



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ  
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Прикладной информатики

**БАКАЛАВРСКАЯ РАБОТА**

На тему Разработка информационной системы учета успеваемости и посещаемости студентов (на примере ГБПОУ КК «Туапсинский гидрометеорологический техникум»)

Исполнитель Зубарев Алексей Игоревич

(фамилия, имя, отчество)

Руководитель кандидат технических наук

(ученая степень, ученое звание)

Слесарева Людмила Сергеевна

(фамилия, имя, отчество)

«К защите допускаю»

Заведующий кафедрой

(подпись)

кандидат технических наук  
Слесарева Людмила Сергеевна

«22» 06 2016г.

Санкт-Петербург  
2016г



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ  
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Прикладной информатики

**БАКАЛАВРСКАЯ РАБОТА**

На тему Разработка информационной системы учета успеваемости и посещаемости студентов (на примере ГБПОУ КК «Туапсинский гидрометеорологический техникум»)

Исполнитель Зубарев Алексей Игоревич  
(фамилия, имя, отчество)

Руководитель кандидат технических наук  
(ученая степень, ученое звание)  
Слесарева Людмила Сергеевна  
(фамилия, имя, отчество)

«К защите допускаю»  
Заведующий кафедрой

(подпись)

кандидат технических наук  
Слесарева Людмила Сергеевна

«\_\_» \_\_\_\_\_ 20\_\_ г.

Санкт–Петербург  
2016г

## ОГЛАВЛЕНИЕ

|                                                                                                              |           |
|--------------------------------------------------------------------------------------------------------------|-----------|
| <b>ВВЕДЕНИЕ.....</b>                                                                                         | <b>3</b>  |
| <b>1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ .....</b>                                                 | <b>5</b>  |
| 1.1    Общая характеристика ГБПОУ КК «Туапсинский гидрометеорологический техникум» .....                     | 5         |
| 1.2    Организация информационного обеспечения ГБПОУ КК Туапсинского гидрометеорологического техникума ..... | 9         |
| 1.3    Уровень автоматизации бизнес-процессов в настоящее время .....                                        | 11        |
| <b>2    МЕТОДОЛОГИЯ    АВТОМАТИЗАЦИИ    ПРОЦЕССА СИСТЕМАТИЗАЦИИ ПОКАЗАТЕЛЕЙ УСПЕВАЕМОСТИ СТУДЕНТОВ .....</b> | <b>14</b> |
| 2.1    Анализ существующих подходов и разработок по автоматизации процесса .....                             | 14        |
| 2.2    Выбор и обоснование стратегии автоматизации .....                                                     | 15        |
| 2.3    Описание выбранного средства программирования .....                                                   | 23        |
| <b>3 РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....</b>                                                              | <b>26</b> |
| 3.1    Описание технического задания.....                                                                    | 26        |
| 3.2    Реализация базы данных .....                                                                          | 28        |
| 3.3    Программная реализация информационной системы.....                                                    | 34        |
| 3.4    Работа с программой.....                                                                              | 39        |
| 3.5    Обоснование экономической эффективности.....                                                          | 42        |
| <b>ЗАКЛЮЧЕНИЕ .....</b>                                                                                      | <b>53</b> |
| <b>СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....</b>                                                                 | <b>55</b> |
| <b>ПРИЛОЖЕНИЯ .....</b>                                                                                      | <b>57</b> |

## ВВЕДЕНИЕ

**Актуальность темы исследования.** В настоящее время растет интерес к передовым системам, которые могут помочь более эффективно организовать управление деловыми процессами на предприятии. Для управления ими на предприятии широко используют компьютерные информационные системы, которым делегируются формализуемые функции по ведению различных бизнес процессов.

Так же актуальность темы исследования обусловлена увеличением численности количества преподавателей, внештатных сотрудников, студентов учебного заведения, учебных модулей, дисциплин и возникает потребность в создании и использовании автоматизированной системы обработки данных (АСОД) для структурного подразделения учебная часть. В связи с актуальностью данной проблемы было решено провести исследования в данной области, разработать автоматизированную систему обработки данных структурного подразделения учебная часть и рассмотреть вопросы её внедрения в эксплуатацию.

**Объект исследования:** учебная часть ГБПОУ КК «Туапсинского гидрометеорологического техникума».

**Предмет исследования** является процесс учета успеваемости и посещаемости студентов.

**Целью работы** является разработка информационной системы учета успеваемости и посещаемости студентов (на примере ГБПОУ КК «Туапсинский гидрометеорологический техникум»).

В результате поставленной цели решаются следующие **задачи:**

1. Анализ предметной области;
2. Анализ существующих подходов и разработок по автоматизации процесса;
3. Определение стратегии автоматизации;
4. Реализация базы данных;

5. Программная реализация информационной системы;
6. Обоснование экономической эффективности.

Указанные цели и задачи определили логику изложения материала и **структуру** работы, состоящую из трех разделов.

В первом разделе дана общая характеристика и организация информационного обеспечения в ТГМТ, а также уровень автоматизации бизнес-процессов.

Во втором разделе освещается анализ существующих подходов и разработок по автоматизации процесса, обоснование стратегии автоматизации и описание выбранного средства программирования.

В третьем разделе описана реализация базы данных, программная реализация информационной системы, описана работа с программой и обоснована экономическая эффективность.

**Теоретической и методической** основой для выполнения бакалаврской работы послужили работы таких авторов, как М.Э. Абрамян, С.С. Михалкович, Н.И. Амелина, Е.М. Королева, О.Л. Голицына и др.

Бакалаврская работа содержит 99 страниц печатного текста, 31 рисунок, 3 таблицы.

# **1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ**

## **1.1 Общая характеристика ГБПОУ КК «Туапсинский гидрометеорологический техникум»**

Наименование предприятия или организации: Государственное бюджетное профессиональное образовательное учреждение Краснодарского края «Туапсинский гидрометеорологический техникум».

Образовательную деятельность техникум осуществляет на основе лицензии, а также Устава техникума. Учебное заведение может вести образовательную деятельность по следующим специальностям: метеорология, экономика и бухгалтерский учет, рациональное использование природоохозяйственных комплексов, информационные системы; а также по программам дополнительного профессионального образования: лаборант химического анализа, охрана труда для руководителей и специалистов предприятий, учреждений, организаций; подготовительные курсы для поступающих в техникум.

Характеристика использования современных информационных технологий в решении задач управления:

1. Учет и контроль. Учетная функция осуществляется непосредственно в бухгалтерском отделе. Контрольная функция осуществляется главным бухгалтером, заместителем директора, непосредственно директором: контроль выполнения планов, расходования материальных ресурсов, использования финансовых средств.

