

Министерство образования и науки Российской Федерации
Филиал Федерального государственного автономного образовательного учреждения
высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
в г. Нижневартовске

Кафедра «Информатика»

ДОПУСТИТЬ К ЗАЩИТЕ

И.о.зав.кафедрой «Информатика»

к.ф-м.н, доцент

/ А.В.Ялаев

« ___ » _____ 2018 г.

Разработка информационной системы по обработке данных анкетирования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ-09.03.04. 2018.429.ПЗ ВКР

Консультанты

Экономическая часть

к.э.н., доцент

/А.В.Прокопьев/

« ___ » _____ 2018г.

Безопасность жизнедеятельности

к.ф-м.н, доцент

/ А.В.Ялаев

« ___ » _____ 2018 г.

Руководитель работы

к.ф-м.н, доцент

/ А.В.Ялаев /

« ___ » _____ 2018 г.

Автор работы

обучающийся группы НвФл-422

/И.И. Шпуганыч/

« ___ » _____ 2018г.

Нормоконтролер

старший преподаватель

/Л.Н.Буйлушкина /

« ___ » _____ 2018г.

Нижневартовск 2018

АННОТАЦИЯ

Шпуганыч И.И. Разработка информационной системы по обработке данных анкетирования - Нижневартовск: филиал ЮУрГУ, Информатика: 2018, 85 с., 14 ил., 14 табл., библиогр. список – 20 наим., 3 прил.

В рамках выпускной квалификационной работы была разработана информационная система по обработке данных анкетирования.

Для достижения указанной цели в работе решены следующие задачи:

- изучение особенностей обработки анкетирования в филиале на данный момент;
- реализация программных модулей информационной системы;
- создание и ведение единой базы данных, обеспечение единства и целостности информации;
- описание работы системы.

Проведен анализ предметной области. Выполнен расчет технико-экономической эффективности информационной системы данных анкетирования сотрудников. Предоставлены рекомендации и требования по обеспечению безопасности жизнедеятельности. Проведен литературный обзор.

					00 02 04 2010 120 ПЗ								
<i>Изм</i>	<i>Лист</i>	<i>№ докум</i>	<i>Подпис</i>	<i>Дата</i>	Разработка информационной системы по обработке данных анкетирования			<i>Лит</i>		<i>Лист</i>	<i>Листов</i>		
<i>Разработал</i>	<i>Шпуганыч</i>							D	V	D	5	25	
<i>Проверил</i>	<i>Ялаев Д. Р.</i>							Филиал ФГАОУ ВО «ЮУрГУ (НИУ)»					
<i>И контр</i>	<i>Буйлушкима Л.</i>												
<i>Утвердил</i>	<i>Ялаев Д. Р.</i>												

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	8
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ	10
1.1 Процесс анкетирования.....	10
1.2 Обоснование целесообразности разработки.....	11
1.3 Изучение программных средств, необходимых для разработки и реализации работы.....	11
1.4 Формулировка требований к функциональности проектируемого программного средства	16
1.5 План разработки.....	18
1.6 Постановка задачи	19
1.7 Анализ существующей ситуации в филиале ЮУрГУ	19
1.8 Обзор существующих подходов к проектированию пользовательских интерфейсов.....	20
1.9 Критерии качества программы.....	23
2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА.....	25
2.1 Выбор средств разработки для реализации системы.....	25
2.2 Техническое задание	36
2.3 Реализация СУБД на Microsoft Access	39
2.4 Реализация информационной системы.....	42
2.5 Описание программы	44
3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ	52
3.1 Составление сметы затрат.....	52
3.2 Оценка технико-экономической эффективности	55
4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ.....	57
4.1 Характеристика условий труда программиста	58
4.2 Требования к производственным помещениям.....	59
4.3 Эргономические требования к рабочему месту	63

4.4	Режим труда.....	66
4.5	Организация пожарной профилактики.....	68
	ЗАКЛЮЧЕНИЕ	71
	БИБЛИОГРАФИЧЕСКИЙ СПИСОК	72
ПРИЛОЖЕНИЯ		
	ПРИЛОЖЕНИЕ А. СВИДЕТЕЛЬСТВО О ПУБЛИКАЦИИ.....	74
	ПРИЛОЖЕНИЕ Б. ЛИСТИНГ ФОРМЫ ВВОДА ДАННЫХ	75
	ПРИЛОЖЕНИЕ В. КОМПАКТ-ДИСК.....	85

ВВЕДЕНИЕ

Анкетирование – один из наиболее распространенных и эффективных методов сбора первичной социологической и статистической информации. Это опрос сотрудников с помощью специального бланка с вопросами (анкеты). Анкетирование по сравнению с беседой и интервьюированием представляет собой как бы следующую ступень еще более жестко запрограммированного опроса. Цель анкетирования состоит в том, чтобы быстро и без больших затрат времени и средств получить объективное представление о мнении сотрудников по той или иной проблеме [1].

Преимущества анкетирования заключаются в том, что этот метод дает возможность оперативно собрать большой массив информации и выявить мнения по той или иной проблеме.

Принимая во внимание актуальность и востребованность анкетирования, принято решение о разработке информационной системы (далее – ИС).

Цель выпускной квалификационной работы заключается в разработке ИС по обработке результатов анкетирования. Отталкиваясь от поставленных задач, которые должны выполняться программой, нами определена модель программирования. Разработка ИС предполагается по принципу императивной модели программирования. Степень разработанности данной проблемы не позволяет говорить об ее достаточном исследовании, что подтверждает актуальность исследования.

Объект исследования – Филиал Федерального государственного автономного образовательного учреждения высшего образования «Южно-Уральский государственный университет» (национальный исследовательский университет) в г. Нижневартовске (далее – филиал ЮУрГУ).

Предмет исследования – процесс анкетирования сотрудников филиала на предмет удовлетворенности условиями труда. Область исследования очерчена проблематикой автоматизации процессов хранения и предоставления информации

в филиале с целью мониторинга и обработки анкетирования сотрудников. Предметом исследования установлены вопросы теоретического и практического характера, связанные с повышением результативности сотрудника посредством информационных технологий.

Для достижения указанной цели в выпускной квалификационной работе решаются следующие задачи:

- изучение особенностей обработки анкетирования в филиале на данный момент;
- реализация программных модулей ИС;
- создание и ведение единой базы данных, обеспечение единства и целостности информации;
- описание работы системы.

Разработанная ИС позволит значительно сократить трудозатраты, оперативно проводить обработку и мониторинг результатов проведенного анкетирования.

Выпускная квалификационная работа состоит из введения, четырех разделов, заключения и приложений.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ

1.1 Процесс анкетирования

Одним из наиболее информативных методов выборочного опроса считается анкетирование. Анкетирование представляет собой заполнение таблиц с предварительно подготовленным перечнем вопросов по интересующим вопросам. Анкета – это вопросник, на который предлагается ответить опрашиваемому лицу (респонденту). Анкетирование – письменная форма опроса, осуществляющаяся, как правило, заочно, т.е. в отсутствие непосредственного и прямого контакта опрашивающего с респондентом.

Анкетирование целесообразно в тех случаях, когда необходимо опросить большое количество респондентов за сравнительно короткое время, причем респонденты должны основательно подумать над своими ответами, имея перед глазами распечатанный вопросник.

Анкетирование крайне редко бывает сплошным (охватывающим всех членов исследуемой общности), гораздо чаще оно имеет выборочный характер.

Анкетирование на современном этапе применяется не только с целью исследования потребительских предпочтений, запросов, нужд и т.д., но и для отслеживания мнений сотрудников о предприятии. Часть специалистов области анкетирования сходятся во мнении, что опрос сотрудников даёт возможность получения разнообразной информации о состоянии предприятия, настроя сотрудников, стимулирования их расти по карьерной лестнице, о взаимоотношении в коллективе.

В последнее время сфера применения анкет расширилась: если раньше при их помощи исследовалось потребительское поведение, то в настоящее время к ним прибегают, чтобы отслеживать мнение специалистов по тем или иным направлениям. С каждым годом появляются новые технологии формирования анкет и проведения непосредственно анкетирования. В целом преимущества

анкетирования сотрудников заключаются в том, что этот метод дает возможность оперативно собрать большой массив информации, которую необходимо быстро обрабатывать, т.к. она стремительно устаревает и утрачивает свою ценность для исследователя, а так же выявить мнения сотрудников по той или иной проблеме.

1.2 Обоснование целесообразности разработки

На данный момент обработка данных анкетирования в филиале ЮУрГУ, происходит в ручную, что занимает довольно много времени и допускает вероятность возникновения ошибок при работе с большим объемом однотипных данных.

Сегодняшний рынок предлагает услуги по обработке данных анкетирования сторонними компаниями, что не подходит политикой конфиденциальности, и не предлагает информационных систем, которые подошли бы по определенным критериям для работы в филиале. По этой причине данная разработка является целесообразной.

Пользователями данной ИС будут сотрудники филиала ЮУрГУ ответственные за сбор и обработку анкетирования. Применение ИС значительно облегчит труд и сократит затраты на выполнение работы.

1.3 Изучение программных средств, необходимых для разработки и реализации работы

Прежде чем приступить к рассмотрению средств разработки, которые могут быть применены для создания программ, необходимо определиться с ключевыми понятиями, которые будут использоваться. В соответствии с тематикой базовым термином для нас, безусловно, является «средства разработки программ». Применительно к области разработки программного обеспечения это определение может звучать следующим образом: средства разработки программного

обеспечения – совокупность приемов, методов, методик, а кроме того набор инструментальных, используемых разработчиком для создания программного кода, отвечающего заданным требованиям.

С учетом данного определения термин «разработка программ» будет звучать следующим образом: разработка программ – сложный процесс, главной целью которого является создание, сопровождение программного кода, обеспечивающего необходимый уровень надежности и качества. Для достижения основной цели разработки программ применяются средства разработки программного обеспечения.

1.3.1 Основные средства, используемые на разных этапах разработки

В зависимости от предметной области и задач, поставленных перед разработчиками, разработка программ может представлять собой довольно сложный, постепенный процесс, в котором задействовано большое число участников и различных средств. Для того чтобы определить, когда и в каких случаях применяются те ли иные средства, выделим ключевые этапы разработки программного обеспечения. Наибольший интерес для проблематики рассматриваемого вопроса представляют следующие этапы разработки:

- проектирование приложения;
- реализация программного кода приложения;
- тестирование приложения.

Сознательно опустим этапы, связанные с написанием технического задания, планирования сроков и т.д. Причина этого состоит в том, что на данных этапах практически не применяются специфические средства разработки.

1.3.2 Средства проектирования приложений

На стадии проектирования приложения в зависимости от сложности разрабатываемого программного продукта, непосредственно зависящего от предъявляемых требований, выполняются следующие задачи:

- анализ требований;
- разработка архитектуры будущего программного обеспечения;
- разработка устройств основных компонент программного обеспечения;
- разработка макетов пользовательских интерфейсов.

Результатом проектирования как правило является «Технический проект» (Software Architecture Document) или «Эскизный проект» (Software Design Document). Цель анализа требований выполняется с применением методов анализа и синтеза с учетом экспертного опыта проектировщика. Итогом анализа является содержательная или формализованная модель процесса функционирования программы. В зависимости от трудности процесса с целью построения данных моделей могут быть применены различные методы и вспомогательные средства. В общем случае для описания моделей как правило применяются следующие нотации:

- блок-схемы;
- ER-диаграммы;
- макеты и математические модели.

Когда разрабатываемый программный продукт предназначен для автоматизации какой-либо сложной деятельности, моделирование выполняется до формирования технических требований к будущему продукту. Результаты анализа позволяют сформировать обоснованные требования к той или иной функциональности разрабатываемой программы и просчитать реальную выгоду от внедрения разрабатываемого продукта. Более того, выходит так, что согласно результатам анализа первоначальные цели и задачи автоматизации кардинально

меняются или по результатам оценки эффективности разработки и внедрения принимается решение продукт не разрабатывать.

Целью второй и третьей задачи из приведенного перечня является описание будущей системы, понятной для кодировщика. Здесь большое значение имеет то, какую парадигму программирования следует применять при написании программы. В качестве примера основных парадигм необходимо привести следующее:

- функциональное программирование;
- структурное программирование;
- императивное программирование;
- логическое программирование;
- объектно-ориентированное программирование.

Выбор её во многом зависит от сформировавшихся привычек, навыка, традиций, инструментальных средств, которыми располагает коллектив разработчиков. В некоторых случаях разрабатываемый программный продукт до такой степени сложен, что для решения ряда задач в разных компонентах системы используются разные парадигмы. Следует отметить, что выбор того или иного подхода накладывает ограничения на средства, которые будут использованы на этапе реализации программного кода. Результатом решения данной задачи в зависимости от подхода могут быть:

- диаграмма классов и т.д;
- описание модулей структур и их программного интерфейса.

Разработка макетов пользовательских интерфейсов предполагает создание наглядного представления того, как будут выглядеть те или иные формы, окна в разрабатываемом приложении. Решение данной задачи основывается на применение средств дизайнера, которые в данном разделе рассматриваться не будут.

1.3.3 Средства реализации программного кода

На стадии реализации программного кода выполняется кодирование отдельных компонент программы в соответствии с разработанным техническим проектом. Средства, которые могут быть применены, в значительной степени зависят от того, какие подходы были использованы во время проектирования и, кроме этого, от уровня проработанности технического проекта. Тем не менее, из числа средств разработки программного кода следует отметить следующие основные виды средств (в скобках приведены примеры средств):

- языки программирования (C++, Java, C#, Delphi и т.д.);
- методики алгоритмирования;
- средства создания пользовательского интерфейса (MFC, WPF и т.д.) ;
- средства управления версиями программного кода (cvs, svn, VSS) ;
- средства получения исполняемого кода (MS Visual Studio, gcc и другие) ;
- средства управления базами данных (Oracle, MS SQL, InterBase, MS Access, MySQL);
- отладчики (MS Visual Studio, gdb и т.д.).

1.3.4 Средства тестирования программ

Основными задачами тестирования является контроль соответствия функциональности разработанной программы первоначальным требованиям, а кроме того обнаружение ошибок, которые в явном либо неявном виде проявляются во время работы программы. Среди основных работ по тестированию можно отметить следующее:

- тестирование процесса установки;
- функциональное тестирование;
- тестирование безопасности;
- конфигурационное тестирование;
- тестирование взаимодействия;

- тестирование процесса установки;
- тестирование удобства пользования;
- тестирование на отказ и восстановление;
- нагрузочное тестирование.

Из числа основных видов средств, которые могут быть использованы для выполнения поставленных работ можно привести следующие:

- средства для тестирования функциональности (TEST – Parasoft, QACenter – Compuware, Borland SilkTest);
- средства анализа кода (Code Wizard – ParaSoft, Purify – Rational Softawre. Test Coverage – Semantic);
- средства для тестирования производительности (QACenter Performance – Compuware).

1.4 Формулировка требований к функциональности проектируемого программного средства

Информационное обеспечение – это совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, методология построения баз данных.

Информационное обеспечение должно удовлетворять пользователя по своей упорядоченности, точности, достоверности и оперативности представления информации с целью решения поставленных задач, а кроме того однозначности и удобства ее восприятия всеми потребителями.

