Form3: TForm3;
implementation
{$R *.dfm}
procedure TForm3.Button1Click(Sender: TObject);
var
spisok1: string;
begin
if (MaskEdit1. Text='') or (MaskEdit2. Text='') or (Dbedit4. text='') or
(DblookupCombobox1. text='') or (Dbedit5. text='') or (MaskEdit3. Text='') or
(Dbedit8. text='') then Begin
if MaskEdit1. Text="" then spisok1: =spisok1+'Удостоверение ';
if MaskEdit2. Text="" then spisok1: =spisok1+'Номер, ';
if dbedit4. text="" then spisok1: =spisok1+'Выдан, ';
if DblookupCombobox1. text="" then spisok1: =spisok1+' Место жительства,
';
if dbedit5. text="" then spisok1: =spisok1+'Адрес, ';
// if dbedit6. text="" then spisok1: =spisok1+'Дата рождения, ';
if MaskEdit3. Text="" then spisok1: =spisok1+'Индекс, ';

```



```

Showmessage('Вы не заполнили поля: '+Spisok1);
End else begin
table2. FieldByName('Серия'). Asstring: =(MaskEdit1. Text);
table2. FieldByName('Номер'). Asstring: =(MaskEdit2. Text);
table2. FieldByName('Дата рождения'). AsDateTime: =DateTimePicker1.
DateTime;
table2. FieldByName('Индекс'). Asstring: =(MaskEdit3. Text);
table2. post;
end; end;
procedure TForm3. Button2Click(Sender: TObject);
begin
close;
end;
procedure TForm3. FormCreate(Sender: TObject);
begin
Table2. ConnectionString: ='Provider=Microsoft. Jet. OLEDB.4.0; Data
Source='+getcurrentdir+'\Database. mdb '; // связывает форму "Паспортные
данные" с базой данных, таблицей " Паспортные данные "
table2. Active: =true;
Table8. ConnectionString: ='Provider=Microsoft. Jet. OLEDB.4.0; Data
Source='+getcurrentdir+'\Database. mdb '; table8. Active: =true;
table2. insert;
end;
procedure TForm3. DBEdit1KeyPress(Sender: TObject; var Key: Char);
begin
case Key of 'a'. 'я':; 'A'. 'Я':; #8:
else Key: = #0;
end; end; end.
unit Unit4;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Grids, DBGrids, DB, ADODB, ExtCtrls, DBCtrls, StdCtrls,
ComCtrls;
type
Form4 = class(TForm)
DataSource1: TDataSource; Table3: TADOTable; DBGrid1: TDBGrid;
Edit1: TEdit; Edit2: TEdit; Edit3: TEdit; Button1: TButton;
Edit5: TEdit; Edit7: TEdit; Edit8: TEdit; Edit9: TEdit; Edit10: TEdit;
Label1: TLabel; Label2: TLabel; Label3: TLabel; Label4: TLabel; Label5:
TLabel; Label6: TLabel; Label7: TLabel; Label8: TLabel; Label9: TLabel;
Label10: TLabel; Label11: TLabel; Button2: TButton; DateTimePicker1:
TDateTimePicker; Edit4: TEdit; Edit6: TEdit; procedure FormCreate(Sender:
TObject); procedure Button1Click(Sender: TObject); procedure

```