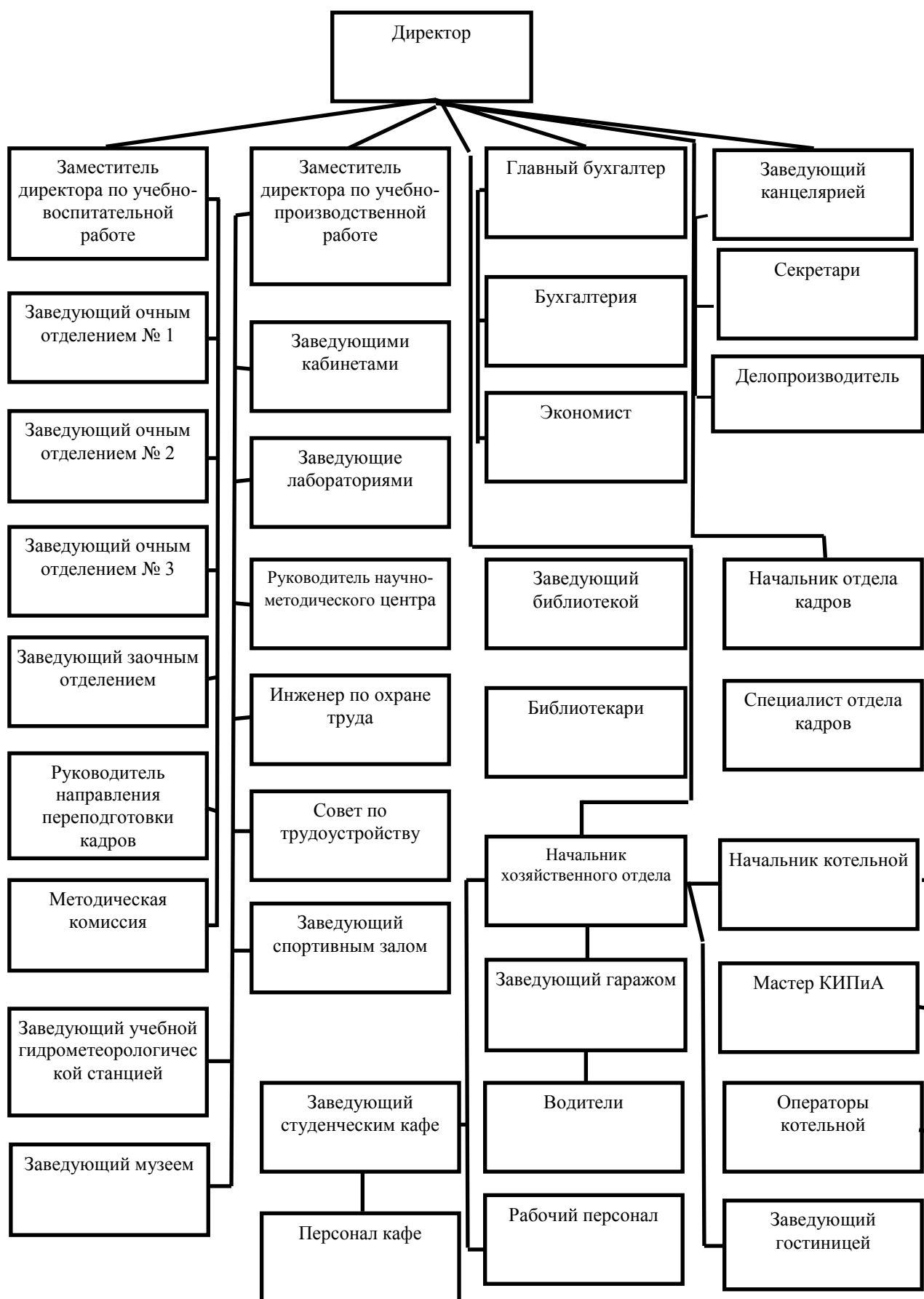
2. Анализ осуществляется по двум направлениям: анализ учебной деятельности, которым занимается учебный отдел и анализ хозяйственной деятельности, который выполняется различными специалистами в зависимости от сложности и уровня анализируемого объекта или процесса. Анализ результатов хозяйственной деятельности фирмы за год и более проводит главный бухгалтер учебного заведения совместно со специалистом-экономистом.

3. Автоматизация документооборота. Документооборот на предприятии смешанный, для подготовки электронных документов чаще всего используются такие программные продукты, как пакет Microsoft Office, Adobe Acrobat Reader DC, встроенный бесплатный почтовый клиент Outlook Express, свободный файловый архиватор 7Zip, 1С: Колледж — удачное средство, чтобы управлять деятельностью учреждений начального и среднего профессионального образования, позволяющее автоматизировать работу существенных сегментов колледжа – административного отдела, отдела кадров, приемной комиссии, деканата, учебно-методического отдела [6, с.142].

В организационную структуру предприятия включают (рисунок 1):

- учебная часть;
- курсы по охране труда;
- учебные лаборатории и кабинеты;
- учебная гидрометеорологическая станция;
- учебная бухгалтерия; учебная мастерская;
- учебная гостиница;
- библиотека с помещением для читального зала;
- электронная библиотека; музей истории техникума;
- научно-методический центр;
- спортивный и тренажерный залы;
- студенческое кафе;
- котельная;
- гаражи.

Для обеспечения образовательной деятельности в техникуме выделяются 3 очных отделения и одно заочное: отделение №1: метеорология, рациональное использование природохозяйственных комплексов; отделение №2: экономика и бухгалтерский учет, информационные системы; отделение №3: рациональное использование природохозяйственных комплексов, экономика и бухгалтерский учет.



**Рис.1 Организационно-управленческая структура предприятия<sup>1</sup>**

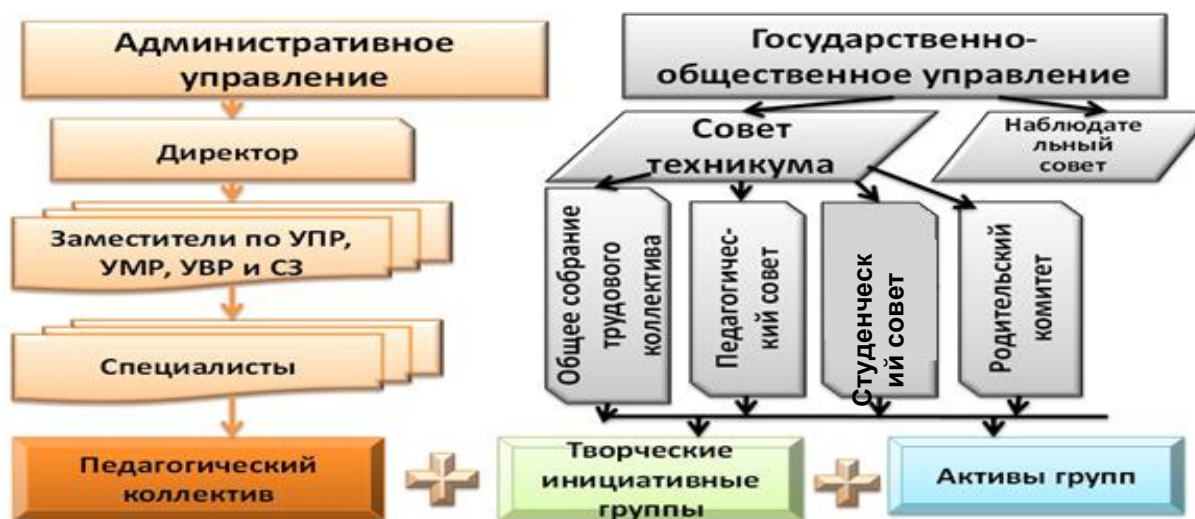
<sup>1</sup> Рисунок составлен автором на основе данных ГБПОУ КК «ТГМТ»



Заочное отделение: экономика и бухгалтерский учет.