Свойство объектов системы должны иметь возможность оформляться в виде сложной структуры, ссылающейся на прочие объекты и хранящей историческую последовательность значений.

Система должна обеспечивать поддержку иерархической структуры типов объектов, реализующей механизм наследования свойств объектов.

Система должна иметь механизм регистрации событий, который объединяет изменение свойств объектов, спровоцированное каким-либо внешним событием, с дальнейшей реакцией системы на это событие.

Форма представления выходной информации должна согласовываться с пользователем системы. При разработке форм выходных документов в ИС должны использоваться термины и сокращения общепринятые в этой предметной области и согласованные с заказчиком системы.

Внутри информационная база представляет собой физически реализованную базу данных. Носителем данных является жесткий диск, на котором находится Система Управления Базами Данных (далее – СУБД). Доступ к данным осуществляется с помощью SQL-запросов к СУБД.

Основные принципы построения информационной базы:

- проектирование таблиц осуществляется с принципами построения и организации реляционных баз данных;
- информационный массив накапливается и находится в реляционной базе данных;
- уменьшение избыточности данных не должно приводить к усложнению доступа и уменьшению скорости обрабатывания информации.

Внутри информационной базы осуществляется контроль целостности данных с помощью процедур, применяемых к элементам базы данных в качестве ограничения целостности.

На этапе ввода происходит сопоставление типов, вносимых данных с типом поля базы данных, а также проверка на допустимые значения.

Хорошо спроектированная база данных:

- удовлетворяет всем требованиям пользователей к ее содержимому. Перед проектированием базы следует провести исследования требований пользователей к ее функционированию;
- удовлетворяет требованиям пользователей к ее производительности. При больших объемах данных вопросы сохранения производительности начинают

играть главную роль, сразу выявляя все без исключения недостатки этапа проектирования;

- гарантирует естественное, простое для восприятия структурирование информации. Качественное построение базы позволяет делать запросы к ней более легкими для понимания, таким образом, уменьшается вероятность внесения некорректных данных и улучшается качество сопровождения базы;

- обеспечивает непротиворечивость и единство данных. При проектировании таблиц необходимо определить их атрибуты и некоторые правила, ограничивающие возможность ввода пользователем неверных значений. Для верификации данных перед непосредственной записью их в таблицу, база данных должна осуществлять вызов правил модели данных и этим обеспечивать сохранение целостности информации.

1.5 План разработки

Назначение системы: автоматизации обработки данных анкетирования сотрудников филиала.

Цель создания системы:

- сокращение временного интервала на обработку информации;
- сокращение временного интервала на мониторинг данных анкетирования за предыдущие года;
- улучшение качества учета обрабатываемой информации;
- сокращение затрат на обработку информации.

Объект автоматизации: филиал ЮУрГУ.

Функции:

- хранение обработанных данных анкетирования;
- доступ к данным анкетирования за любой период времени;
- вывод отчета по обработанным данным;
- защищенность.

1.6 Постановка задачи

Цель выпускной квалификационной работы заключается в разработке ИС по обработке данных анкетирования.

Требования к приложению:

- подсчет количества респондентов (опрошенных);
- подсчет процентного соотношения ответивших на тот или иной вопрос;
- вывод необходимых графиков;
- мониторинг данных за прошлые года;
- формирование и вывод на печать готового отчета;
- интуитивно понятный, графический интерфейс;
- по возможности простота и универсальность применения разрабатываемого алгоритма;
- мобильность.

Отталкиваясь от поставленных задач, которые должны выполняться программой, нами определена модель программирования. Разработка ИС предполагается по принципу императивной модели программирования. В связи с этим проведен ряд следующих работ.

- 1) Анализ существующей ситуации в филиале ЮУрГУ.
- 2) Исследование различных подходов к проектированию пользовательских интерфейсов.

1.7 Анализ существующей ситуации в филиале ЮУрГУ

Обработка данных анкетирования в филиале ЮУрГУ на данный момент происходит в ручном режиме без элементов автоматизации.

Стоит отметить, что анкетирование проходит на бумажных носителях, обрабатывается, а после хранится в архиве и предоставляется по требованию проверяющих органов, если это потребуется.

Обработка анкетирования происходит в несколько этапов. Сначала происходит сбор бумажных носителей, после просматривается каждая анкета и происходит подсчет ответов и респондентов. После обработки на бумажном носителе все переносится в программу Microsoft Excel для дальнейшего построения необходимых графиков и подсчета процентного соотношения, ответивших на тот или иной вопрос. Далее формируется отчет по филиалу. В связи с этим, обработка данных анкетирования может занимать до нескольких дней.

Если же требуется провести мониторинг данных анкетирования за определенный год, то для этого приходится искать файл Excel или бумажный носитель и сверять по нему. Данное действие происходит не часто, но если происходит, то занимает значительное время от времени обработки.

Подводя все выше сказанное к одному, можно сказать, что раз операция по обработке анкетирования сотрудников занимает большое количество времени, которое можно потратить на осуществление других трудовых функций, необходимость автоматизации данного процесса очевидна.

1.8 Обзор существующих подходов к проектированию пользовательских интерфейсов

Пользовательский интерфейс обеспечивает взаимодействие между пользователем и компьютером, обмен действиями и ответными реакциями на них.

Юзабилити – степень, с которой продукт может быть использован определенными пользователями при определенном контексте использования для достижения определенных целей с должной эффективностью, продуктивностью, удовлетворенностью. [2]

Исследования в сфере взаимодействия пользователя и компьютера демонстрируют, то что каждый пользовательский интерфейс обязан выполнять следующие четыре функций:

- отображение данных от пользователя;
- ввод данных от пользователя, обратная реакция системы;
- управление компьютером только путем пользовательских действий: прерывание, инициация, отмена компьютерных процессов;
- поддержка пользователя в процессе деятельности, что включает в себя обратную связь и сбор информации об ошибочных либо случайных действиях пользователя. [3]

Хорошо спроектированный пользовательский интерфейс должен соответствовать следующим принципам:

- обладать низким порогом вхождения, полностью способствовать быстрому изучению интерфейса пользователем, формировать устойчивые навыки;
- предоставлять ввод информации естественным образом, не демонстрируя процесс вычислений;
- соответствовать требованиям рабочих потребностей пользователя, не заостряя его внимание на процессе обработки данных.

Для получения эффективного результата разработки пользовательского интерфейса используют различные подходы к проектированию.

Основные подходы к проектированию:

- целеориентированный подход (Goal Centered Design). В основе данного метода лежат конечные цели пользователей, которые должны быть ими достигнуты с помощью взаимодействия с программным продуктом;
- подход, направленный на пользователя (User-Centered Design), который характеризуется активным вовлечением пользователей в процесс проектирования и тестирования программного продукта, четким пониманием пользовательских требований и задач, оптимальным распределением функций между пользователями и технологиями, интерактивностью и мультидисциплинарностью подхода. Использование этого подхода при разработке пользовательского интерфейса для достижения высоких показателей в области юзабилити (в соответствии с ISO 9241-11, это степень эффективности, продуктивности и

удовлетворенности, с которой продукт может использоваться определенными пользователями для достижения определенных). Так же приводит к уменьшению затрат на разработку и повышению эффективности продукта, как в отношении бизнеса, так и в удовлетворенности пользователей (повышение лояльности к продукту и разработчику);

– подход, направленный на деятельность (Activity-Centered Design). Согласно определению Дональда Нормана (Donald Norman), деятельность содержит задачи, которые состоят из действий, в свою очередь составленных из операций. Подход к проектированию интерфейсов, предлагаемый Норманом, уделяет внимание, прежде всего, пониманию деятельности пользователя. Он утверждает, что человек адаптируется к существующим инструментам и что понимание деятельности, выполняемой человеком при помощи инструментов, способно благоприятно влиять на интерфейс этих инструментов. [3]

При разработке интерфейса рационально гибко пользоваться существующими подходами, учитывая при выборе методов назначение разрабатываемого продукта, целевую аудиторию, время и бюджет разработки.

Интерфейс должен быть простым и понятным с первого взгляда, у пользователя не должно возникать ни каких проблем и вопросов, как взаимодействовать с приложением для достижения собственных целей. Помимо этого, число шагов, которые сделает пользователь для достижения цели, должно быть минимальным. У пользователя, работающего с хорошо спроектированным интерфейсом, не возникнет проблем и вопросов, и он не сможет совершить ошибку, производя то или иное действие, пользователь должен понимать на интуитивном уровне, какие изменения или ошибки это повлечет.

Одним из главных правил проектирования интерфейса считается наличие обратной связи с пользователем, то есть уведомление его о работе приложения, о том, что оно отреагировало на какой-либо запрос. Возможность показа анимацией процесс загрузки страницы/изображения/экрана, оповещение пользователя о тех или иных изменениях с помощью всплывающих окон.

1.9 Критерии качества программы

Оценка качества программы является весьма трудной задачей из-за разнообразия интересов пользователей. По этой причине нельзя предложить одну универсальную меру качества и приходится применять ряд характеристик, охватывающих весь спектр предъявляемых требований. Совокупность некоторых критериев определяет показатель качества, формируемый исходя из требований, предъявляемых к программному обеспечению.

Вначале формируются характеристики качества, в числе которых могут быть общая полезность и удобство эксплуатации.

Далее формируются показатели, к числу которых могут быть причислены:

- 1) Целостность – регулирование доступа, контроль доступа.
- 2) Эффективность – эффективность функционирования.
- 3) Практичность – работоспособность, удобство интерфейса.
- 4) Надежность – устойчивость к ошибкам, простота.
- 5) Модифицируемость – программа должна быть сконструирована в виде отдельных модулей, что позволяет с легкостью вводить новые функции.
- 6) Гибкость – общность, предназначение программы для определенных пользователей.
- 7) Удобство обслуживания – простота, краткость, информативность. В программе должен быть интуитивно понятный интерфейс, присутствовать меню.

Выводы по разделу один:

При разработке программного обеспечения значимым фактором является точная формулировка цели проекта и анализ предметной области.

В данном разделе рассмотрены и проанализированы подходы проектирования пользовательских интерфейсов, проблемы существующие при обработке

анкетирования в филиале, средства разработки, а также сформирована необходимость автоматизации обработки данных анкетирования.

Для решения обозначенных проблем было принято решение о разработке ИС. Необходимым условием при создании системы является знание предметной области и анализ, который проведен в данном разделе.

2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

Разработка программного продукта включает в себя 4 этапа:

- 1) Обоснование выбора средств разработки для реализации системы.
- 2) Обоснование выбора СУБД.
- 3) Реализация СУБД.
- 4) Проектирование структуры и интерфейса программы.

2.1 Выбор средств разработки для реализации системы

2.1.1 Обоснование выбора средств разработки

Для создания программ под Windows существует огромное число интегрированных сред разработки. К таким возможно отнести: Visual Basic, Visual C++, Delphi, C++ Builder.

С появлением средств быстрой разработки приложений (RAD – rapid application development) появилась возможность программировать с помощью готовых компонентов и шаблонов. Сопоставим Delphi и C++ Builder, выберем среду программирования для автоматизированной системы.

Система визуального программирования Delphi, разработана компанией Borland International на базе языка Object Pascal. Объектно-ориентированный подход к созданию компонент был серьезным шагом вперед. Дополнительный выигрыш обеспечивался за счет нормальной компиляции, обеспечивающей получение наиболее производительных программ. Первые две версии Delphi достаточно стремительно добились расположения не только у вузовских аудиторий, где Pascal пользовался особым уважением, но и среди профессионалов. Это подтолкнуло одно из подразделений фирмы Borland International на перенос визуальной технологии в среду C++. Таким образом, практически одновременно с

появлением Delphi 2.0 на рынке появилась первая версия Borland C++ Builder (далее – BCB).

Основу первой версии BCB составила библиотека визуальных компонент VCL (Visual Component Library), перенесенная без изменений из Delphi 2. Интерфейсы сред Delphi и BCB похожи друг на друга, безусловно и значительная часть BCB была разработана на языке Object Pascal в среде Delphi. Вследствие своего происхождения система BCB оказалась двуязычной. Помимо собственного основного языка программирования она позволяет практически без каких-либо доработок использовать формы, объекты и модули, разработанные в среде Delphi. Для того чтобы еще больше расширить сферу влияния среды BCB, ее авторы в последующих версиях обеспечили возможность использования библиотеки классов MFC (Microsoft Foundation Classes), разработанной фирмой Microsoft.

Delphi 7 – это удобный инструмент, однако в то же время и непростая программная среда, состоящая из многочисленных элементов. Включает в себя новый интерфейс Galileo, а кроме того interbase server и desktop, remote debugger server, Model Maker, Install Shield.

Особенно удобными в Delphi считаются такие начальные возможности, как объектно-ориентированный подход к программированию, базирующийся на формах, ее высоко производительный компилятор, замечательная поддержка баз данных, тесная интеграция с программированием под Windows и технология компонентов. Однако наиболее значимой составляющей считается язык Object Pascal, на фундаменте которого строится все остальное.

Delphi 7 обладает открытой архитектурой, целиком поддерживает технологии Microsoft OLE Automation, ActiveX, ODBC. Компилятор позволяет иметь доступ ко всем ресурсам операционных систем, реализующих интерфейс Win32 (Windows XP и Windows 7).

Программы Delphi используют объектно-ориентированную структуру под названием VCL – Visual Component Library (библиотека визуальных компонентов). Непосредственно VCL поднимает быструю разработку приложений на новый

уровень. Можно увеличить свои способности за счет создания своих собственных компонентов. К тому же независимые поставщики уже создали множество компонентов такого рода.

Delphi 7 обладает большим количеством улучшений IDE, расширенную поддержку баз данных (по специальным наборам данных ADO и InterBase), усовершенствованную версию MIDAS с поддержкой Сети интернет, инструмент управления версиями TeamSours, возможности перевода, концепцию фреймов и огромное число новых компонентов.

В основе идеологии Delphi возлежат технологии визуального проектирования и программирование процедур обработки событий, использование которых дает возможность существенно сократить время разработки и облегчить процесс создания приложений.

C++ Builder 6 – очередная версия системы объектно-ориентированного программирования для операционных систем Windows 2000, Windows XP, Windows Vista и Windows 7. Интегрированная среда системы (Integrated Development Environment, IDE) гарантирует ускорение визуального проектирования, а кроме того продуктивность многократно используемых компонентов в сочетании с улучшенными инструментами и разномасштабными средствами доступа к базе данных.

Стандарты пользовательских интерфейсов изменяются и развиваются так же стремительно, как и операционные системы. Открытость среды IDE позволяет настраивать ее с учетом наиболее модных тенденций в области графических интерфейсов. Визуальный интерфейс сочетает в себе простоту использования для начинающего и богатство возможностей для профессионала. [3]

Система C++ Builder 6 может быть использована там, где необходимо дополнить существующие приложения (как прикладные, так и системные) расширенным стандартом языка C++, увеличить быстродействие и надежность программ, придать пользовательскому интерфейсу качество профессионального уровня, позволяет быстро создавать использующие сенсорный ввод данных

графические интерфейсы и приложения для КПК, сенсорных панелей и автономных общедоступных систем и усовершенствовать имеющиеся приложения с минимальным добавлением кода или без него.