```

Button2Click(Sender: TObject); procedure Edit1KeyPress(Sender: TObject; var Key:
Char); procedure Edit2KeyPress(Sender: TObject; var Key: Char);
    procedure Edit6KeyPress(Sender: TObject; var Key: Char);
    private
    { Private declarations }
    public
    { Public declarations }
    end; var
    Form4: TForm4;
    implementation
    {$R *. dfm}
    procedure TForm4. FormCreate(Sender: TObject);
    begin
        Table3. ConnectionString: ='Provider=Microsoft. Jet. OLEDB.4.0; Data
Source='+getcurrentdir+'\Database. mdb '; // связывает форму "Образование" с
базой данных, таблицей "Образование"
        table3. Active: =true;
    end;
    procedure TForm4. Button1Click(Sender: TObject);
    var
    spisok: string;
    begin
        table3. Insert;
        table3. FieldByName('код'). AsInteger: =strtoint(edit1. Text);
        table3. FieldByName('Учебное заведение'). Asstring: =(edit2. Text);
        table3. FieldByName('Наименование заведения'). Asstring: =(edit3. Text);
        table3. FieldByName('Вид обучения'). Asstring: =(edit4. Text);
        table3. FieldByName('Факультет'). Asstring: =(edit5. Text);
        table3. FieldByName('Образование'). Asstring: =(edit6. Text);
        table3. FieldByName('Документ'). Asstring: =(edit7. Text);
        table3. FieldByName('Специальность'). Asstring: =(edit8. Text);
        table3. FieldByName('Квалификация'). Asstring: =(edit9. Text);
        table3. FieldByName('Номер'). Asstring: =(edit10. Text);
        table3. FieldByName('Дата выдачи'). AsDateTime: =DateTimePicker1.
DateTime;
        if (edit1. Text='') or(edit4. Text='') then Begin
        if edit1. Text='' then spisok: =spisok+'код, ';
        if edit4. Text='' then spisok: =spisok+'Вид обучения, ';
        Showmessage('Вы не заполнили поля: '+Spisok);
        End else
        table3. Post;
        end;
    procedure TForm4. Button2Click(Sender: TObject);
    begin

```

```

close;
end;
procedure TForm4. Edit1KeyPress(Sender: TObject; var Key: Char);
begin
case Key of '0'. '9':; #8:
else Key: = #0;
end; end;
procedure TForm4. Edit2KeyPress(Sender: TObject; var Key: Char);
begin
case Key of 'a'. 'я':; 'A'. 'Я':; #8:
else Key: = #0;
end; end;
procedure TForm4. Edit6KeyPress(Sender: TObject; var Key: Char);
begin
case Key of 'a'. 'я':; 'A'. 'Я':; #8;; #32:
else Key: = #0;
end;
end; end.
unit Unit5;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Grids, DBGrids, DB, ADODB, ExtCtrls, DBCtrls, StdCtrls, Mask,
ComCtrls;
type
TForm5 = class(TForm)
DataSource1: TDataSource; Table4: TADOTable; Label1: TLabel;
Label2: TLabel; Label3: TLabel; Label4: TLabel; Label5: TLabel;
Label6: TLabel; DBEdit1: TDBEdit; DBEdit2: TDBEdit; DBEdit3: TDBEdit;
DBEdit4: TDBEdit; DBEdit6: TDBEdit; Button1: TButton; Button2:
TButton;
DBEdit5: TDBEdit; procedure Button2Click(Sender: TObject);
procedure Button1Click(Sender: TObject); procedure FormCreate(Sender:
TObject); procedure DBEdit4KeyPress(Sender: TObject; var Key: Char); procedure
DBEdit1KeyPress(Sender: TObject; var Key: Char); procedure
DBEdit2KeyPress(Sender: TObject; var Key: Char);
private
{ Private declarations }
public
{ Public declarations }
end;
var
Form5: TForm5;
implementation

```

```

{$R *. dfm}
procedure TForm5. Button2Click(Sender: TObject);
begin
Close; // закрывает форму "Работа"
end;
procedure TForm5. Button1Click(Sender: TObject);
var
spisok2: string;
begin
if (Dbedit4. text="") then
Begin
if dbedit4. text="" then spisok2: =spisok2+'Профессия(Должность), ';
Showmessage('Вы не заполнили поля: '+Spisok2);
End else table4. post;
end;
procedure TForm5. FormCreate(Sender: TObject);
begin
Table4. ConnectionString: ='Provider=Microsoft. Jet. OLEDB.4.0; Data
Source='+getcurrentdir+"\Database. mdb ' ; // связывает форму "Работа" с базой
данных, таблицей "Работа"
table4. Active: =true;
table4. insert;
end;
procedure TForm5. DBEdit4KeyPress(Sender: TObject; var Key: Char);
begin
case Key of 'a'. 'я'; 'A'. 'Я'; #8:
else Key: = #0;
end; end;
procedure TForm5. DBEdit1KeyPress(Sender: TObject; var Key: Char);
begin
case Key of '0'. '9'; #8:
else Key: = #0;
end; end;
procedure TForm5. DBEdit2KeyPress(Sender: TObject; var Key: Char);
begin
case Key of '0'. '9'; #8:
else Key: = #0;
end; end; end.
unit Unit6;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ADODB, ExtCtrls, DBCtrls, Grids, DBGrids, StdCtrls, Mask,
ComCtrls;

```