Высшим руководящим органом техникума является выборный Совет учебного заведения, возглавляемый директором.

Ведение учебного процесса, практическую подготовку учеников, методическую и воспитательную работу проводит педагогический совет техникума под руководством директора и его заместителей (рисунок 2).



**Рис. 2 Модель организационной структуры управляющей системы техникума<sup>2</sup>**

Непосредственное управление всей деятельностью техникума осуществляет его директор, прошедший аттестацию в Министерстве науки и образования Краснодарского края.

Руководство учебной, воспитательной, методической, административно-хозяйственной и другими видами работ осуществляют заместители директора техникума, назначаемые на должность директором, заключившие или переведенные на работу по срочному трудовому договору (контракту).

Кадрово-правовое управление состоит из следующих подразделений: отдел кадров, общий отдел, которые имеют печать с наименованием. В состав кадрово-правового управления входят также юрисконсульт, инженер по охране

<sup>2</sup>Рисунок составлен автором на основе собственного исследования

труда и специалист по гражданской обороне и чрезвычайным ситуациям (ГОЧС).

Специалист отдела кадров руководит работой отдела кадров и подчиняется начальнику кадрово-правового управления. Начальник отдела кадров назначается на должность директором техникума по представлению начальника кадрово-правового управления. Непосредственное руководство деятельностью общего отдела исполняет руководитель общего отдела, подчиняющийся начальнику кадрово-правового управления. Начальник общего отдела назначается на должность директором техникума по представлению начальника кадрово-правового управления.

## **1.2 Организация информационного обеспечения ГБПОУ КК Туапсинского гидрометеорологического техникума**

Основной терминопределяющий процесс информатизации в каждой образовательной организации – это «информационно-образовательная среда». Ее можно охарактеризовать как системно-организованную совокупность информационного, технического, учебно-методического обеспечения, имеющая связь с человеком как субъектом образовательного процесса.

В общем информационно-образовательную среду можно охарактеризовать следующим:

- присутствием организационной структуры, копирующей и сохраняющей информационные ресурсы, а также оказывающей информационные услуги;
- внешними и внутренними информационными ресурсами, которые ориентируются на разнообразные сегменты клиентов;
- информационной образованностью преподавателей и учеников;
- локальной сетью и нынешним лицензионным программным обеспечением;
- использованием последних информационных технологий.

электронные каталоги, доступ в глобальную сеть Интернет, высокочастотные каналы связи с глобальной сетью[11, с.215].

В техникуме используются следующие информационные ресурсы и источники:

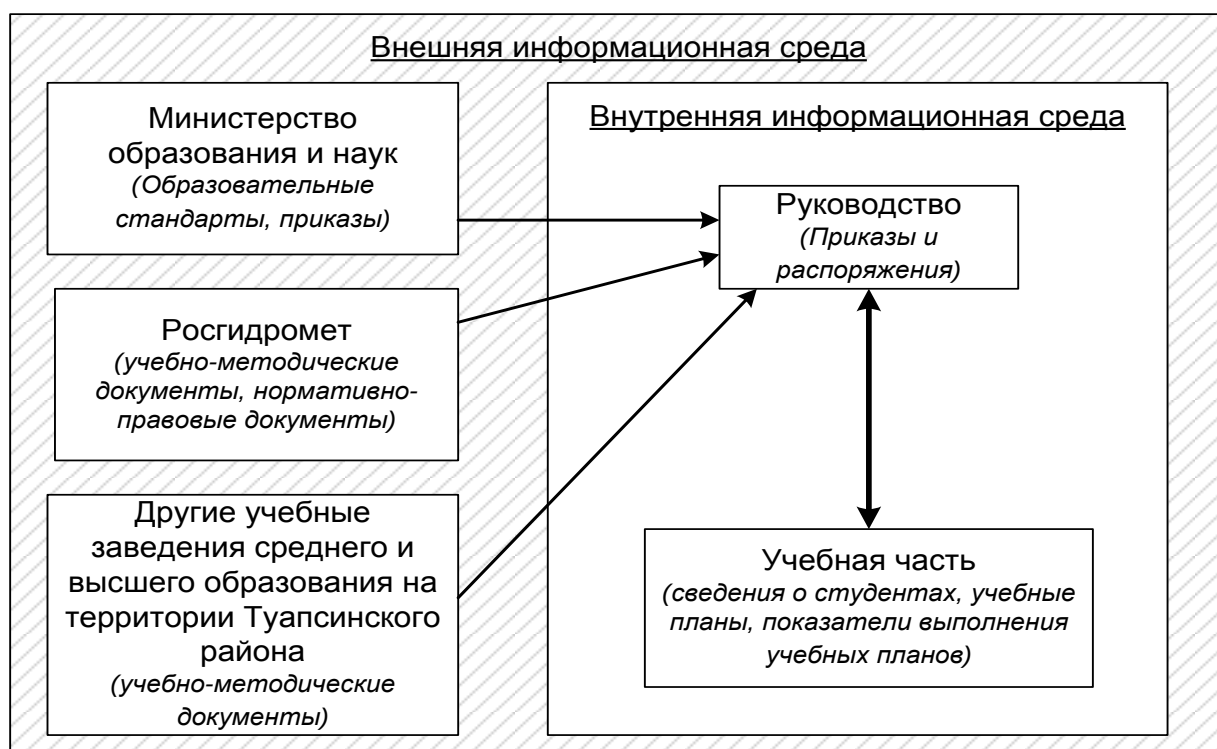
Внутренние источники информации: руководство техникума, учебный отдел, бухгалтерия, отдел кадров и т.д.

Внутренняя информация: данные бухгалтерского учета, информация о студентах, учебные планы, показатели выполнения учебного плана.

Источники внешней информации: Росгидромет, Министерство науки и образования Краснодарского края, другие учебные заведения среднего и высшего образования на территории Туапсинского района.

Внешняя информация: учебно-методические документы, нормативно-правовые документы.

На рисунке 3 представлены основные информационные потоки связанные с видом деятельности «Предоставление образовательных услуг».



**Рис.3 Схема информационных потоков<sup>3</sup>**

<sup>3</sup>Рисунок составлен автором на основе данных ГБПОУ КК «ТГМТ»

Значительное место в информационном обеспечении образовательного процесса и управления выступают наличие и качество информационно-методического снабжения внутренних и внешних сетевых ресурсов [18, с.168].

Внутренние сетевые ресурсы изображены локальной сетью образовательного учреждения, обеспечивающая информационно-справочную функцию, т.е. снабжает рабочие места каждого структурных подразделений нормативными, отчетными, справочными сведениями по существенным назначениям работы.