В контексте C++ Builder предполагает не только реальное ускорение стандартного цикла «редактирование – компиляция – компоновка – прогон – отладка», но и придает создаваемым проектам изящество компонентной модели.

Уникальная среда разработки C++ Builder 6 2010 IDE Insight дает возможность прибегать ко всем возможностям, параметрам и компонентам интегрированной среды разработки, не тратя время на их поиск в меню и диалоговых окнах; обозреватель классов, обеспечивающий управление классами в проекте и быстрый переход между ними; объединяет Дизайнер форм, Инспектор объектов, Палитру компонентов, Менеджер проектов и полностью интегрированные Редактор кода и Отладчик.

В C++ Builder 6 добавлены поддерживаемые отладчиком средства визуализации данных, упрощающие отладку, позволяя настраивать отображение типов данных в отладчике; поддерживаемые отладчиком средства управления потоками, обеспечивающие заморозку, разморозку и изоляцию потоков, а кроме того установку контрольных точек для выбранных потоков, что упрощает разрешение проблем; C++ Builder 6 включает новые параметры отладчика: Scroll new events into view («Прокрутка новых событий в представлении») и Ignore non-user breakpoints («Игнорирование не пользовательских контрольных точек»), как на уровне исходных инструкций, так и на уровне ассемблерных команд – в расчете угодить высоким требованиям программистов-профессионалов. [4]

Оптимизирующий 32-разрядный компилятор построен согласно уникальной и проверенной адаптивной технологии, обеспечивающей только прочную и скорую оптимизацию, равно как объема выходного исполняемого кода, так и требуемой памяти.

Визуальная разработка методом «перетаскивания» (drag-and-drop) многократно упрощает и ускоряет кропотливый процесс программирования

приложений СУБД. В основе объектно-ориентированного взаимодействия клиент – сервер лежит понятие наборов данных (dataset) – таблиц, запросов, хранимых процедур – основных сущностей баз данных, которыми оперируют компоненты доступа. Обширный подбор компонентов визуализации и редактирования позволяет легко изменять вид представления наборов данных. С++ Builder использует проводник баз данных и масштабируемый словарь данных для того, чтобы автоматически настроить средства отображения и редактирования применительно к специфике вашей информации.

Наравне с дизайнерами и художниками-оформителями к прикладным разработкам в ключевых областях сети все больше привлекаются профессионалы-программисты. Качественное приложение способно динамически выбирать информацию с сервера и предоставлять ее в формате, подходящем для потребителей разного уровня, так, чтобы они смогли сформировать свое заключение и принять адекватное решение в кратчайший срок. С++ Builder 6 полностью удовлетворяет данным требованиям, обеспечивая:

- значительную надежность, степень интеграции и качество управления;
- быстрое визуальное проектирование эффективных приложений для переработки крупных объемов информации;
- поддержку механизмов принятия решения и доступа к удаленным базам данных.

С++ Builder предоставляет свою мощь и широкие возможности языка С++ всему семейству систем объектно-ориентированного программирования. Система С++ Builder может быть применена повсюду, где необходимо дополнить имеющиеся приложения расширенным промышленным стандартом языка С++, увеличить быстродействие и внести пользовательскому интерфейсу профессиональный облик.

Все без исключения компоненты, формы и модули данных, которые накопили программисты, работающие в Delphi, могут быть многократно применены в приложениях С++ Builder без каких-либо изменений. С++ Builder безупречно

сгодится тем разработчикам, которые предпочитают выразительную мощность языка C++, однако хотят сохранить продуктивность Delphi. Неповторимая взаимосвязь этих систем программирования дает возможность при создании приложения без труда переходить из одной среды разработки в другую.

Проведем подбор среды программирования способом экспертного оценивания. В состав экспертной группы вошли специалисты инженерно-технического направления.

Выделим критерии оценки среды программирования. Значимость каждого из представленных критериев была оценена экспертами по 100 бальной шкале. Отталкиваясь из полученных данных, находится средний балл и коэффициент относительной значимости критерия.

Коэффициент относительной важности рассчитывается по формуле:

$$K = \frac{\text{Средний балл}}{\Sigma} \cdot 100\% , \quad (1)$$

где *Средний балл* – сумма усреднённых экспертных оценок первого этапа экспертизы вычисляется из суммы экспертных оценок;

Σ – сумма экспертных оценок первого этапа экспертизы.

Результаты экспертизы представлены в таблицах 2.1–2.2.

Таблица 2.1 – Результаты экспертизы сред разработки, первый этап

Функция	Эксперт 1	Эксперт 2	Эксперт 3	Средний балл по 100 бальной шкале	Коэффициент относительной важности
Стоимость	82	92	75	83	13,6
Простота сопровождения	84	78	76	79	13
Временные затраты на разработку	92	83	97	91	15
Быстродействие	89	90	89	89	14,6
Удобный дизайн	87	80	91	86	14,1
Мощность пакета	77	93	85	85	14

Продолжение таблицы 2.1

Функция	Эксперт 1	Эксперт 2	Эксперт 3	Средний балл по 100 бальной шкале	Коэффициент относительной важности
Возможности языка	91	100	98	96	15,7
Сумма				609	100,0%

Таблица 2.2 – Результаты экспертизы сред разработки, второй этап

Функция	Коэффициент относительной важности	Среда программирования	
		Delphi	C++ Builder
Стоимость	13,6	+	+
Простота сопровождения	13	+	-
Временные затраты на разработку	15	+	-
Быстродействие	14,6	+	+
Удобный дизайн	14,1	+	+
Мощность пакета	14	+	+
Возможности языка	15,7	+	+
Сумма	100%	100%	72%

Принимая во внимание все вышесказанное и результаты анализа экспертным оцениванием можно сделать выбор среды программной разработки в пользу Delphi, который гарантирует чрезвычайно высокую производительность и удобство использования.

2.1.2 Обоснование выбора СУБД

Определим СУБД, которая будет применяться для разработки.

Сегодня на рынке программных продуктов представлено большое число различных СУБД. Несмотря на то, что СУБД могут обладать некоторыми характерными отличительными чертами, большинство из них полагаются на единый устоявшийся комплекс основных функций.

Рассмотрим, сравним и выберем СУБД методом экспертной оценки.

Осуществим выбор СУБД исходя из следующего перечня: InterBase 7 SQL Server, Microsoft Access 2010.

InterBase 7 SQL Server был разработан и продавался фирмой InterBase Software Corporation (ISC). Сотрудник DhC James Starkey, разработавший DSRI для Rdb, хотел расширить возможности Rdb, однако его предложения были отвергнуты DEC. Поэтому он создал собственную компанию, разработавшую свою RDBMS, первоначальное название которой было Groton Database System (GDS) [5].

Во время существования ISC дистрибуцией IB (под названием StarBase) занималась фирма Cognos Inc, и вплоть до настоящего времени являющаяся одной из основных фирм, оказывающей технические консультации и сопровождение по InterBase. Позже фирма ISC была приобретена компанией Ashton-Tate (в этот момент James Starkey ушел в Harbor Software), и перешла в Borland при приобретении последним Ashton-Tate.

При создании и развитии InterBase SQL Server было использовано большое число нестандартных решений и новейших технологий.

Расширяемость SQL-сервера не за счет увеличения числа нестандартных интегрированных функций, или реализации сложного встроенного языка программирования, а при помощи внешних модулей, создаваемых на компилируемых языках.

SQL-сервер IB Database построен на архитектуре множественных поколений записей (Multi-Generational Architecture – MGA). Данная архитектура использует уникальный версионизирующий механизм, который обладает значительной производительностью при обработке коротких транзакций и транзакций принятия решений. Обычно серверы баз данных поддерживают модель On-Line Transaction Processing (OL TP), характеризующуюся большим числом коротких, одиночных транзакций. В то время как IB Database поддерживает такой режим, дополнительно поддерживаются продолжительные транзакции поддержки принятия решений.

Механизм версионирования позволяет транзакциям избавиться от лишних блокировок, используемых данных, и применяемый принцип чтение данных не приводит к блокировке их изменения. В отличие от других баз данных, не блокирующие транзакции IBM Database не требуют дополнительного программирования, чтобы обеспечить постоянный, воспроизводимый результат для любого запроса. Таким образом, механизм версионирования позволяет сосуществовать коротким и длинным транзакциям и гарантирует максимальную эффективность для обоих.

В настоящее время одной из наиболее распространенных и новых СУБД является Microsoft Access 2010, которая предоставляет большие возможности и удобство, это приложение позволит эффективно работать даже в случае, если вы не являетесь серьезным специалистом по работе с базами данных. Благодаря большому числу грамотно разработанных шаблонов, работа в Microsoft Access 2010 удобна в использовании даже при большом объеме. СУБД Microsoft Access 2010 позволяет создавать базы данных различного объема, с которыми функционируют в монопольном режиме или режиме коллективного доступа. Абсолютно новое представление Microsoft Office Backstage позволяет с максимальным результатом управлять базами данных при подходящем доступе и результативно управлять ходом работы.

Обновленный конструктор макросов теперь упрощает значительно разработку непростых логических выражений и делает это интуитивно понятным для любого квалифицированного разработчика, позволяет очень легко и быстро объединять данные и повысить качество работы. Теперь непосредственно в разрабатываемые проекты можно подключать данные web-служб, а кроме того SharePoint 2010 Business Connectivity Services, появляется возможность для импорта данных из разных внешних источников (Excel, Outlook, SQL Server и других приложений пакета программ Microsoft) и формирования связи с ними. При сборе либо отправке данных по электронной почте сервер теперь не потребуется.

Методы доступа к базам данных теперь выходят на качественно новый уровень, вследствие интегрированию в Microsoft Access 2010 дополнения SharePoint Server, что отныне позволит размещать базы данных при помощи новой технологии web-баз данных. Данная функция будет весьма полезна для тех, кто занимается web-разработками, так как с помощью этого приложения опубликовать базу данных на web-сайте не составит большого труда. Теперь комфортнее станет использовать базы данных совместно. Можно будет к ним не только обращаться, но и вносить коррективы напрямую посредством сети Интернет. А пользователи, у которых не установлено приложение Microsoft Access 2010, все равно смогут открыть web-формы или отчеты через браузер. При этом вносимые изменения синхронизируются автоматически. Microsoft Access 2010 обладает характерным для всех приложений Microsoft Windows удобным графическим интерфейсом, ориентированным на удобную работу пользователя.

Принципиально новыми функциями СУБД Microsoft Access 2010 являются:

- сжатие базы данных;
- преобразование базы данных в формат Microsoft Access 2007/2010;
- анализ быстродействия базы данных;
- импортирование объекта в свою базу данных;
- сохранение базы данных в виде accde-файла;
- анализ данных в Microsoft Excel;
- повышение быстродействия Microsoft Access 2010;
- просмотр и изменение свойств документа Microsoft Access 2010 и др.

Следует отметить критерии оценки СУБД. Значимость каждого из представленных критериев была оценена экспертами по 100 бальной шкале. Исходя из полученных данных, находится средний балл и коэффициент сравнительной важности критерия. Итоги первого этапа экспертизы представлены в таблице 2.3.

Таблица 2.3 – Сравнение СУБД, первый этап

Наименование критерия	Эксперт 1	Эксперт 2	Эксперт 3	Средний балл по 100 бальной шкале	Коэффициент относительной важности
Стоимость СУБД	87	92	77	85	13,4
Операционная система	73	84	82	80	12,6
Простой интерфейс	92	97	96	95	15
Широта распространения	87	92	78	86	13,6
Качество справочной информации к системе	93	86	87	89	14
Количество пользователей	100	100	100	100	15,8
Простота использования	100	97	99	99	15,6
Сумма				634	100,0%

Результаты второго этапа экспертизы представлены в таблице 2.4.

Таблица 2.4 – Результаты второго этапа экспертизы

Функция	Коэффициент относительной важности	InterBase	Microsoft Access
Стоимость СУБД	13,4	-	+
Операционная система	12,6	+	+
Простой интерфейс	15	+	+
Широта распространения	13,6	+	+
Качество справочной информации к системе	14	+	+
Количество пользователей	15,8	+	+
Простота использования	15,6	-	+
Сумма	100,0%	71%	100%

Из всего вышеперечисленного и принимая во внимание анализ СУБД методом экспертного оценивания, подходящим средством проектирования базы данных выбрано Microsoft Access 2010. Interbase по методу экспертных оценок и по объективным показателям программирования не подходит. Поэтому

однозначно выбор стоит за Microsoft Access 2010. Microsoft Access 2010 же удовлетворяет абсолютно всем потребностям проектирования физической модели данных, по этой причине выбираем эту СУБД.

2.2 Техническое задание

2.2.1 Требования к системе в целом

2.2.1.1 Требования к структуре и функционированию системы

Данное техническое задание разрабатывается для ИС, нацеленную на автоматизацию обработки данных анкетирования в филиале ЮУрГУ.

Приложение, разрабатываемое в рамках данной выпускной квалификационной работы, состоит из двух окон, а именно из окна «меню» для выбора и окна для заполнения данных.

Окно «меню» состоит из трех кнопок, а именно: кнопки для открытия окна заполнения данных анкетирования, обработки результатов и кнопки закрытия «меню».

Второе окно используется для ввода количества ответов, количества респондентов и года анкетирования.

Интерфейс программы должен быть интуитивно понятным. Информация подается в удобном для пользователя виде таблиц.

2.2.1.2 Показатели назначения

- максимальное время обработки одной анкеты не должно превышать 15 минут, а на полную обработку не более 1 рабочего дня;
- функция вывода данных (формирования отчетов) не должна превышать пяти минут для одного отчета.

2.2.1.3 Требования по сохранности информации при авариях

Безопасность информации при непредвиденных обстоятельствах необходимо обеспечивать путём резервного копирования базы данных. Резервное копирование данных должно осуществляться автоматически ежедневно или в любой момент времени работником системы по его инициативе.

2.2.1.4 Требования по стандартизации и унификации

При создании системы необходимо использовать среду разработки Borland Delphi 7.

Экранные формы системы должны быть разработаны в соответствии с требованиями к интерфейсу программных средств ГОСТ Р ИСО 14915-1-2010 «Эргономика мультимедийных пользовательских интерфейсов». [6]

Общая оценка программного продукта должна быть проведена в соответствии с ГОСТ 28195-89 «Оценка качества программных средств». В том числе программный продукт должен быть оценен по следующим критериям [7]:

- надежность;
- сопровождаемость;
- удобство применения;
- эффективность;
- универсальность;
- корректность.

Осуществление действий по вводу данных и получению отчетов должно осуществляться в соответствии с должностями инструкциями пользователей системы.