```

type
TForm6 = class(TForm)
Label1: TLabel; Label2: TLabel; Label3: TLabel; Label4: TLabel;
Label5: TLabel; Label6: TLabel; DBEdit1: TDBEdit; DBEdit2: TDBEdit;
DBEdit3: TDBEdit; Label7: TLabel; Label8: TLabel; Label9: TLabel;
Label10: TLabel; Label11: TLabel; DBEdit4: TDBEdit; DBEdit5: TDBEdit;
DBEdit6: TDBEdit; DBEdit7: TDBEdit; Table5: TADOTable;
DataSource1: TDataSource; Button1: TButton; Button2: TButton;
DBLookupComboBox1: TDBLookupComboBox; DataSource2:
TDataSource; Table9: TADOTable; Table10: TADOTable; DataSource3:
TDataSource;
DBLookupComboBox2: TDBLookupComboBox; DBLookupComboBox3:
TDBLookupComboBox; DataSource4: TDataSource; Table11: TADOTable;
DateTimePicker1: TDateTimePicker; procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject); procedure FormCreate(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end; var
Form6: TForm6;
implementation
{$R *.dfm}
procedure TForm6.Button1Click(Sender: TObject);
var
spisok4: string;
begin
if (DblookupCombobox1. text="") or (DblookupCombobox2. text=") then
Begin
if DblookupCombobox1. text=" then spisok4: =spisok4+'Годность к военной
службе,';
if DblookupCombobox2. text=" then spisok4: =spisok4+'Обязанность, ';
Showmessage('Вы не заполнили поля: '+Spisok4);
end
else
begin
table5. FieldByName('Выдан'). AsDateTime: =DateTimePicker1. DateTime;
table5. post;
end; end;
procedure TForm6.Button2Click(Sender: TObject);
begin
close; // закрывает форму "ВУС"
end;
procedure TForm6.FormCreate(Sender: TObject);

```

```

begin
  Table5. ConnectionString: ='Provider=Microsoft. Jet. OLEDB.4.0; Data
Source='+getcurrentdir+'\Database.mdb '; table5. Active: =true;
  Table9. ConnectionString: ='Provider=Microsoft. Jet. OLEDB.4.0; Data
Source='+getcurrentdir+'\Database.mdb '; table9. Active: =true;
  Table10. ConnectionString: ='Provider=Microsoft. Jet. OLEDB.4.0; Data
Source='+getcurrentdir+'\Database.mdb '; table10. Active: =true;
  Table11. ConnectionString: ='Provider=Microsoft. Jet. OLEDB.4.0; Data
Source='+getcurrentdir+'\Database.mdb '; table11. Active: =true;
  table5. insert;
end; end.
unit Unit7;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, Grids, DBGrids, ExtCtrls, StdCtrls, DBCtrls, ImgList;
type
  TForm7 = class(TForm)
    DBGrid1: TDBGrid; Edit1: TEdit; Edit2: TEdit; Edit3: TEdit; Edit4: TEdit;
    Edit5: TEdit; Edit6: TEdit; Edit7: TEdit; Edit8: TEdit; Edit9: TEdit;
    Edit10: TEdit; Edit11: TEdit; Edit12: TEdit; Panel1: TPanel; Button1:
    TButton; Button2: TButton; ADOQuery1: TADOQuery; Label1: TLabel;
    Label2: TLabel; Label3: TLabel; Label4: TLabel; Label5: TLabel;
    Label6: TLabel; Label7: TLabel; Label8: TLabel; Label9: TLabel;
    Label10: TLabel; Label11: TLabel; Label12: TLabel; DataSource1:
    TDataSource; Button4: TButton; DBNavigator1: TDBNavigator;
    Button3: TButton; Button5: TButton; procedure Button1Click(Sender:
    TObject); procedure Button2Click(Sender: TObject);
    procedure Button4Click(Sender: TObject); procedure Button5Click(Sender:
    TObject); procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure Edit2KeyPress(Sender: TObject; var Key: Char);
    procedure Edit3KeyPress(Sender: TObject; var Key: Char);
    procedure Edit11KeyPress(Sender: TObject; var Key: Char);
    procedure Button3Click(Sender: TObject);
    procedure Edit4KeyPress(Sender: TObject; var Key: Char);
    procedure Edit5KeyPress(Sender: TObject; var Key: Char);
    procedure Edit7KeyPress(Sender: TObject; var Key: Char);
    procedure Edit8KeyPress(Sender: TObject; var Key: Char);
    procedure Edit10KeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

```