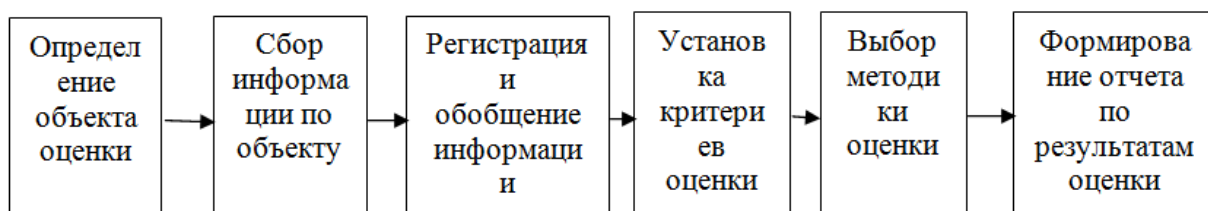
Административная внутренняя сеть кроме справочно-информационной выполняет аналитические функции по проверке учебного процесса и по иным направлениям.

Внешние сетевые ресурсы в учебной организации применяются, сначала, как эффективное средство рекламы образовательного учреждения, далее, как средство передачи педагогического опыта учителей по средствам электронных публикаций статей, докладов, методических разработок.

В качестве аппаратного обеспечения информационных систем в техникуме используются ПК (процессор Intel Core 2 Duo Processor E6750 2.66 GHz, 4 Гб RAM). На ПК установлена операционная система Windows 7 [2, с.9].

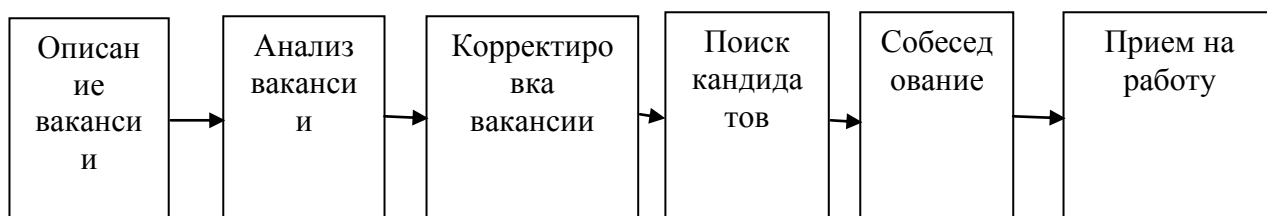
### 1.3 Уровень автоматизации бизнес-процессов в настоящее время

В техникуме осуществляются следующие бизнес-процессы: бухгалтерский учет (рисунок 4), подбор персонала (рисунок 5), управление персоналом (рисунок 6), предоставление образовательных услуг (рисунок 7).

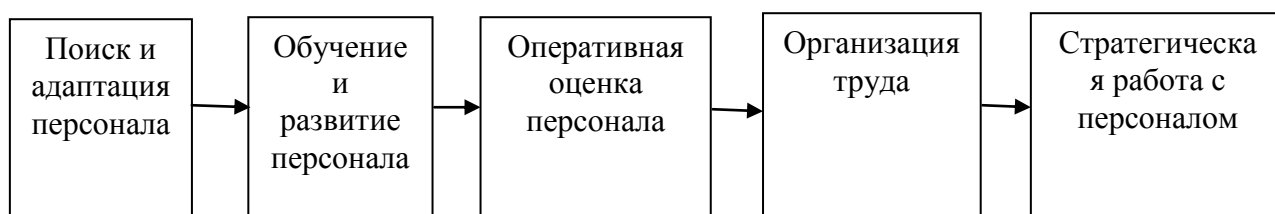


**Рис.4 Бухгалтерский учет<sup>4</sup>**

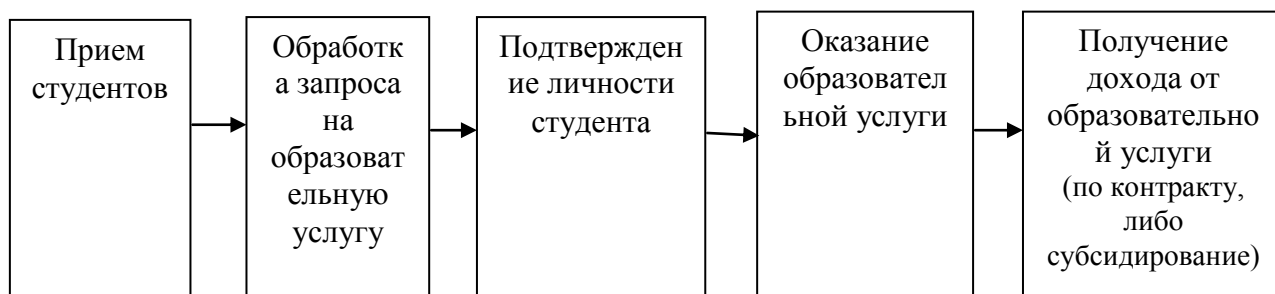
<sup>4</sup>Рисунок составлен автором на основе данных ГБПОУ КК «ТГМТ»



**Рис.5 Подбор персонала<sup>5</sup>**



**Рис.6 Управление персоналом<sup>6</sup>**



**Рис.7 Предоставление образовательных услуг<sup>7</sup>**

В ходе исследования бизнес-процессов Туапсинского Гидрометеорологического техникума было обнаружено, что в данной организации не автоматизирована систематизация знаний студентов, которая является частью бизнес-процесса «Предоставление образовательных услуг».

Государство выдаёт техникуму разрешение на проведение образовательной деятельности и осуществляет финансирование этой деятельности на основании оценки результативности учебного процесса в техникуме. Одним из важнейших показателей оценки результативности учебной деятельности в техникуме является качество обучения, которое включает в себя такие показатели как успеваемость студентов и посещаемость

<sup>5</sup>Рисунок составлен автором на основе данных ГБПОУ КК «ТГМТ»

<sup>6</sup> То же

<sup>7</sup> То же

ими занятий. Итоги успеваемости студентов и посещаемости ими занятий определяется каждый семестр (учебное полугодие). Успеваемость студентов оценивается по пятибалльной шкале по каждому изучаемому ими предмету, а посещаемость занятий учитывается в виде количества часов аудиторных учебных занятий, пропущенных ими по уважительным и неуважительным причинам [17, с.125].

По итогам каждого семестра проводятся совещания педагогического коллектива с обязательным обсуждением и рассмотрением результатов оценки качества обучения. Также такие совещания проводятся в середине семестра по итогам предварительной аттестации студентов. Во время проведения этих совещаний существует необходимость наглядного представления итогов оценки качества обучения студентов. Помимо подведения итогов оценки качества обучения, с целью введения соревновательного подхода в процессе обучения, по результатам оценки качества обучения группам студентов присваиваются места, а особо отличившиеся студенты выносятся для дополнительного поощрения. Подведением итогов оценки качества обучения занимается подразделение техникума под названием учебная часть. Ввиду крайней важности проведения оценки качества обучения в таком виде, появилась значимость автоматизации этого типа деятельности учебной части техникума.

Таким образом, была сформулирована цель — разработать информационную систему систематизации показателей успеваемости учеников для улучшения учебного процесса [20, с.87].