2.2.1.5 Требования к качеству и форме реализации каждой функции или задачи

В разрабатываемых функциях необходимо предусмотреть выполнение следующих условий:

- выходная информация должна отображаться на экран и сохраняться во внешние файлы в формате MS Excel для дальнейшей распечатки отчета, а также во внешние файлы в текстовом формате;

- входная информация в систему должна поступать только путем ручного ввода в определенные поля. Необходимо осуществить контроль вводимой информации. Так поля, в которых предусматривается ввод цифр, должны быть только целыми числами, не превышающими установленный. Поле для ввода даты должно предусматривать выбор необходимого значения из календаря. Это позволит сократить число ошибок при вводе информации.

2.2.2 Требования к видам обеспечения

2.2.2.1 Требования к программному обеспечению системы

Для работы с информационной системой компьютер пользователя должен быть оснащен ОС Microsoft Windows 7 и выше.

2.2.2.2 Требования к техническому обеспечению системы

Для нормального функционирования ИС компьютер конечного пользователя должен удовлетворять техническим требованиям, предъявляемым к аппаратному обеспечению, а именно:

- процессор: Intel Celeron Dual-Core 2000Mhz;
- оперативная память: не менее 1024 Мб;
- жесткий диск: не менее 160 Гб;

- VGA видеоадаптер;
- монитор LCD;
- клавиатура, мышь.

2.3 Реализация СУБД на Microsoft Access

В СУБД были созданы таблицы базы данных (Рисунок 2.1 – 2.4):

- «Answer»;
- «Question»;
- «Results»;
- «YearResp».

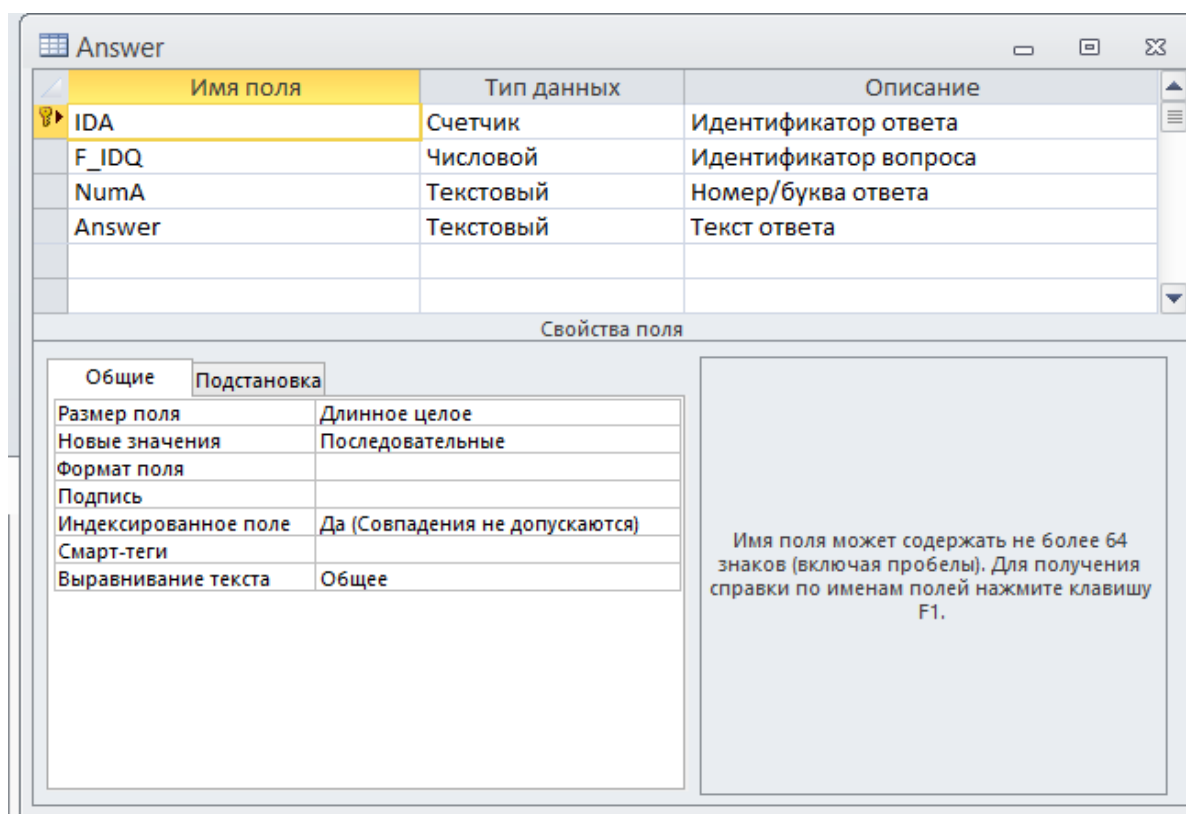


Рисунок 2.1 – Таблица «Answer»

Имя поля	Тип данных	Описание
IDQ	Счетчик	Идентификатор вопроса
NumQ	Текстовый	Номер/буква вопроса
Question	Текстовый	Текст вопроса

Свойства поля

Общие	
Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	
Индексированное поле	Да (Совпадения не допускаются)
Смарт-теги	
Выравнивание текста	Общее

Имя поля может содержать не более 64 знаков (включая пробелы). Для получения справки по именам полей нажмите клавишу F1.

Рисунок 2.2 – Таблица «Question»

Имя поля	Тип данных	Описание
AnkYear	Числовой	Год анкетирования
AllResp	Числовой	Количество респондентов
AllWorkers	Числовой	Количество сотрудников
DateBegin	Дата/время	Дата начала тестирования
DateEnd	Дата/время	Дата окончания тестирования

Свойства поля

Общие	
Размер поля	Длинное целое
Формат поля	
Число десятичных знаков	Авто
Маска ввода	
Подпись	
Значение по умолчанию	0
Условие на значение	
Сообщение об ошибке	
Обязательное поле	Нет
Индексированное поле	Да (Совпадения не допускаются)
Смарт-теги	
Выравнивание текста	Общее

Имя поля может содержать не более 64 знаков (включая пробелы). Для получения справки по именам полей нажмите клавишу F1.

Рисунок 2.3 – Таблица «YearResp»

Имя поля	Тип данных	Описание
IDR	Счетчик	Идентификатор результата
AnkYear	Числовой	Год анкетирования
IDA	Числовой	Идентификатор ответа
CountResp	Числовой	Количество респондентов, выбравших ответ
PercResp	Числовой	Процент выбравших ответ от общего числа респондентов

Свойства поля	
Общие	Подстановка
Размер поля	Длинное целое
Новые значения	Последовательные
Формат поля	
Подпись	
Индексированное поле	Да (Совпадения не допускаются)
Смарт-теги	
Выравнивание текста	Общее

Имя поля может содержать не более 64 знаков (включая пробелы). Для получения справки по именам полей нажмите клавишу F1.

Рисунок 2.4 – Таблица «Results»

После создания таблиц была реализована схема данных (Рисунок 2.5).

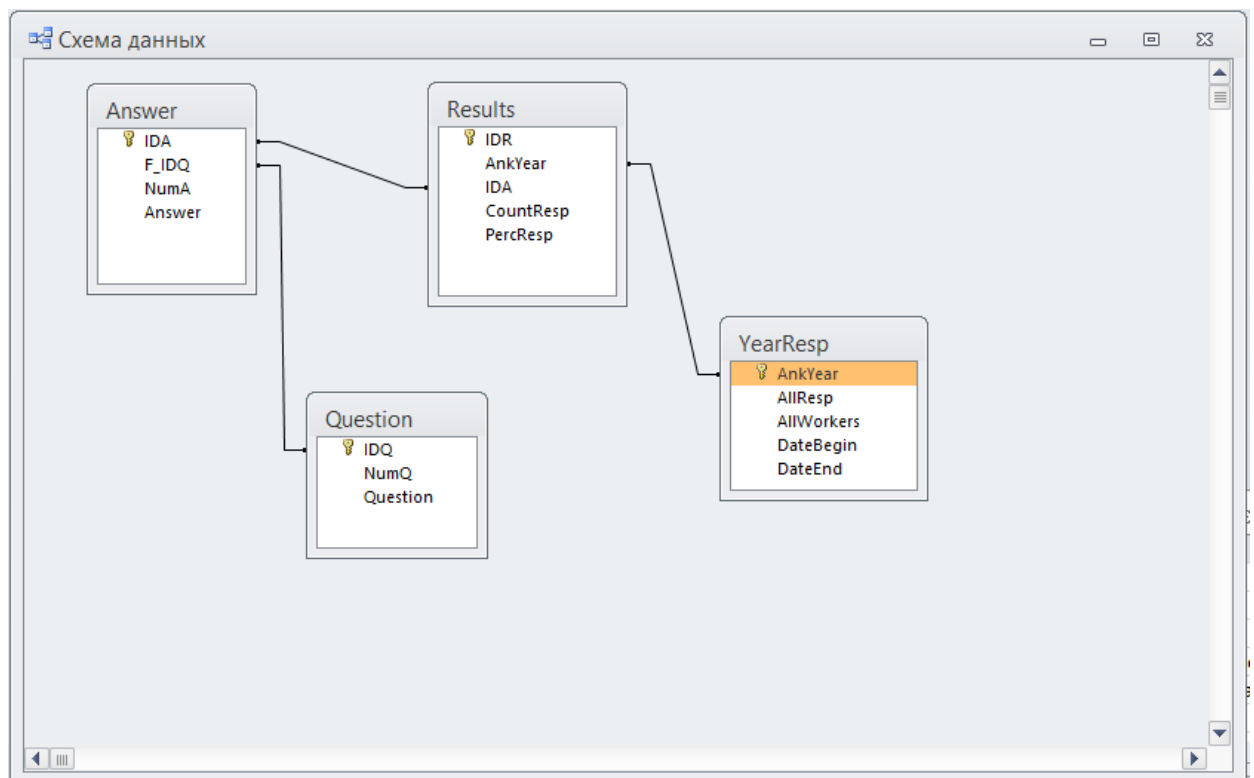


Рисунок 2.5 – Структурная схема данных базы данных

2.4 Реализация информационной системы

Формальной постановке задачи соответствует контекстная диаграмма методологии IDEF0 (Рисунок 2.5). В данной методологии входными данными определены данные анкетирования, выходными потоками – отчет, входные управления – анкеты сотрудников, а входными ресурсами ПК (компьютеры, с помощью которых происходит обработка данных анкетирования) и специалист по кадрам (специалисты, выполняющие назначенные руководством указания).

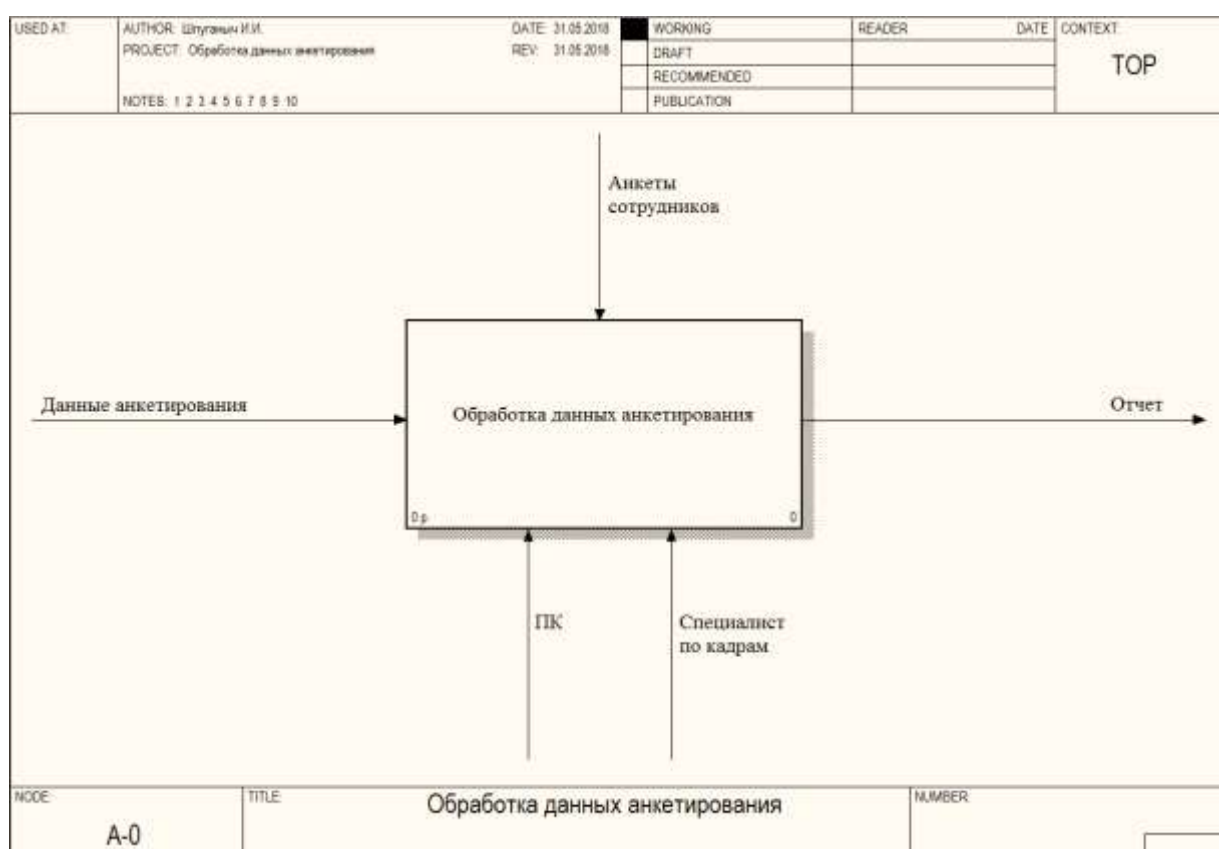


Рисунок 2.5 – Диаграмма IDEF0

Для решения поставленной задачи необходимо провести следующие операции:

- ввод информации в базу данных;
- формирование результатов анкетирования;
- составление отчета.

Декомпозиция процесса обработки данных анкетирования на рисунке 2.6.