```

var
Form7: TForm7;
implementation
uses Unit9, Unit10;
{$R *.dfm}
procedure TForm7. Button1Click(Sender: TObject);
begin
if (edit1. text="") and (edit2. text="") and (edit3. text="") and (edit4. text="") and
(edit5. text="") and (edit6. text="") and (edit7. text="") and (edit8. text="") and (edit9.
text="") and (edit10. text="") and (edit11. text="") and (edit12. text="") then
Showmessage('Вы не внесли критерии поиска')
else
Begin
adoquery1. Parameters. ParamByName('p1'). Value: =' '+edit1. text+'%';
adoquery1. Parameters. ParamByName('p2'). Value: =' '+edit2. text+'%';
adoquery1. Parameters. ParamByName('p3'). Value: =' '+edit3. text+'%';
adoquery1. Parameters. ParamByName('p4'). Value: =' '+edit4. text+'%';
adoquery1. Parameters. ParamByName('p5'). Value: =' '+edit5. text+'%';
adoquery1. Parameters. ParamByName('p6'). Value: =' '+edit6. text+'%';
adoquery1. Parameters. ParamByName('p7'). Value: =' '+edit7. text+'%';
adoquery1. Parameters. ParamByName('p8'). Value: =' '+edit8. text+'%';
adoquery1. Parameters. ParamByName('p9'). Value: =' '+edit9. text+'%';
adoquery1. Parameters. ParamByName('p10'). Value: =' '+edit10. text+'%';
adoquery1. Parameters. ParamByName('p11'). Value: =' '+edit11. text+'%';
adoquery1. Parameters. ParamByName('p12'). Value: =' '+edit12. text+'%';
ADOQuery1. Active: = True;
end; end;
procedure TForm7. Button2Click(Sender: TObject);
begin
Form9. QuickRep1. Preview;
end;
procedure TForm7. Button4Click(Sender: TObject);
begin
Edit1. Text: = ""; Edit2. Text: = ""; Edit3. Text: = ""; Edit4. Text: = "";
Edit5. Text: = ""; Edit6. Text: = ""; Edit7. Text: = ""; Edit8. Text: = "";
Edit9. Text: = ""; Edit10. Text: = ""; Edit11. Text: = ""; Edit12. Text: = "";
ADOQuery1. Active: = False;
end;
procedure TForm7. Button5Click(Sender: TObject);
begin
close;
end;
procedure TForm7. Edit1KeyPress(Sender: TObject; var Key: Char);
begin

```

```

case Key of 'a'. 'я'; 'A'. 'Я'; #8:
else Key: = #0;
end; end;
procedure TForm7. Edit2KeyPress(Sender: TObject; var Key: Char);
begin
case Key of 'a'. 'я'; 'A'. 'Я'; #8:
else Key: = #0;
end; end;
procedure TForm7. Edit3KeyPress(Sender: TObject; var Key: Char);
begin
case Key of 'a'. 'я'; 'A'. 'Я'; #8:
else Key: = #0;
end; end;
procedure TForm7. Edit11KeyPress(Sender: TObject; var Key: Char);
begin
case Key of 'a'. 'я'; 'A'. 'Я'; #8:
else Key: = #0;
end; end;
procedure TForm7. Button3Click(Sender: TObject);
begin
Form10. QuickRep1. Preview;
end;
procedure TForm7. Edit4KeyPress(Sender: TObject; var Key: Char);
begin
case Key of '0'. '9'; #8:
else Key: = #0;
end; end;
procedure TForm7. Edit5KeyPress(Sender: TObject; var Key: Char);
begin
case Key of '0'. '9'; #8:
else Key: = #0;
end; end;
procedure TForm7. Edit7KeyPress(Sender: TObject; var Key: Char);
begin
case Key of '0'. '9'; #8:
else Key: = #0;
end; end;
procedure TForm7. Edit8KeyPress(Sender: TObject; var Key: Char);
begin
case Key of '0'. '9'; #8:
else Key: = #0;
end; end;
procedure TForm7. Edit10KeyPress(Sender: TObject; var Key: Char);
begin

```