## **2 МЕТОДОЛОГИЯ АВТОМАТИЗАЦИИ ПРОЦЕССА СИСТЕМАТИЗАЦИИ ПОКАЗАТЕЛЕЙ УСПЕВАЕМОСТИ СТУДЕНТОВ**

### **2.1 Анализ существующих подходов и разработок по автоматизации процесса**

В техникуме внедрена система автоматизации образовательного процесса «1С: Колледж ПРОФ», данная система характеризуется как совокупность мер для управления деятельностью учреждений начального и среднего профессионального образования и охватывает каждый уровень управленческой деятельности главных сегментов колледжа.

«1С: Колледж ПРОФ» помогает автоматизировать рабочие места председателя и секретаря приемной комиссии, абитуриента, заведующего отделением, заместителя директора по учебной работе, по воспитательной работе, по производственному обучению, классного руководителя (куратора), преподавателя, секретаря стипендиальной комиссии, бухгалтера по расчетам со студентами, заведующего ЦМК (ПЦК). Также программный продукт снабжает работу с ФБС ЕГЭ, печать дипломов и приложений на бланках Гознака, поддержку ФГОС-3, создание регламентированных документов, работу с сайтом, подготовку информационных рассылок работникам и ученикам.

«1С: Колледж ПРОФ» выполнен на платформе «1С: Предприятие 8.2». Полный функционал решения доступен как в веб-, так и в тонком клиенте, поддерживается многопользовательская работа в локальной сети или в глобальной сети Интернет.

«1С: Колледж ПРОФ» содержит базу данных сведений, необходимых для автоматизации информационной деятельности данного предприятия, но имеющиеся возможности по анализу успеваемости студентов и посещаемости ими учебных занятий не соответствуют требованиям техникума по анализу этих показателей. В связи с этим, возникла необходимость создания отдельной информационной системы для решения данной задачи, по оценке качества обучения студентов техникума [4, с.563-564].

## 2.2 Выбор и обоснование стратегии автоматизации

Анализ данных в сфере образования—новоиспеченная и злободневная часть науки, в которой бытует масса задач и вопросов для анализа и исследования, которые следует решить. Благодаря значительному объему сведений и данных в образовательном процессе (базы данных учеников с информацией об успеваемости, прошлом обучении или образовании, программой курсов, дистанционные системы обучения и т.п.) возможно разрешить массу задач, например, таких как понимание учащихся, усовершенствование качества лекционных занятий, снижение издержек на организацию процесса обучения [5, с.214].

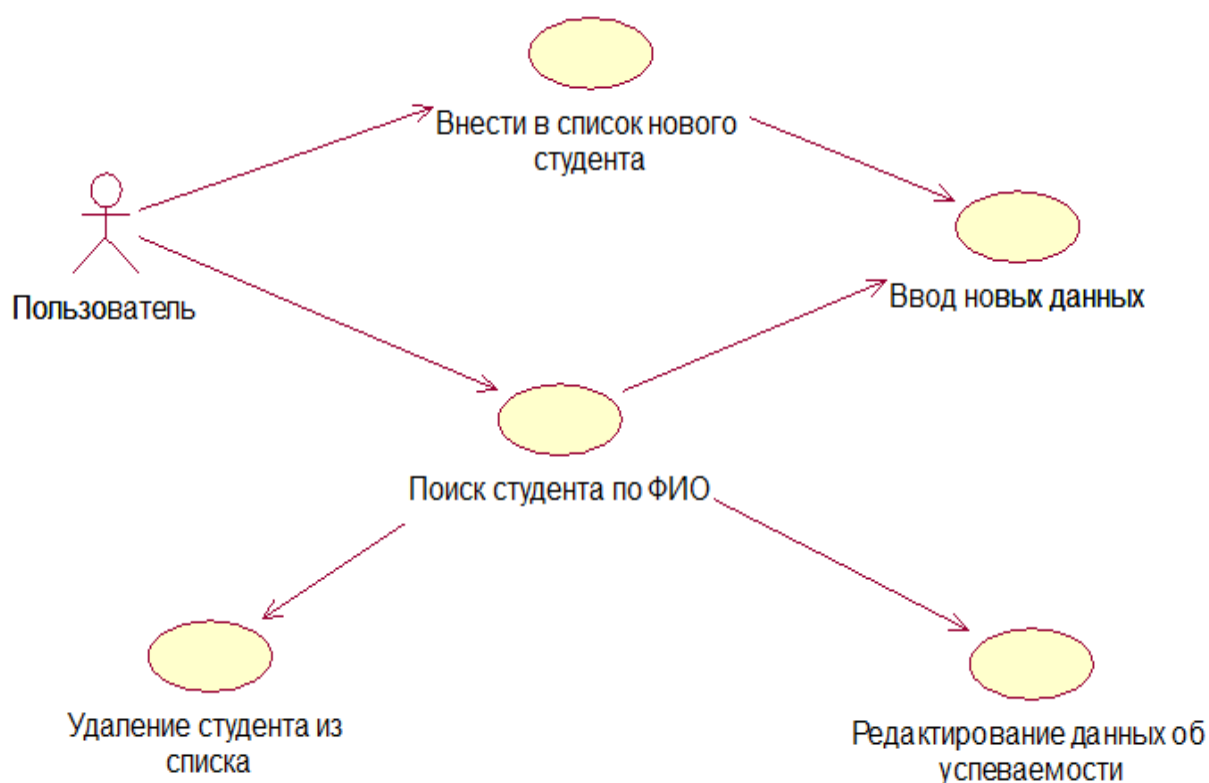
Результативное управление базой данных учащихся невыполнимо без системы автоматизации. Информационная система существует для ведения личных дел учащихся техникума и работает отдельно или в составе ИС. Такая система позволяет автоматизировать:

- управление учебными группами и специальностями, включая образование некоторых списков групп на каждый учебный год;
- организация электронных личных дел учащихся;
- перевод студентов в другую группу, зачисление, отчисление;
- расчет итогового рейтинга студентов за обусловленный отрезок времени;
- поиск студентов в базе данных;
- получение сводных данных по контингенту студентов и формирование отчетов;
- получение статистики для подготовки отчетов [15, с.98].

Для отображения процесса автоматизации были построены диаграммы в CASE среде проектирования и разработки информационных систем и программного обеспечения для управления организациями – Rational Rose, а именно: диаграмма прецедентов, диаграмма классов, диаграмма последовательности действий и диаграмма взаимодействия.