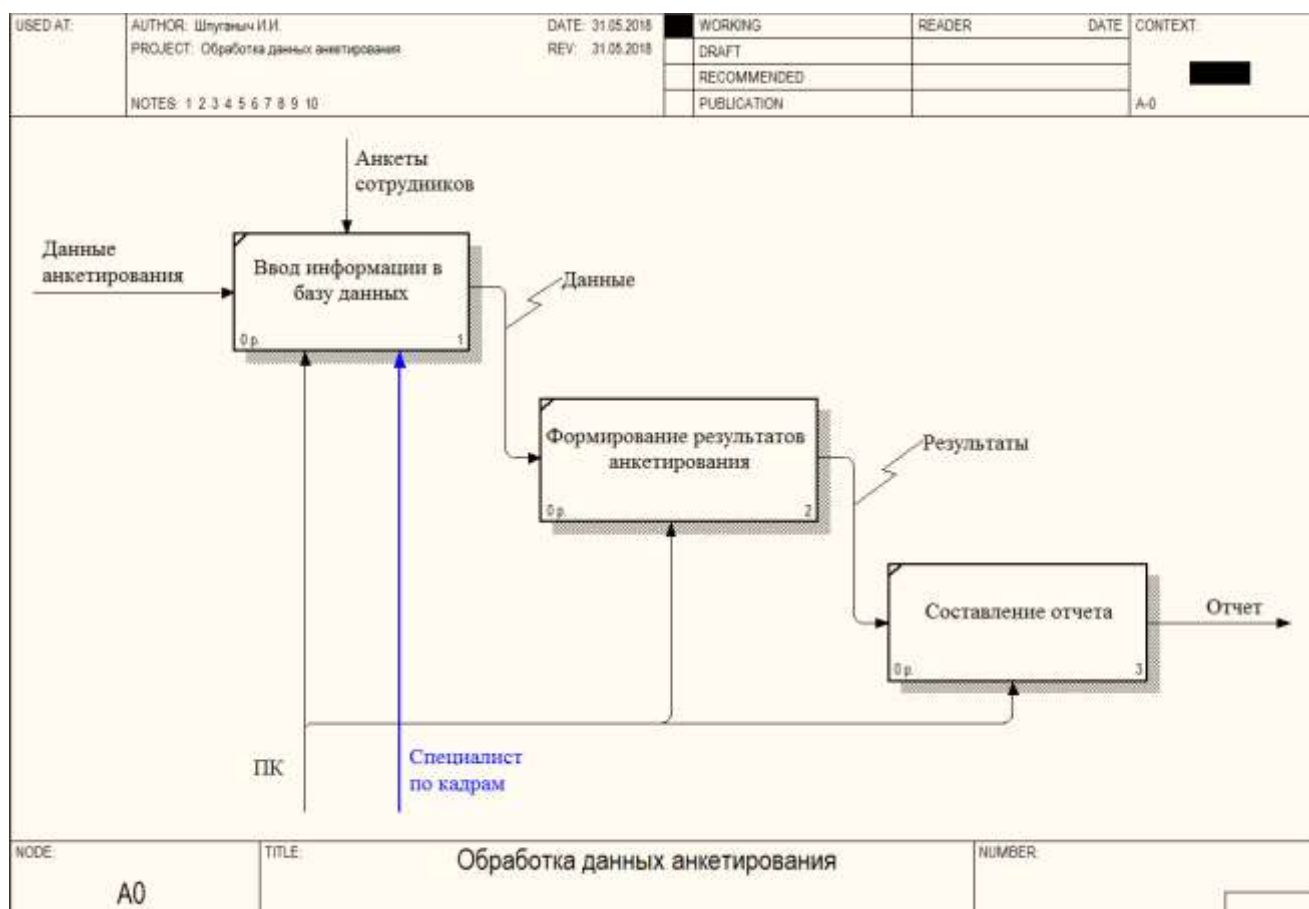


Рисунок 2.6 – Декомпозиция процесса обработки данных анкетирования

Беря во внимание все вышеперечисленное, построена блок-схема приложения (Рисунок 2.7).

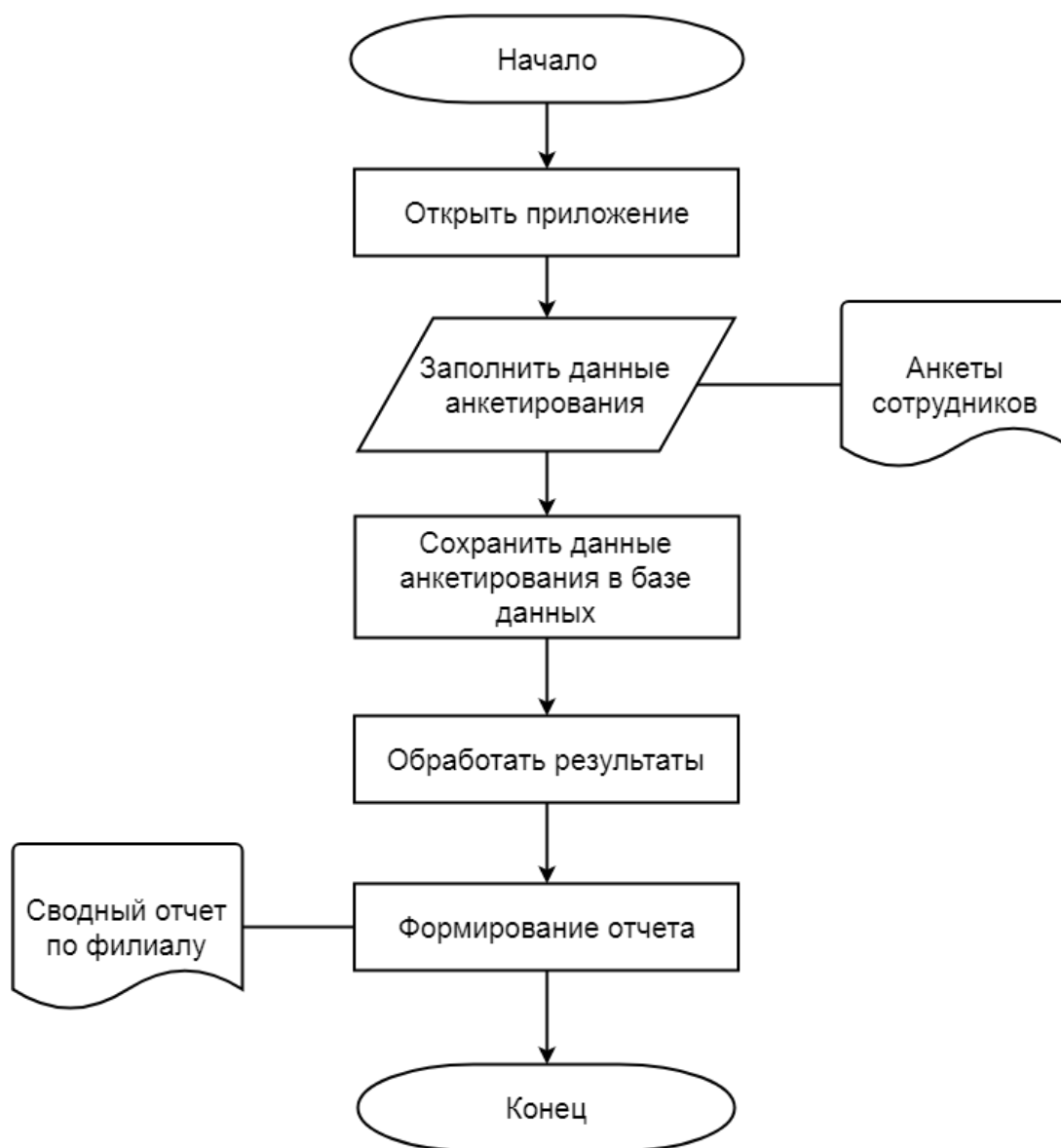


Рисунок 2.7 – Блок-схема приложения

2.5 Описание программы

Для работы базы данных с Borland Delphi 7 использовалась технология доступа к данным ADO (Active X Data Objects). Подключение базы данных осуществлялась с помощью механизма привязок (ADODConnection).

Запустив приложение, первым делом открывается форма «меню» с выбором пункта, который требуется на данный момент времени.

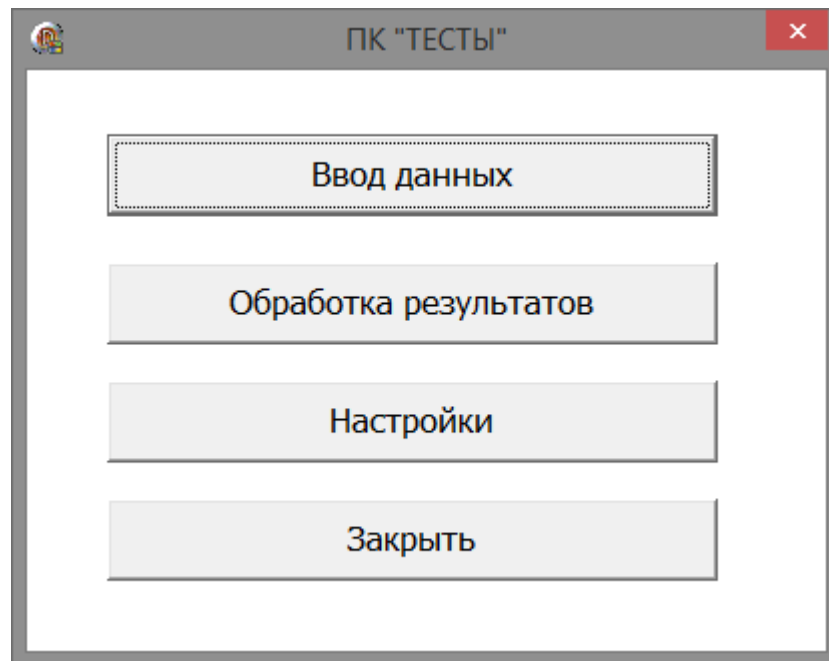


Рисунок 2.8 – «Меню» программы

По нажатию на кнопку «Ввод данных» происходит открытие формы для заполнения данных анкетирования (Рисунок 2.9).

```
procedure TFormMenu.Button1Click(Sender: TObject);  
begin  
    FMainForm.MIInputInfo.Click;  
    ModalResult := mrOk;  
end;
```

Рисунок 2.9 – Ввод данных

Загрузка вопросов и ответов из базы данных на форму происходит через процедуру FormCreate.

```

procedure TForm1.FormCreate(Sender: TObject);
var
  i, j : Integer;
begin
  DecimalSeparator := '.';
  ADOC_InputInfo.Connected := False; //Подключение к базе данных
  ADOC_InputInfo.ConnectionString:='Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' + ExtractFilePath(paramstr(0)) + 'Results.mdb';
  Persist Security Info=False';
  ADOC_InputInfo.Connected := True;
  ADOQ_InputInfo.Active := True;
  ADOQ_InputInfo.First;
  i := 1;
  j := 1;
  While not ADOQ_InputInfo.Eof do
    begin

```

```

if (ADOQ_InputInfo.FieldName('NumA').AsString = '1')
    or (ADOQ_InputInfo.FieldName('NumA').AsString = '')
then
begin
    Questions[i] := TQuestion.Create(Self.SB_Anketa);
    Questions[i].LQuestion.Caption := '          ' +
ADOQ_InputInfo.FieldName('NumQ').AsString +
ADOQ_InputInfo.FieldName('Question').AsString; //Вывод вопросов
    Questions[i].PHeader.Visible := True;
    i := i + 1;
end;
if ADOQ_InputInfo.FieldName('NumA').AsString <> '' then
begin
    Answers[j] := TAnswer.Create(Self.SB_Anketa);
    Answers[j].LAnswer.Caption := '          + ' +
ADOQ_InputInfo.FieldName('Answer').AsString; //Вывод ответов
    Answers[j].MainPanel.Visible := True;
    Answers[j].IDA:=ADOQ_InputInfo.FieldName('IDA').AsInteger;
    j := j + 1;
end;
ADOQ_InputInfo.Next;
end;
ADOQ_InputInfo.Active := False;
end;

```

Для сохранения результатов в базе данных после ввода данных анкетирования, следует нажать на «сохранить данные» на панели с кнопками (Рисунок 2.10).

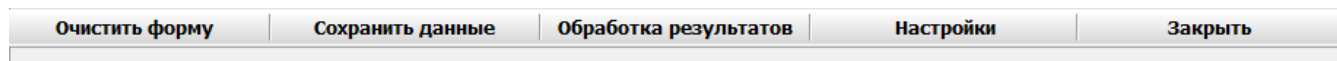


Рисунок 2.10 – Панель с кнопками

Сохранение происходит через процедуру `tbSaveClick`. Если введены данные анкетирования за год, которого нет в базе данных, сохранение происходит следующим образом:

```
begin
  ADOQ_SaveAnswers.Active := False;
  ADOQ_SaveAnswers.SQL.Clear;
  ADOQ_SaveAnswers.SQL.Text := 'SELECT count(*) as RowCount FROM
YearResp WHERE AnkYear = '
+ FormatDateTime('YYYY', DTP_DateBegin.DateTime);
  ADOQ_SaveAnswers.Active := True;
  if ADOQ_SaveAnswers.FieldName('RowCount').AsInteger = 0 then
    begin //Добавление новых данных по количеству человек за год
      ADOQ_SaveAnswers.Active := False;
      ADOQ_SaveAnswers.SQL.Text := 'INSERT INTO YearResp
(AnkYear, AllResp, AllWorkers, DateBegin, DateEnd) '
+ 'VALUES ('
+ FormatDateTime('YYYY', DTP_DateBegin.DateTime) + ', '
+ IntToStr(SE_CountResp.Value) + ', '
+ IntToStr(SE_CountWorkers.Value) + ', '
+ '''' + DateToStr(DTP_DateBegin.DateTime) + ''', '
+ '''' + DateToStr(DTP_DateEnd.DateTime) + ''')';
      ADOQ_SaveAnswers.ExecSQL;
      for i := 1 to CountAnswer do
        begin // Подсчет и добавление новых данных в базу результатов
          ADOQ_SaveAnswers.SQL.Text := 'INSERT INTO results
(AnkYear, IDA, CountResp, PercResp) '
+ 'VALUES ('
+ FormatDateTime('YYYY', DTP_DateBegin.DateTime) + ', '
+ IntToStr(Answers[i].IDA) + ', '
+ IntToStr(Answers[i].SpinEdit.Value) + ', '
+ FloatToStr(Answers[i].SpinEdit.Value /
SE_CountResp.Value * 100) + ')';
          ADOQ_SaveAnswers.ExecSQL;
        end;
    end;
```

```
ShowMessage('Данные успешно сохранены в БД!');
```

end В случае если сохраняем данные за год, который уже есть в базе данных, выходит предупреждение (Рисунок 2.11).

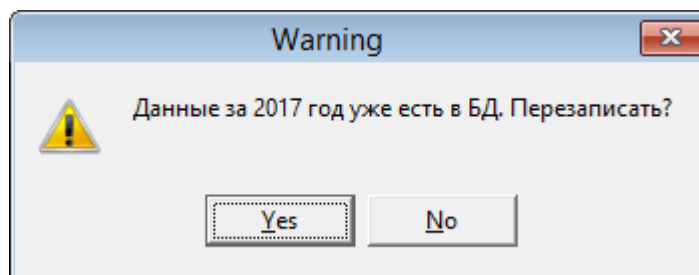


Рисунок 2.11 – Предупреждение

В данном случае выполняется следующий код:

```
else
begin //Вывод предупреждения и вопроса пользователю
    if MessageDlg('Данные за ' + FormatDateTime('YYYY',
DTP_DateBegin.DateTime) + ' год уже есть в БД. Перезаписать?',
mtWarning, [mbYes, mbNo], 0) = 6
    then
        begin //Удаление сохраненных данных в базе данных
            ADOQ_SaveAnswers.Active := False;
            ADOQ_SaveAnswers.SQL.Text := 'DELETE FROM results WHERE AnkYear = '
            + FormatDateTime('YYYY', DTP_DateBegin.DateTime);
            ADOQ_SaveAnswers.ExecSQL;
            ADOQ_SaveAnswers.SQL.Text := 'DELETE FROM YearResp WHERE AnkYear = '
            + FormatDateTime('YYYY', DTP_DateBegin.DateTime);
            ADOQ_SaveAnswers.ExecSQL; //Запись новых данных в базу
            ADOQ_SaveAnswers.SQL.Text := 'INSERT INTO YearResp (AnkYear,
AllResp, AllWorkers) '
            + 'VALUES ('
            + FormatDateTime('YYYY', DTP_DateBegin.DateTime) + ', '
            + IntToStr(SE_CountResp.Value) + ', '
            + IntToStr(SE_CountWorkers.Value) + ')';
            ADOQ_SaveAnswers.ExecSQL;
```

```

for i := 1 to CountAnswer do
begin //Запись новых данных в базу
ADOQ_SaveAnswers.SQL.Text := 'INSERT INTO results (AnkYear, IDA,
CountResp, PercResp) '
+ 'VALUES ('
+ FormatDateTime('YYYY', DTP_DateBegin.DateTime) + ', '
+ IntToStr(Answers[i].IDA) + ', '
+ IntToStr(Answers[i].SpinEdit.Value) + ', '
+ FormatFloat('0.0', Answers[i].SpinEdit.Value /
SE_CountResp.Value * 100) + ')';
ADOQ_SaveAnswers.ExecSQL;
end;
ShowMessage('Данные успешно сохранены в БД!');
end;
end;
end;
end;

```

Вывод отчета по результатам обработки анкетирования происходит в шаблон файла Microsoft Word (Рисунок 2.12).