```

case Key of 'a'. 'я':; 'A'. 'Я':; #8.; #32:
else Key: = #0;
end; end; end.
unit Unit8;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ADODB, StdCtrls, ExtCtrls, DBCtrls, Grids, DBGrids;
type
TForm8 = class(TForm)
Label1: TLabel; Label2: TLabel; Label3: TLabel; Label4: TLabel;
Label5: TLabel; Label6: TLabel; Label7: TLabel; Label8: TLabel;
Label9: TLabel; Label10: TLabel; Label11: TLabel; Label12: TLabel;
DBGrid1: TDBGrid; DBNavigator1: TDBNavigator; Edit1: TEdit;
Edit2: TEdit; Edit3: TEdit; Edit4: TEdit; Edit5: TEdit; Edit6: TEdit;
Edit7: TEdit; Edit8: TEdit; Edit9: TEdit; Edit10: TEdit; Edit11: TEdit;
Edit12: TEdit; Panel1: TPanel; Button1: TButton; Button4: TButton;
ADOQuery1: TADOQuery; DataSource1: TDataSource; Button2: TButton;
Button3: TButton; procedure Button1Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
procedure Edit2KeyPress(Sender: TObject; var Key: Char);
procedure Edit3KeyPress(Sender: TObject; var Key: Char);
procedure Edit11KeyPress(Sender: TObject; var Key: Char);
procedure Edit4KeyPress(Sender: TObject; var Key: Char);
procedure Edit5KeyPress(Sender: TObject; var Key: Char);
procedure Edit7KeyPress(Sender: TObject; var Key: Char);
procedure Edit8KeyPress(Sender: TObject; var Key: Char);
procedure Edit10KeyPress(Sender: TObject; var Key: Char); private
{ Private declarations }
public
{ Public declarations }
end;
var
Form8: TForm8;
implementation
{$R *. dfm}
procedure TForm8. Button1Click(Sender: TObject);
begin
if (edit1. text=") and (edit2. text=") and (edit3. text=") and (edit4. text=") and
(edit5. text=") and (edit6. text=") and (edit7. text=") and (edit8. text=") and (edit9.
text=") and (edit10. text=") and (edit11. text=") and (edit12. text=") then

```

Showmessage('Вы не внесли критерии поиска')

else

Begin

adoquery1.Parameters.ParamByName('p1').Value:='%'+edit1.text+'%';

adoquery1.Parameters.ParamByName('p2').Value:='%'+edit2.text+'%';

adoquery1.Parameters.ParamByName('p3').Value:='%'+edit3.text+'%';

adoquery1.Parameters.ParamByName('p4').Value:='%'+edit4.text+'%';

adoquery1.Parameters.ParamByName('p5').Value:='%'+edit5.text+'%';

adoquery1.Parameters.ParamByName('p6').Value:='%'+edit6.text+'%';

adoquery1.Parameters.ParamByName('p7').Value:='%'+edit7.text+'%';

adoquery1.Parameters.ParamByName('p8').Value:='%'+edit8.text+'%';

adoquery1.Parameters.ParamByName('p9').Value:='%'+edit9.text+'%';

adoquery1.Parameters.ParamByName('p10').Value:='%'+edit10.text+'%';

adoquery1.Parameters.ParamByName('p11').Value:='%'+edit11.text+'%';

adoquery1.Parameters.ParamByName('p12').Value:='%'+edit12.text+'%';

ADOQuery1.Active:=True;

end;

end;