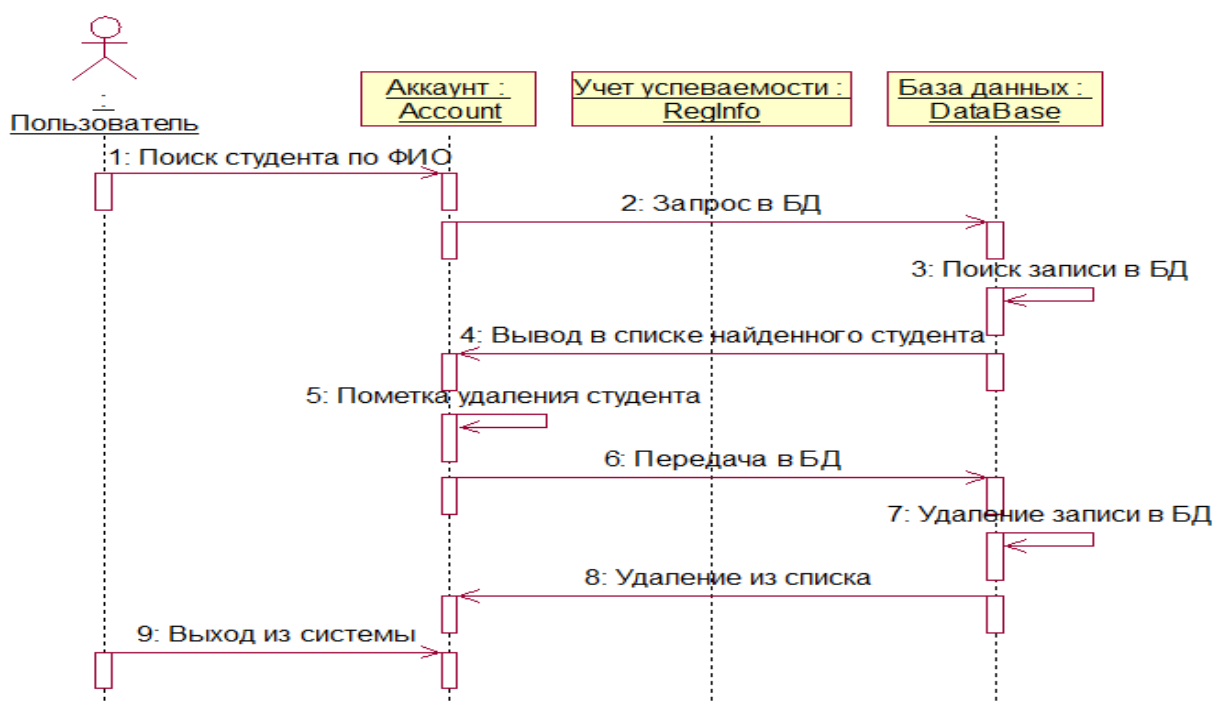
Для организации диаграммы вариантов применения следует найти действующие лица, которые будут иметь отношение к работе подсистемы. Для такой информационной подсистемы действующим лицом будет пользователь [16, с.173]. Диаграмма последовательности отображает поток событий, происходящих в рамках варианта использования (рисунок 8).



**Рис.8 Диаграмма прецедентов информационной подсистемы<sup>8</sup>**

При работе с базой у пользователя есть выбор между двумя основными функциями: добавить запись о новом студенте, либо поиск информации о занесенном в базе. После занесения информации об учащимся программа предложит добавить следующую запись. Если же запись во время поиска не была найдена, то пользователю будет предложено добавить информацию о студенте самостоятельно. С выбранной записью можно осуществить следующие операции: удалить запись полностью и все связанные с ней данные или отредактировать требуемые поля и/или связанные значения[19, с.132].

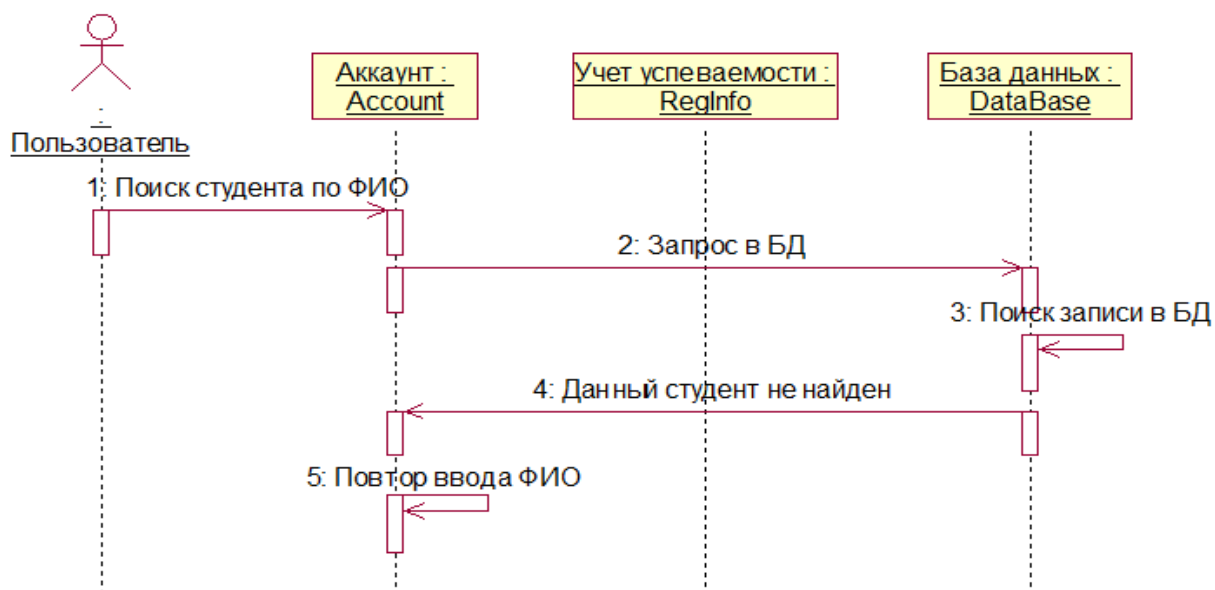
<sup>8</sup>Рисунок составлен автором на основе собственного исследования



**Рис. 9 Диаграмма последовательности для варианта использования «Удаление студента из списка»<sup>9</sup>**

Рассмотрим подробнее происходящее на диаграмме. После выбора пользователем операции поиска и ввода интересующей его фамилии, по нажатию соответствующе запрограммированной кнопки объект «Аккаунт» отправит запрос к БД. После этого сработает алгоритм поиска всех записей, содержащих заданное значение в соответствующем разделе БД. Если таких записей несколько (например, полные тезки), то объект «Аккаунт» отобразит их все. Далее, если пользователь решил удалить одну из найденных записей, то сработает следующий алгоритм: по нажатию запрограммированной кнопки в объекте «Аккаунт» вызывается запрос на удаление данных из базы. После этого осуществляется удаление указанной записи в базе и только потом запись удаляется из отображаемых данных в списке объекта «Аккаунт». Если пользователь активировал кнопку «выход», это приведет к закрытию программной среды [21, с.142]. При этом все изменения сохранятся, кроме тех случаев, когда не завершился процесс внесения изменений (например, аварийное отключение электропитания)(рисунок 9).