Рисунок 2.12 – Пример вывода отчета

Выводы по разделу два:

В процессе работы над данной выпускной квалификационной работой было проведено более глубокое изучение предметной области, освоены язык программирования Delphi и SQL-запросы к базе данных Microsoft Access 2010. Получены навыки работы со средой разработки Delphi и базой данных Microsoft Access 2010.

Было реализована ИС для обработки данных анкетирования. Программное приложение представляет собой листинг формы ввода данных (Приложение Б). Все разрешенные для публикации исходные модули представлены на компакт диске (Приложение В). Реализован интерфейс ИС, а также модуль обработки данных. Таким образом, все поставленные цели по разработке выпускной квалификационной работы, успешно решены.

3 ОРГАНИЗАЦИОННО-ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

3.1 Составление сметы затрат

Сметная стоимость работ складывается из следующих статей затрат:

- 1) Затраты на материалы.
- 2) Транспортные расходы.
- 3) Контрагентские расходы.
- 4) Покупные изделия и полуфабрикаты.
- 5) Специальное оборудование.
- 6) Заработная плата.
- 7) Накладные расходы.

Сметная стоимость работ по разработке приложения включает следующие статьи затрат:

- 1) Затраты на материалы и электроэнергию.
- 2) Контрагентские расходы.
- 3) Заработная плата.
- 4) Накладные расходы.

Расчет затрат на материалы и электроэнергию, необходимые для разработки программного продукта представлен в таблице 3.1.

Произведем расчет потраченной электроэнергии.

Считаем, что за один час работы за компьютером расходуется 0,4 кВт/ч электроэнергии. За время разработки ИС время работы за компьютером составляло 4ч. Отсюда следует, что за 4 часов будет потрачено: $0.4 \cdot 4 = 1.6$ кВт/ч.

Следовательно, за 22 рабочих дня, по 4 часов работы за компьютером, получается: $22 \cdot 4 = 88$ часа.

За 3 месяца работы получаем расход электроэнергии в размере: $88 \cdot 3 \cdot 1,6 = 422,4$ кВт/ч.

Отсюда следует, что при стоимости электроэнергии (на момент разработки ИС) 4,68 рублей за 1 кВт/ч, затраты на электроэнергию составят:
 $\Sigma = 422,4 \cdot 4,68 = 1976,83$ рублей

Таблица 3.1 – Затраты на материалы и электроэнергию

Наименование	Единица измерения	Цена за единицу, руб.	Количество	Стоимость, руб.
Бумага для принтера, А4	пачка	200	1	200,00
Картридж для принтера	шт	2099,00	1	2099,00
Электричество	кВт×ч	4,68	422,4	1976,83
Папка для бумаг	шт	10	8	80,00
Компакт-диск CD-R	шт	50	1	50,00
Итого:				4405,83

В статью «Контрагентские расходы» включается стоимость работ, выполненных сторонними организациями. В нашем случае такими расходами являются использование интернета. В таблице 3.2 представлен расчет затрат по статье «Контрагентские расходы».

Таблица 3.2 – Контрагентские расходы

Наименование работы	Кол-во	Цена за единицу, руб.	Стоимость, руб.
Предоставление доступа в интернет	3 месяца (безлимит)	1000	3000,00
Итого:			3000,00

При разработке программного продукта был использован стационарный компьютер стоимостью 33000 рублей со сроком службы 5 лет.

$$\frac{33000 \text{ руб.}}{5 \text{ лет}} \approx 18,08 \text{ руб. в день.}$$

Амортизация компьютера за период разработки составит:

$$18,08 \text{ руб.} \cdot 66 \text{ день} = 1193,28 \text{ руб.}$$

Расчет основной заработной платы выполняется на основе трудоемкости выполнения каждого этапа в человеко-часах и величины месячного должностного оклада исполнителя. Среднее количество рабочих дней в месяце равно 22. Средняя продолжительность рабочего дня 8 часов. Таким образом, дневная заработная плата определяется делением размера ежемесячной заработной платы на количество рабочих дней в месяце, а почасовая оплата труда определяется делением дневной заработной платы на продолжительность рабочего дня. Произведение трудоемкости на сумму почасовой оплаты труда определяет затраты по зарплате для каждого работника на все время разработки. Расчет заработной платы представлен в таблице 3.3.

Таблица 3.3 – Расчет заработной платы

Исполнители:	Консультант по экономике	Консультант по БЖД	Руководитель работы	Практикант - программист
НАЧИСЛЕНИЯ:				
Оклад, руб./мес	20500,00	23000,00	21500,00	20863,28
Северная надбавка (50%)	10250,00	11500,00	10750,00	–
Районный коэффициент (1,7)	14350,00	16100,00	15050,00	–
Итого заработная плата:	45100,00	50600,00	47300,00	20863,28
Заработная плата, руб. день	2050,00	2300,00	2150,00	948,3
Заработная плата, руб. час	256,25	287,5	268,75	118,54
Количество часов	1	1	1	264
Сумма, руб.	256,25	287,5	268,75	31294,56
Итого:	32107,06			

При расчете себестоимости также учитываются плановые отчисления на социальные нужды и в бюджет, которые представлены в таблице 3.4.

Таблица 3.4 – Отчисления от заработной платы

Отчисления	% от заработной платы	Сумма, руб.
Отчисления в ПФР	22	7063,55
Отчисления в ФСС	2,9	931,10
Отчисления в ФФОМС	5,1	1637,5
Итого:	30	9632,15

Полная смета затрат представлена в таблице 3.5

Таблица 3.5 – Смета затрат, руб.

Статья затрат	Сумма, руб.
Затраты на материалы и электроэнергию	4405,83
Контрагентские расходы	3000,00
Амортизация оборудования	1193,28
Заработная плата	32107,06
Отчисления от заработной платы	9632,15
Итого:	50338,32

3.2 Оценка технико-экономической эффективности

Без использования приложения разработанного в рамках данной выпускной квалификационной работы, происходит нерациональная трата рабочего времени сотрудника, который обрабатывает результаты анкетирования вручную.

На обработку одной анкеты тратится 15 минут. В среднем в анкетировании принимают участие 55 человек, умножаем минуты на количество человек и получаем: $15 \cdot 55 = 825$ минут или 13 часов 45 минут только на обработку данных анкетирования. К этому времени следует добавить время, которое тратится на заполнение сводного отчета по филиалу. Отсюда следует, что среднее время обработки данных анкетирования достигает 16 часов, при 8 часовом рабочем дне, обработка занимает 2 дня.

Разработанная ИС позволит сократить время на обработку данных анкетирования в 2 раза.

Теперь сотруднику потребуется в среднем 8 часов рабочего времени (1 рабочий день) на полную обработку данных анкетирования. Время, которое было сэкономлено с помощью данной ИС, сотрудник филиала может потратить на выполнение других должностных обязанностей.

Выводы по разделу три:

Без использования приложения разработанного в рамках выпускной квалификационной работы происходит нерациональная трата рабочего времени сотрудников.

Основной экономической эффективностью разработанной ИС являются следующие факторы:

- автоматизация обработки данных анкетирования;
- скорость обработки.

Необходимо подчеркнуть, что у программных продуктов практически отсутствует процесс физического старения и износа. Основные затраты приходятся на разработку и внедрение. Так как затраты на разработку данного продукта минимальны, а его экономическая выгодность доказана, программа может быть благополучно внедрена в филиале.

4 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

С развитием прогресса существенную роль играет возможность безопасного выполнения людьми своих трудовых обязанностей. В связи с этим была создана и развивается наука о безопасности и жизнедеятельности человека (далее – БЖД).

Цель и содержание БЖД:

- создание нормального, то есть комфортного состояния среды обитания человека;
- предупреждение воздействия тех или иных негативных факторов на человека;
- ликвидация отрицательных последствий воздействия опасных и вредных факторов;
- идентификация опасности, распознавание и количественная оценка негативных воздействий среды обитания.

На рабочем месте должны быть предусмотрены меры защиты от возможного воздействия небезопасных и вредоносных факторов производства. Уровни этих факторов не должны быть выше предельных значений, оговоренных правовыми, техническими и санитарно-техническими нормами. Данные нормативные документы обязывают к созданию на рабочем месте условий труда, при которых влияние небезопасных и вредоносных факторов на работников либо устранено совсем, либо находится в допустимых пределах.

Данный раздел выпускной квалификационной работы посвящен рассмотрению следующих вопросов:

- определение оптимальных условий труда программиста;
- требования к производственным помещениям (освещение, микроклимат, шум, магнитные, электромагнитные поля);
- эргономические требования к рабочему месту;
- режим труда;
- организация пожарной профилактики.

4.1 Характеристика условий труда программиста

Технический прогресс принес значительные перемены в требования производственной деятельности сотрудников интеллектуальной работы. Их деятельность стала наиболее активна, интенсивна, вызывающая существенные затраты интеллектуальной, психологической и физиологической энергии. Это потребовало единого решения трудностей эргономики, гигиены и организации труда, регламентации режимов труда и отдыха.

На сегодняшний день компьютерная техника обширно используется в абсолютно всех сферах работы человека. При работе с компьютером человек подвергается влиянию ряда небезопасных и вредоносных производственных условий: электромагнитных полей, инфракрасного, ионизирующего излучений, шума и др.

Деятельность работы за компьютером характеризуется существенным интеллектуальным усилием и нервно-эмоциональной нагрузкой работников, большой напряженностью визуальной деятельности и довольно высокой нагрузкой на мышцы рук при работе с клавиатурой электронно-вычислительной машины (далее – ЭВМ). Огромной ролью обладает рациональная конструкция и расположение элементов рабочего места, что немаловажно для поддержания оптимальной рабочей позы работника.

В ходе работы с компьютером следует придерживаться правильным режимом труда и отдыха. В противном случае у работника отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

4.2 Требования к производственным помещениям

4.2.1 Шум на рабочих местах

По характеру спектра шума выделяют [8]:

– тональный шум, в спектре которого имеются выраженные тоны. Тональный характер шума для практических целей устанавливается измерением уровней звукового давления в 1/3–октавных полосах частот в диапазоне частот 25–10000 Гц по превышению уровня в одной из 1/3–октавных полос над соседними не менее чем на 10 дБ или по превышению суммарного уровня двух соседних 1/3–октавных полос, уровни которых отличаются менее чем на 3 дБ, над соседними не менее чем на 12 дБ;

– широкополосный шум, не содержащий выраженных тонов.

По временным характеристикам шума выделяют [8]:

– постоянный шум, уровень звука которого за 8 часовой рабочий день или за время измерения изменяется не более чем на 5 дБА при режиме усреднения шумомера S (медленно);

– непостоянный шум, уровень звука которого за 8 часовой рабочий день, рабочую смену или за время измерения изменяется более чем на 5 дБА при измерениях с постоянной времени усреднения шумомера S (медленно);

– импульсный шум, состоящий из одного или нескольких звуковых событий, каждый длительностью менее 1с, при этом уровни звука $L_{p,AI\max}$ и $L_{p,AS\max}$, измеренные соответственно с временными коррекциями I (импульс) и S (медленно), отличаются не менее чем на 7 дБ.

Нормативным уровнем звука на рабочих местах, является 80 дБА.

Для отдельных отраслей допускается эквивалентный уровень шума на рабочих местах от 80 до 85 дБА при условии подтверждения приемлемого риска здоровью работающих по результатам проведения оценки профессионального риска здоровью трудящихся, а также выполнения комплекса мероприятий, направленных на минимизацию рисков здоровью трудящихся.

В случае превышения уровня шума на рабочем месте выше 80 дБА, работодатель обязан провести оценку риска здоровью трудящихся и подтвердить приемлемый риск здоровью.

Работы в условиях воздействия эквивалентного уровня шума выше 85 дБА не допускаются.

4.2.2 Освещение

Грамотно спроектированное и сделанное производственное освещение улучшает условия зрительной работы, уменьшает утомляемость, способствует увеличению производительности труда, хорошо оказывает большое влияние в производственную среду.

Существует три вида освещения – естественное, искусственное и совмещенное (естественное и искусственное вместе). В помещениях с ЭВМ необходимо применять системы комбинированного освещения [10].

При выполнении работ категории высокой зрительной точности (наименьший размер объекта 0,3–0,5мм) величина коэффициента естественного освещения (далее – КЕО) должна быть не ниже 1,5%, а при зрительной работе средней точности (наименьший размер объекта 0,5–1,0 мм) КЕО должен быть не ниже 1,0% [9].

В качестве источников искусственного освещения обычно применяются люминесцентные лампы типа ЛБ или ДРЛ, которые попарно объединяются в светильники, которые должны располагаться над рабочими поверхностями равномерно.

Требования к освещенности в помещениях, где установлены компьютеры, согласно СанПиН 2.2.2/2.4.1340-03 следующие: при выполнении зрительных работ высокой точности общая освещенность должна составлять 300 лк, а комбинированная – 750 лк; аналогичные требования при выполнении работ средней точности – 200 и 300 лк соответственно [10].

Кроме этого, все без исключения поле зрения должно быть освещено достаточно равномерно – это основное гигиеническое требование. Уровень освещения помещения и яркость экрана компьютера должны быть примерно одинаковыми, т.к. яркий свет в области периферийного зрения повышает напряженность глаз, что приводит к их быстрой утомляемости.

С целью обеспечения нормируемых значений освещенности в помещениях необходимо не реже двух раз в год мыть окна, оконные рамы и светильники и вовремя заменять перегоревшие лампы.

Рабочие места должны располагаться таким образом, чтобы естественный свет падал сбоку, преимущественно слева.

4.2.3 Параметры микроклимата

На рабочем месте, оборудованном монитором либо компьютер, обязаны осуществляться требования к микроклимату.

Должна поддерживаться оптимальная влажность воздуха, с целью чего могут быть использованы специальные увлажнители воздуха или кондиционеры.

Допустимые и оптимальные параметры относительной влажности воздуха и температуры приведены в таблице 4.2.