<sup>9</sup>Рисунок составлен автором на основе собственного исследования

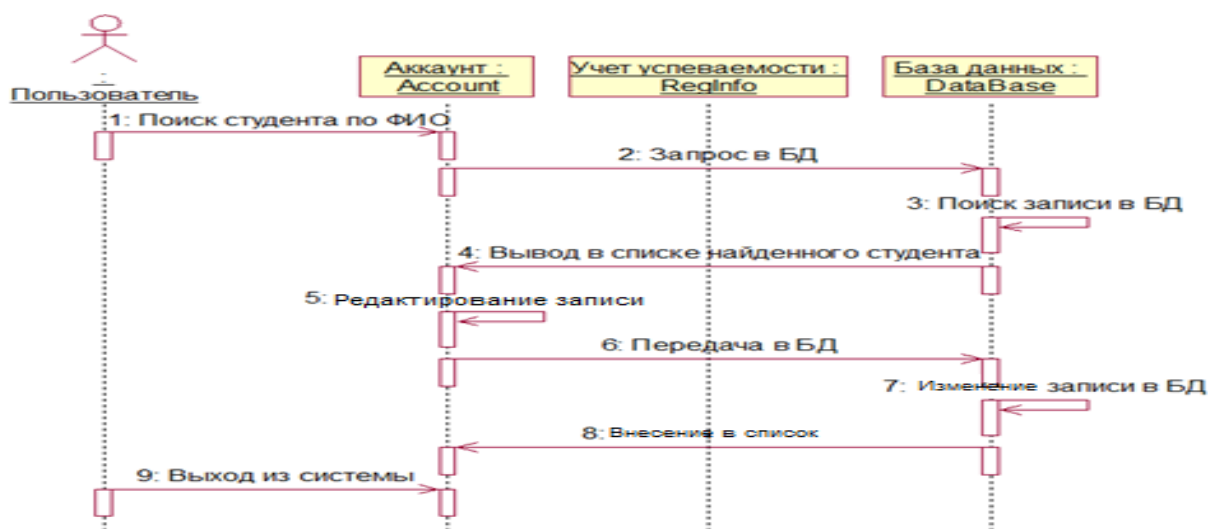


**Рис. 10 Альтернативный поток «Студент не найден» для варианта использования «Поиск студента по ФИО»<sup>10</sup>**

На рисунке 10 изображен альтернативный поток «Студент не найден», появляющийся в том случае, если введенные данные ФИО студента не найдены в базе данных. После ввода ФИО с клавиатуры, объект «Аккаунт» делает запрос базе данных для того, чтобы проверить существование введенных данных ФИО студента, следом осуществляется поиск в базе данных студентов. Если подобных данных нет, то, следовательно, возникает ошибка «данный студент не найден», и пользователю предлагается еще раз ввести данные.

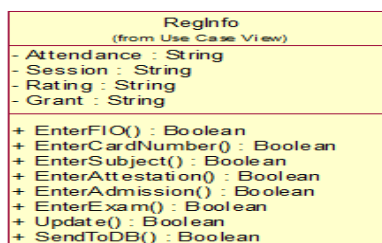
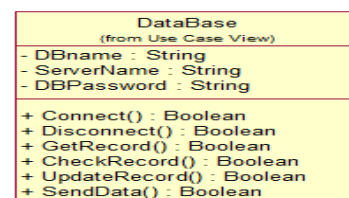
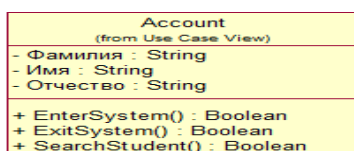
На рисунке 11 представлен информационный поток «Редактирование данных о студенте». До пятой итерации, алгоритм идентичен описанному на рисунке 10. Рассмотрим происходящее именно с этого момента. После того, как в объекте «Аккаунт» был задействован пункт выбора редактирования информации в найденной записи, осуществится переход непосредственно к самому редактированию. Отредактированные данные будут отправлены в БД с запросом на обновление данных. Там произойдут редактирование информации и отправка актуальных данных обратно в объект «Аккаунт» [3, с.185]. Логика операции выход аналогична описанию к рисунку 9.

<sup>10</sup>Рисунок составлен автором на основе собственного исследования



**Рис. 11** Диаграмма последовательности для варианта использования «Редактирование данных»<sup>11</sup>

Этапы формирования диаграммы классов содержатся в установлении классов, нужных для функционирования подсистемы. Образование диаграммы классов представляет собой завершение процесса анализа и начало процесса проектирования. Для разработки информационной системы обособлены следующие классы: Account, DataBase, RegInfo (рисунок 12).



**Рис. 12** Диаграмма классов для информационной подсистемы учета студентов<sup>12</sup>

<sup>11</sup>Рисунок составлен автором на основе собственного исследования

<sup>12</sup>То же

Диаграммы состояний устанавливают любые потенциальные состояния, в которых может находиться конкретный объект, а также процесс замены состояний объекта в итоге прихода отдельных событий. Последний вариант диаграммы состояний представлен на рисунке 13 для класса «Account», на рисунке 14 для «RegInfo» и на рисунке 15 для класса «DataBase».

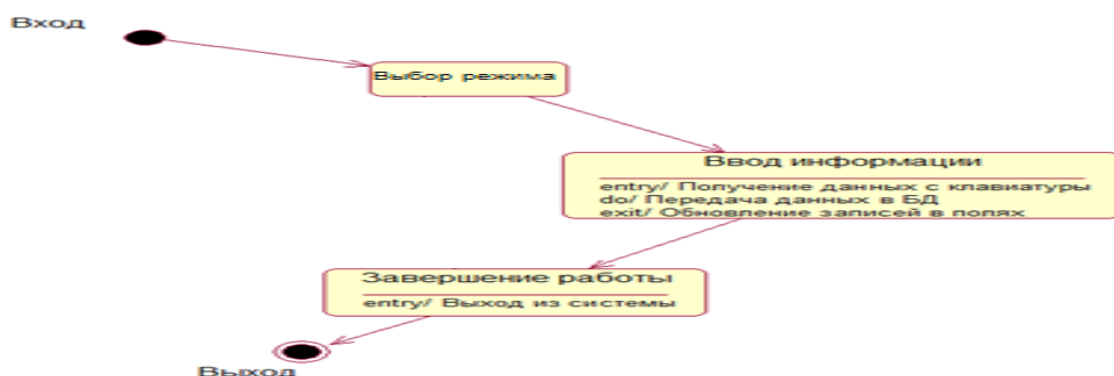


Рис. 13 Диаграмма состояний для класса «Account»<sup>13</sup>

Диаграмма состояний для класса «Account» иллюстрирует его алгоритм работы. За входом в класс (в частности инициализацией пользователя) следует выбор режима и ввод необходимой информации, после чего работа с классом завершается и происходит передача управления выбранному классу: «RegInfo» или «DataBase» [22, с.85].