Таблица 4.2 – Допустимые и оптимальные параметры влажности и температуры воздуха

Допустимые параметры		Оптимальные параметры	
Температура, С°	Относительная влажность, %	Температура, С°	Относительная влажность, %
18	39	19	62
22	31	20	58
-	-	21	55

Воздух в помещении должен соответствовать нормам по содержанию вредных химических веществ, а также аэронов. Уровень ионизации воздуха согласно СанПиН 2.2.2/2.4.1340-03 представлен в таблице 4.3 [10].

Таблица 4.3 – Уровни ионизации воздуха в помещениях

	Число ионов в 1см ³ воздуха	
	n-	N+
Необходимые	600	400
Оптимальные	300 – 5000	1500 – 3000
Максимально допустимые	50000	50000

Объем помещений, в которых размещены сотрудники вычислительных центров, не должен быть меньше 19,5 м³/человека с учетом максимального числа одновременно работающих в смену.

С целью предоставления комфортных условий используются как рациональная организация проведения работ в зависимости от времени года и суток, чередование труда и отдыха, так и вентиляция, кондиционирование воздуха, отопительная система.

4.2.4 Магнитные, электромагнитные поля на рабочих местах

Постоянное магнитное поле:

– оценка и нормирование постоянного магнитного поля (далее – ПМП) осуществляется согласно уровню магнитного поля для условий общего (на все тело) и локального (кисти рук, предплечье) влияния в зависимости от времени пребывания трудящегося в постоянном магнитном поле за смену;

– уровень ПМП оценивают в единицах напряженности магнитного поля (Н) в кА/м или в единицах магнитной индукции (В) в мТл. Предельно допустимые уровни (далее – ПДУ) напряженности (индукции) ПМП на рабочих местах представлены в таблице 4.4.

Таблица 4.4 – ПДУ постоянного магнитного поля на рабочих местах [8]

Время Воздействия за рабочий день, мин	Условия воздействия			
	общее		локальное	
	ПДУ напряженности, кА/м	ПДУ магнитной индукции, мТл	ПДУ напряженности, кА/м	ПДУ магнитной индукции, мТл
≤10	24	30	40	50
11-60	16	20	24	30
61-480	8	10	12	15

ПДУ электромагнитных полей на рабочих местах пользователей компьютеров и другими средствами информационно-коммуникационных технологий представлены в таблице 4.5.

Таблица 4.5 – ПДУ электромагнитных полей на рабочих местах пользователей ПК и другими средствами ИКТ [8]

Нормируемые параметры	ПДУ	
Напряженность электрического поля	5 Гц - < 2 кГц	25 В/м
	2 кГц - < 400 кГц	2,5 В/м
Напряженность магнитного поля	5 Гц - < 2 кГц	250 нТл
	2 кГц - < 400 кГц	25 нТл
Плотность потока энергии	300 МГц - 300 ГГц	10 мкВт/см ²
Напряженность электростатического поля		15 кВ/м

4.3 Эргономические требования к рабочему месту

Проектирование рабочих мест относится к числу значимых проблем эргономического проектирования в сфере вычислительной техники.

Рабочее место и взаимное размещение всех абсолютно всех его компонентов должно соответствовать антропометрическим, физическим и психологическим требованиям. Огромную роль имеет также вид деятельность.

Эргономическими нюансами проектирования рабочих мест являются:

– высота рабочей поверхности;

- размеры пространства для ног;
- требования к расположению документов на рабочем месте;
- свойства рабочего кресла;
- требования к поверхности рабочего стола;
- регулируемость элементов рабочего места.

Основные элементы рабочего места программиста – стол и кресло. Главное рабочее положение – положение сидя.

Положение сидя вызывает наименьшее утомление программиста. Рациональная планировка рабочего места предусматривает порядок и постоянство размещения средств труда и документации. В таком случае, то, что необходимо для выполнения работ чаще, находится в зоне легкой досягаемости рабочего пространства.

Моторное поле – пространство рабочего места, в котором могут реализовываться двигательные действия человека.

Максимальная зона досягаемости рук – данная часть моторного поля рабочего места, ограниченного дугами, описываемыми предельно вытянутыми руками при движении их в плечевом суставе.

Оптимальная зона – часть моторного поля рабочего места, ограниченного дугами, описываемыми предплечьями при движении в локтевых суставах с опорой в точке локтя и с сравнительно неподвижным плечом.

Для комфортной работы стол должен удовлетворять следующим условиям:

- высота стола должна быть выбрана с учетом способности сидеть свободно, при необходимости опираясь на подлокотники;
- нижняя часть стола должна быть сконструирована таким образом, чтобы программист мог сидеть не подпирая ноги;
- на плоскости стола не должно быть бликов в поле зрения программиста;
- конструкция стола должна предусматривать наличие выдвигаемых ящиков;

– высота рабочей поверхности рекомендовано в границах 680–760 мм. Высота поверхности, на которую устанавливается клавиатура, должна быть приблизительно 650 мм.

Огромная роль придается характеристикам рабочего кресла. Таким образом, рекомендуемая высота сиденья над уровнем пола находится в пределах 420–550 мм. Поверхность сиденья мягкая, передний край закругленный, а угол наклона спинки – регулируемый [10].

Следует учитывать при проектировании возможность различного размещения документов: слева от монитора, между монитором и клавиатурой и т.д. Наиболее этого, если монитор имеет низкое качество изображения, например видны мелькания, расстояние от глаз до экрана делают больше (приблизительно 700 мм), нежели расстояние от глаза до документа (300–450 мм). В целом при высоком качестве изображения на мониторе, расстояние от глаз работника до экрана, документа и клавиатуры может быть равным [10].

Положение экрана определяется:

- расстоянием считывания (0,6–0,7м);
- направлением взгляда на 20° ниже горизонтали к середине экрана, причем экран перпендикулярен данному направлению.

Так же должна предусматриваться возможность регулирования экрана:

- по высоте +3 см;
- в левом и правом направлениях;
- по наклону от -10° до $+20^\circ$ относительно вертикали.

Кроме того огромная роль придается правильной рабочей позе сотрудника. При неудобной позе могут появиться боли в мышцах, суставах и сухожилиях. Условия к рабочей позе пользователя компьютера следующие:

- голова не должна быть наклонена более чем на 20° ;
- локти – под углом 80° – 100° ;
- предплечья и кисти рук – в горизонтальном положении;
- плечи должны быть расслаблены.

Причина неправильной позы пользователей определена соответствующими условиями: клавиатура находится слишком высоко, некуда положить руки, мало места для ног.

В целях преодоления отмеченных недочетов даются общие рекомендации: должны быть предусмотрены специальные приспособления для регулирования высоты стола, клавиатуры и экрана, а также подставка для рук.

Во время пользования компьютером врачи рекомендуют устанавливать монитор на расстоянии 50–60 см от глаз. Кроме того эксперты считают, что верхняя часть монитора должна быть на уровне глаз или чуть ниже. Когда человек смотрит прямо перед собой, его глаза открываются шире, чем, если он смотрит вниз. В случае если экран установлен высоко, а глаза широко открыты, нарушается функция моргания. Это значит, то что глаза не закрываются полностью, не омываются слезной жидкостью, что приводит к их стремительной утомляемости.

Формирование подходящих условий труда и правильное эстетическое оформление рабочих мест на производстве имеет большое значение, для облегчения труда благоприятно влияющее на производительность.

4.4 Режим труда

Как уже было ранее отмечено, при работе с персональным компьютером весьма значительную роль играет соблюдение правильного режима труда и отдыха. В ином случае у работника отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, нарушение сна и т.д.

В таблице 4.6 представлены сведения о регламентированных перерывах, которые следует делать при работе на компьютере, в зависимости от продолжительности рабочей смены, типов и категорий трудовой деятельности с ЭВМ (в соответствии с СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к

видео дисплейным терминалам, персональным электронно-вычислительным машинам и организации работ») [10].

Таблица 4.6 – Время регламентированных перерывов

Категория работы ЭВМ	Суммарное время регламентированных перерывов, мин		Уровень нагрузки за рабочую смену при видах работы с ЭВМ		
	При 8-часовой смене	При 12-часовой смене	Группа А, количество знаков	Группа Б, количество знаков	Группа В, часов
I	30	70	до 20000	до 15000	до 2,0
II	50	90	до 40000	до 30000	до 4,0
III	70	120	до 60000	до 40000	до 6,0

Примечание. Время перерывов предоставлено при соблюдении отмеченных Санитарных правил и норм. При несоответствии фактических условий труда требованиям Санитарных правил и общепризнанных норм время регламентированных перерывов следует увеличить на 30%.

В соответствии с СанПиН 2.2.2/2.4.1340-03 все без исключения виды трудовой деятельности, связанные с применением компьютера, делятся на три группы [9]:

- группа А: работа по считыванию информации с экрана ЭВМ с предварительным запросом;
- группа Б: работа по вводу информации;
- группа В: творческая работа в режиме диалога с ЭВМ.

Результативность перерывов увеличивается при сочетании с производственной гимнастикой или организации особого помещения для отдыха персонала с удобной мягкой мебелью, аквариумом, зеленой зоной и т.п.

4.5 Организация пожарной профилактики

Противопожарные мероприятия ведутся на основании единых государственных распоряжений, правил и норм, в частности ППБ 01-03 «Правила пожарной безопасности Российской Федерации». В соответствии с ними помещения, в которых ведутся работы с применением персональных компьютеров, а также сами компьютеры содержат огромное количество горючих и легковоспламеняющихся материалов, и для того, чтобы не позволить непреднамеренного возгорания, необходимо строго их соблюдать.

Источником возгорания на рабочем месте могут быть провода, электронные схемы ПК, устройства электропитания, сильно нагревающиеся узлы устройств. По этой причине на 100 м² площади таких помещений, оснащенных компьютерной техникой, обязан располагаться минимум 1 огнетушитель углекислого типа.

Все без исключения сотрудники организаций обязаны допускаться к работе только после прохождения противопожарного инструктажа, а при изменении специфики деятельности проходить дополнительное обучение по предупреждению и тушению возможных пожаров. Главы организаций или индивидуальные предприниматели обладают возможностью назначать лиц, которые по занимаемой должности или по характеру выполняемых работ должны исполняемых надлежащие правила пожарной безопасности.

Государственные органы в пределах своей ответственности реализуют меры пожарной безопасности в подведомственных организациях и на соответствующих территориях, оказывают требуемую помощь пожарной охране.

В абсолютно всех производственных, административных, складских и вспомогательных помещениях на видных местах должны быть вывешены таблички с указанием номера телефона вызова пожарной охраны.

В каждой организации распорядительным документом обязаны быть определены и оборудованы места для курения, установлен порядок обесточивания электрооборудования в случае пожара и по окончании рабочего дня, действия работников при обнаружении пожара.

В случае пожара для тушения компьютерной техники следует использовать газовые и углекислотные огнетушители. Их плюсом является значительная эффективность тушения пожара и сохранность электронного оборудования.

Применение ЭВМ в помещениях приводит к принятию серьезных мероприятий защиты от пожаров, определяемых СП 512-78 «Инструкции по проектированию зданий и помещений для ЭВМ» и СНиП 11-2-80 «Противопожарные нормы проектирования зданий и сооружений». В данных документах изложены главные требования к огнестойкости зданий и сооружений, противопожарным преградам, эвакуации людей из зданий и помещений.

Выводы по разделу четыре:

Обеспечение безопасности условий труда, устранение профессиональных заболеваний и производственного травматизма является одной из главных забот человеческого общества. Выполнение условий, определяющих наилучшую организацию рабочего места, даст возможность сохранить хорошую работоспособность на протяжении всего рабочего дня, повысит эффективность труда программиста.

В данном разделе была рассмотрена проблема безопасности жизнедеятельности, связанных с работой за компьютером. Проведено исследование условий, оказывающих вредное воздействие на органы зрения пользователя ЭВМ. Сформированы общие требования к помещению и освещению. Проведены анализы шума, магнитного и электромагнитного поля, освещения и пожаробезопасности на рабочем месте пользователя ЭВМ. В связи с этим, были изучены различные нормы, сертификаты и правила, определялись пути решения

проблем, для сокращения пагубного влияния на человека и обеспечения безопасных условий труда.

ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе исследовались различные аспекты разработки ИС по обработке данных анкетирования.

Был проведен анализ существующих подходов к проектированию системы, выявлены проблемы анкетирования в филиале ЮУрГУ.

В процессе разработки выпускной квалификационной работы было проведено более глубокое изучение предметной области, освоен язык программирования Delphi и СУБД Microsoft Access.

Разработана ИС обработки данных анкетирования. Реализован интерфейс приложения, а также реализован модуль обработки данных анкетирования.

В экономической части определены затраты при производстве системы, определены показатели экономического эффекта.

В разделе безопасность жизнедеятельности приведены правила работы за компьютером, сформирована характеристика условий труда программиста и приведены основные мероприятия по охране труда.

Таким образом, все поставленные задачи по реализации работы успешно решены, а разработанная ИС соответствует требованиям, изложенным в исходных данных к работе.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Самоукина, Н. В. Эффективная мотивация персонала при минимальных финансовых затратах/ Н. В Самоукина. – М. : Вершина, 2006. -224 с.: ил.
- 2 ГОСТ Р ИСО 9241-11-2010. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 11. Руководство по обеспечению пригодности использования. – М.: ФГУП «СТАНДАРТИНФОРМ, 2011. – 28 с.
- 3 Купер, А. Алан Купер об интерфейсе. Основы проектирования взаимодействия. / А. Купер, Р. Рейман, Д. Кронин – Пер. с англ. – СПб.: Символ-Плюс, 2009. – 688 с., ил.
- 4 Локвуд, Л. Разработка программного обеспечения. / Л. Константайн, Л. Локвуд – СПб.: Питер, 2004. – 592с.: ил.
- 5 Скляр, А.Я. Введение в InterBase / А.Я. Скляр – М.: Горячая линия-Телеком, 2002. – 517 с: ил.
- 6 ГОСТ Р ИСО 14915-1-2016. Эргономика мультимедийных пользовательских интерфейсов. – М.: ФГУП «СТАНДАРТИНФОРМ, 2016. – 19 с.
- 7 ГОСТ 28195-89. Оценка качества программных средств. Общие положения. – М.: Изд-во стандартов, 2001. – 31 с.
- 8 СанПиН 2.2.4.3359-16. Санитарно-эпидемиологические требования к физическим факторам на рабочих местах (утв. постановлением Главного государственного санитарного врача РФ от 21 июня 2016 г. № 81). – М.: Изд-во стандартов, 2016. – 69 с.
- 9 СанПиН 2.2.1/2.1.1.1278-03. Гигиенические требования к естественному, искусственному и совмещенному освещению жилых и общественных зданий. – М.: Информационно-издательский центр Минздрава России, 2003. – 28 с.
- 10 СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы. – М.: Информационно-издательский центр Минздрава России, 2003. – 56 с.

11 Белов, В.В. Программирование в DELPHI: процедурное, объектно-ориентированное, визуальное [Электронный ресурс] : учеб. пособие / В.В. Белов, В.И. Чистякова. – Электрон. дан. – Москва : Горячая линия-Телеком, 2014. – 240 с. – Режим доступа: <https://e.lanbook.com/book/64091> - Загл. с экрана. [дата обращения – 16.04.2018]

12 Дейт, К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ./ К. Дж. Дейт – М.: Издательский дом "Вильямс", 2005. — 1328 с.: ил. – Парал. тит. англ.

13 Шамис, В.А. Borland C++ Builder 6. Для профессионалов / В.А. Шамис. – СПб.: Питер, 2003. – 798 с.: ил.

14 Аблязов, В.И. Проектирование баз данных в среде Microsoft Office Access 2003, 2007 и 2010: учебное пособие / В. И. Аблязов – СПб.: Изд-во Политехн. ун-та, 2014. – 107 с.

15 Фаронов, В.В. Delphi, язык и среда программирования: учебные курсы. / В.В. Фаронов – Питер, 2003.

16 Брауде, Э., Технологии разработки программного обеспечения. / Э. Брауде – СПб.: Питер, 2004. – 655 с.: ил.

17 Осипов, Д.Л. Базы данных и Delphi. Теория и практика. / Д.Л. Осипов – СПб.: БВХ-Петербург, 2011. – 752с.: ил.

18 Фиайли, К. SQL: Пер. с англ./ К. Фиайли – М.: ДМК Пресс. – 456 с.: ил.

19 Бекаревич, Ю. Б. Самоучитель Access 2010 / Ю. Б. Бекаревич, Н. В. Пушкина. – СПб.: БХВ-Петербург, 2011. – 432 с.: ил.

20 Методические рекомендации по подготовке и оформлению выпускной квалификационной работы (проекта) для технических направлений подготовки 09.03.01 Информатика и вычислительная техника, 09.03.04 Программная инженерия, 12.03.01 Приборостроение, 23.03.01 Технология транспортных процессов / сост. Л.Н.Буйлушкина. - Нижневартовск, 2017. - 35с.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А. СВИДЕТЕЛЬСТВО О ПУБЛИКАЦИИ



Рисунок А.1 – Свидетельство о публикации в журнале

ПРИЛОЖЕНИЕ Б. ЛИСТИНГ ФОРМЫ ВВОДА ДАННЫХ

```
unit UInputInfo;
interface
uses Windows, Classes, Graphics, Forms, Controls, StdCtrls,
ExtCtrls, Spin,
    ToolWin, ComCtrls, Dialogs, SysUtils, DB, ADODB;
type
// Панель с утверждением и полем для оценки
TAnswer = class
    MainPanel : TPanel;
    EditPanel : TPanel;
    SpinEdit : TSpinEdit;
    LAnswer : TLabel;
    IDA : Integer;
    constructor Create(AOwner: TComponent);
end;
type
// Панель с вопросом
TQuestion = class
    PHeader : TPanel;
    LQuestion : TLabel;
    constructor Create(AOwner: TComponent);
end;
type
TInputInfo = class(TForm)
    PHeader: TPanel;
    LHeaderName: TLabel;
    ToolBar1: TToolBar;
    tbClear: TToolButton;
    tbSave: TToolButton;
    tbFinishTest: TToolButton;
    tbSep1: TToolButton;
    tbSep2: TToolButton;
    tbClose: TToolButton;
```



```

tbSep3: TToolButton;
    tbSep4: TToolButton;
    SB_Anketa: TScrollBar;
    ADOC_InputInfo: TADOConnection;
    ADOQ_InputInfo: TADOQuery;
    DS_InputInfo: TDataSource;
    Label1: TLabel;
    SE_CountResp: TSpinEdit;
    ADOQ_SaveAnswers: TADOQuery;
    Label2: TLabel;
    SE_CountWorkers: TSpinEdit;
    DTP_DateBegin: TDateTimePicker;
    DTP_DateEnd: TDateTimePicker;
    procedure FormClose(Sender: TObject; var Action:
TCloseAction);
        procedure FormCreate(Sender: TObject);
        procedure tbSaveClick(Sender: TObject);
        procedure tbClearClick(Sender: TObject);
        procedure tbFinishTestClick(Sender: TObject);
        procedure FormDestroy(Sender: TObject);
        procedure tbCloseClick(Sender: TObject);
        procedure FormActivate(Sender: TObject);
        procedure SB_AnketaMouseWheelUp(Sender: TObject; Shift:
TShiftState;
            MousePos: TPoint; var Handled: Boolean);
        procedure SB_AnketaMouseWheelDown(Sender: TObject; Shift:
TShiftState;
            MousePos: TPoint; var Handled: Boolean);
        procedure FormKeyUp(Sender: TObject; var Key: Word; Shift:
TShiftState);
        procedure FormKeyDown(Sender: TObject; var Key: Word; Shift:
TShiftState);
    private
        Answers : array[1..67] of TAnswer;

```

```

Questions : array[1..6] of TQuestion;
  end;
const
  CountAnswer : Integer = 67;
implementation
uses UMain, Math;
{$R *.dfm}
procedure TInputInfo.FormActivate(Sender: TObject);
begin
  DTP_DateBegin.SetFocus;
end;
procedure TInputInfo.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  Action := caFree;
end;
procedure TInputInfo.FormCreate(Sender: TObject);
var
  i, j : Integer;
begin
  DecimalSeparator := '.';
  ADOC_InputInfo.Connected := False;
  ADOC_InputInfo.ConnectionString :=
'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
ExtractFilePath(paramstr(0)) + 'Results.mdb ;Persist Security
Info=False';
  ADOC_InputInfo.Connected := True;
  ADOQ_InputInfo.Active := True;
  ADOQ_InputInfo.First;
  i := 1;
  j := 1;
  While not ADOQ_InputInfo.Eof do
    begin
      if (ADOQ_InputInfo.FieldName('NumA').AsString = '1')

```

```

or (ADOQ_InputInfo.FieldName('NumA').AsString = '')
  then
    begin
      Questions[i] := TQuestion.Create(Self.SB_Anketa);
      Questions[i].LQuestion.Caption := ' ' +
ADOQ_InputInfo.FieldName('NumQ').AsString +
ADOQ_InputInfo.FieldName('Question').AsString;
      Questions[i].PHeader.Visible := True;
      i := i + 1;
    end;
  if ADOQ_InputInfo.FieldName('NumA').AsString <> '' then
    begin
      Answers[j] := TAnswer.Create(Self.SB_Anketa);
      Answers[j].LAnswer.Caption := ' ' + ' +
ADOQ_InputInfo.FieldName('Answer').AsString;
      Answers[j].MainPanel.Visible := True;
      Answers[j].IDA :=
ADOQ_InputInfo.FieldName('IDA').AsInteger;
      j := j + 1;
    end;
  ADOQ_InputInfo.Next;
end;
ADOQ_InputInfo.Active := False;
PHeader.Visible := True;
end;
procedure TInputInfo.FormDestroy(Sender: TObject);
var
  i : integer;
begin
  for i := 1 to CountAnswer do
    FreeAndNil(Answers[i]);
  for i := 1 to 6 do
    FreeAndNil(Questions[i]);
end;

```

Продолжение приложения Б

```
procedure TInputInfo.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  case Key of
    VK_PRIOR:      SB_Anketa.VertScrollBar.Position      :=
SB_Anketa.VertScrollBar.Position - 70;
    VK_NEXT:      SB_Anketa.VertScrollBar.Position      :=
SB_Anketa.VertScrollBar.Position + 70;
  end;
end;

procedure TInputInfo.FormKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  case Key of
    VK_PRIOR:      SB_Anketa.VertScrollBar.Position      :=
SB_Anketa.VertScrollBar.Position - 70;
    VK_NEXT:      SB_Anketa.VertScrollBar.Position      :=
SB_Anketa.VertScrollBar.Position + 70;
  end;
end;

procedure TInputInfo.SB_AnketaMouseWheelDown(Sender: TObject;
  Shift: TShiftState; MousePos: TPoint; var Handled: Boolean);
begin
  SB_Anketa.VertScrollBar.Position      :=
SB_Anketa.VertScrollBar.Position + 70;
end;

procedure TInputInfo.SB_AnketaMouseWheelUp(Sender: TObject;
  Shift: TShiftState;
  MousePos: TPoint; var Handled: Boolean);
begin
  SB_Anketa.VertScrollBar.Position      :=
SB_Anketa.VertScrollBar.Position - 70;
end;

procedure TInputInfo.tbClearClick(Sender: TObject);
```

```

var
  i : integer;
begin
  for i := 1 to CountAnswer do
    Answers[i].SpinEdit.Value := 0;
    SE_CountResp.Value := 1;
    DTP_DateBegin.SetFocus;
  end;
procedure TInputInfo.tbCloseClick(Sender: TObject);
begin
  Close;
end;
procedure TInputInfo.tbFinishTestClick(Sender: TObject);
begin
  FMainForm.MICalcResult.Click;
  ModalResult := mrOk;
end;
procedure TInputInfo.tbSaveClick(Sender: TObject);
var
  i : Integer;
begin
  ADOQ_SaveAnswers.Active := False;
  ADOQ_SaveAnswers.SQL.Clear;
  ADOQ_SaveAnswers.SQL.Text := 'SELECT count(*) as RowCount FROM
YearResp WHERE AnkYear = '
+ FormatDateTime('YYYY', DTP_DateBegin.DateTime);
  ADOQ_SaveAnswers.Active := True;
  if ADOQ_SaveAnswers.FieldByName('RowCount').AsInteger = 0 then
    begin //Добавление новых данных по количеству человек за год
      ADOQ_SaveAnswers.Active := False;
      ADOQ_SaveAnswers.SQL.Text := 'INSERT INTO YearResp
(AnkYear, AllResp, AllWorkers, DateBegin, DateEnd) '
+ 'VALUES ('
+ FormatDateTime('YYYY', DTP_DateBegin.DateTime) + ', '

```

```

+ IntToStr(SE_CountResp.Value) + ', '
  + IntToStr(SE_CountWorkers.Value) + ', '
  + '''' + DateToStr(DTP_DateBegin.DateTime) + ''', '
  + '''' + DateToStr(DTP_DateEnd.DateTime) + ''')';
  ADOQ_SaveAnswers.ExecSQL;
  for i := 1 to CountAnswer do
    begin // Подсчет и добавление новых данных в базу
результатов
      ADOQ_SaveAnswers.SQL.Text := 'INSERT INTO results
(AnkYear, IDA, CountResp, PercResp) '
      + 'VALUES ('
      + FormatDateTime('YYYY', DTP_DateBegin.DateTime) + ', '
      + IntToStr(Answers[i].IDA) + ', '
      + IntToStr(Answers[i].SpinEdit.Value) + ', '
      + FloatToStr(Answers[i].SpinEdit.Value / SE_CountResp.Value *
100) + ')';
      ADOQ_SaveAnswers.ExecSQL;
    end;
    ShowMessage('Данные успешно сохранены в БД!');
  end
else
  begin //Вывод предупреждения и вопроса пользователю
    if MessageDlg('Данные за ' + FormatDateTime('YYYY',
DTP_DateBegin.DateTime) + ' год уже есть в БД. Перезаписать?',
mtWarning, [mbYes, mbNo], 0) = 6
      then
        begin //Удаление сохраненных данных в базе данных
          ADOQ_SaveAnswers.Active := False;
          ADOQ_SaveAnswers.SQL.Text := 'DELETE FROM results
WHERE AnkYear = '
          + FormatDateTime('YYYY', DTP_DateBegin.DateTime);
          ADOQ_SaveAnswers.ExecSQL;
          ADOQ_SaveAnswers.SQL.Text := 'DELETE FROM YearResp WHERE AnkYear = '
          + FormatDateTime('YYYY', DTP_DateBegin.DateTime);

```

Продолжение приложения Б

```
ADOQ_SaveAnswers.ExecSQL; //Запись новых данных в базу
      ADOQ_SaveAnswers.SQL.Text := 'INSERT INTO YearResp
(AnkYear, AllResp, AllWorkers) '
      + 'VALUES ('
      + FormatDateTime('YYYY', DTP_DateBegin.DateTime) + ', '
      + IntToStr(SE_CountResp.Value) + ', '
      + IntToStr(SE_CountWorkers.Value) + ')';
      ADOQ_SaveAnswers.ExecSQL;
      for i := 1 to CountAnswer do
        begin //Запись новых данных в базу
          ADOQ_SaveAnswers.SQL.Text := 'INSERT INTO results (AnkYear, IDA,
CountResp, PercResp) '
          + 'VALUES ('
          + FormatDateTime('YYYY', DTP_DateBegin.DateTime) + ', '
          + IntToStr(Answers[i].IDA) + ', '
          + IntToStr(Answers[i].SpinEdit.Value) + ', '
          + FormatFloat('0.0', Answers[i].SpinEdit.Value /
SE_CountResp.Value * 100) + ')';
          ADOQ_SaveAnswers.ExecSQL;
        end;
        ShowMessage('Данные успешно сохранены в БД!');
      end;
    end;
end;
Constructor TAnswer.Create(AOwner: TComponent);
begin
  with self do
    begin
      MainPanel := TPanel.Create(AOwner);
      MainPanel.Parent := TWinControl(AOwner);
      EditPanel := TPanel.Create(MainPanel);
      SpinEdit := TSpinEdit.Create(EditPanel);
      LAnswer := TLabel.Create(MainPanel);
      SpinEdit.Parent := EditPanel;
```

```
EditPanel.Parent := MainPanel;
    LAnswer.Parent := MainPanel;
    MainPanel.Align := alTop;
    MainPanel.Color := clWhite;
    MainPanel.Height := 50;
    MainPanel.Visible := False;
    MainPanel.Top := 9999;
    EditPanel.Align := alRight;
    EditPanel.Width := 82;
    EditPanel.Height := 34;
    EditPanel.Color := clWhite;
    EditPanel.BevelOuter := bvNone;
    SpinEdit.Top := 2;
    SpinEdit.Left := 9;
    SpinEdit.Font.Size := 13;
    SpinEdit.Width := 49;
    SpinEdit.Height := 31;
    LAnswer.Align := alClient;
    LAnswer.Font.Size := 13;
    LAnswer.WordWrap := True;
    LAnswer.Layout := tlCenter;
end;
end;
Constructor TQuestion.Create(AOwner: TComponent);
begin
    with self do
        begin
            PHeader := TPanel.Create(AOwner);
            PHeader.Parent := TWinControl(AOwner);
            LQuestion := TLabel.Create(PHeader);
            LQuestion.Parent := PHeader;
            PHeader.Align := alTop;
            PHeader.Color := $009BF592;
            PHeader.Height := 57;
```



```
PHeader.Visible := False;  
    PHeader.Top := 9999;  
    LQuestion.Align := alClient;  
    LQuestion.Font.Size := 13;  
    LQuestion.WordWrap := True;  
    LQuestion.Layout := tlCenter;  
    LQuestion.Font.Style := [fsBold];  
end;  
end;  
end.
```

ПРИЛОЖЕНИЕ В. КОМПАКТ-ДИСК

Содержание:

1. Пояснительная записка к выпускной квалификационной работе.
2. Разработанный программный продукт.
3. Презентация.