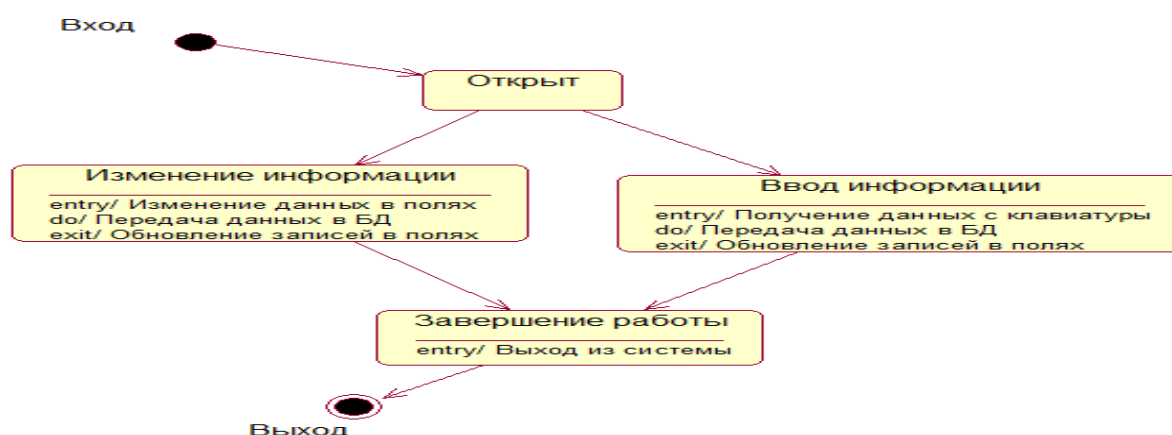
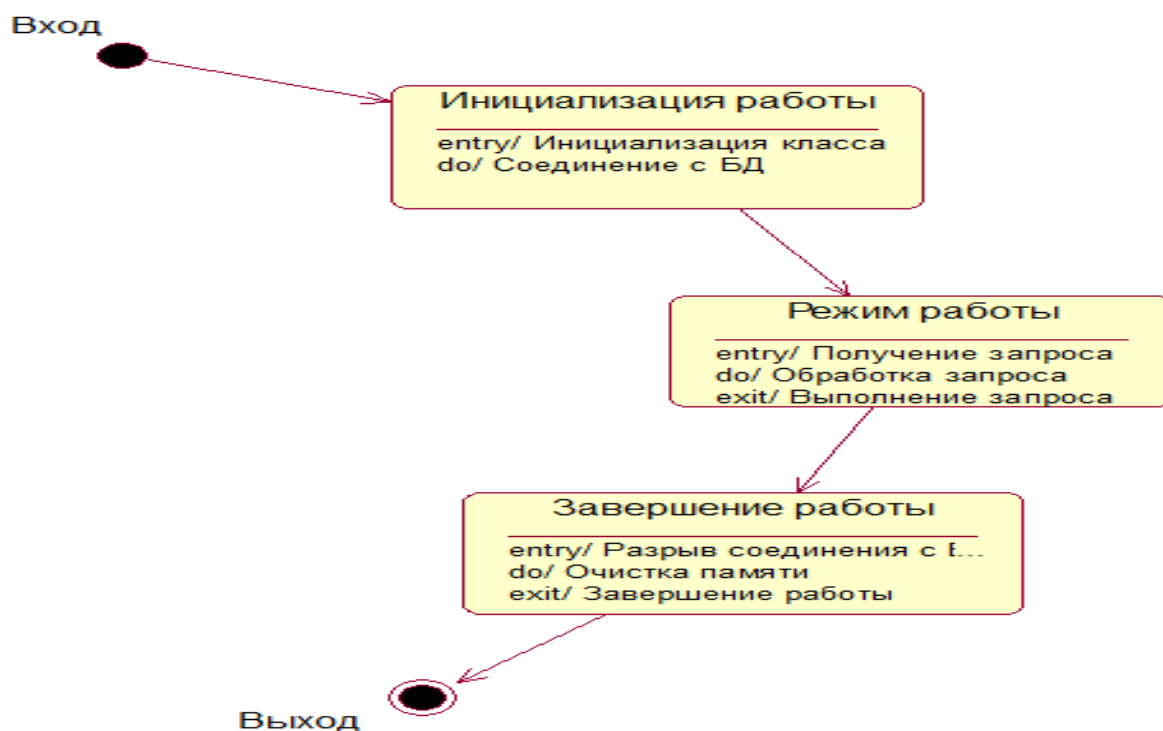


Рис.14 Диаграмма состояний для класса «RegInfo»<sup>14</sup>

<sup>13</sup>Рисунок составлен автором на основе собственного исследования

<sup>14</sup>То же

Выше приведенная диаграмма описывает информационный поток в классе «RegInfo»: вход в класс, открытие искомой записи; затем пользователю следует выбрать необходимое действие – редактирование существующей или создание новой записи. На любом из этих этапов управление временно передается классу «DataBase».

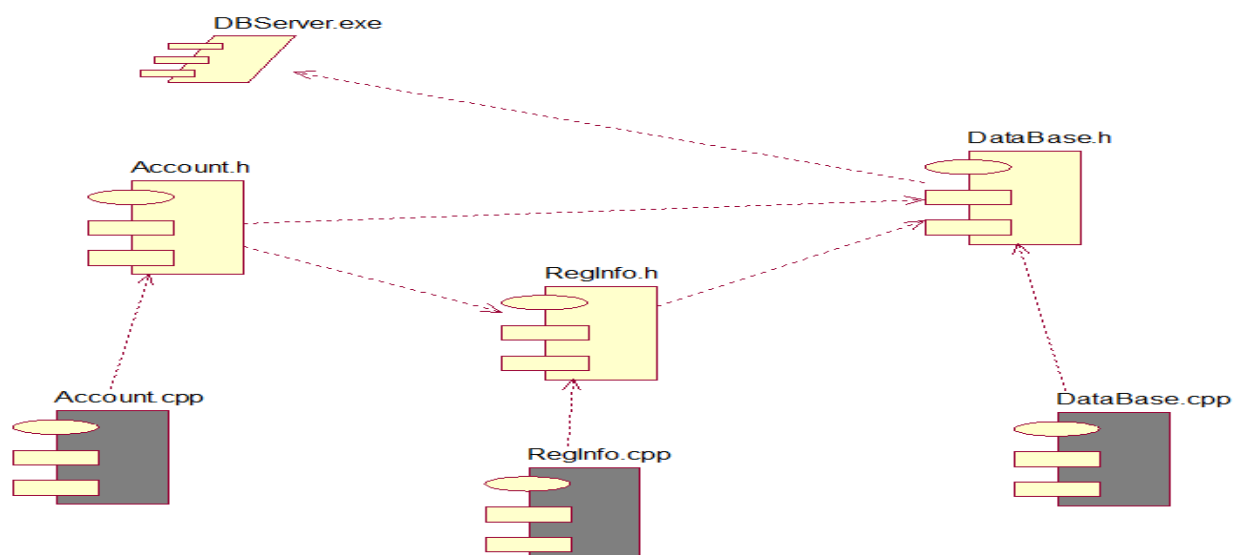


**Рис. 15** Диаграмма состояний для класса «DataBase»<sup>15</sup>

Как только, произошел вход и инициализация класса «DataBase» с установкой соединения с самой базой данных, происходит получение сформированного ранее запроса, обработка информации в соответствии с ним, и непосредственно выполнение самого запроса. После этого мы возвращаемся в класс «RegInfo», записи в полях обновляются, и мы выходим из класса. В классе «DataBase» же происходит отсоединение от базы, очистка памяти и завершение работы [14, с.126].

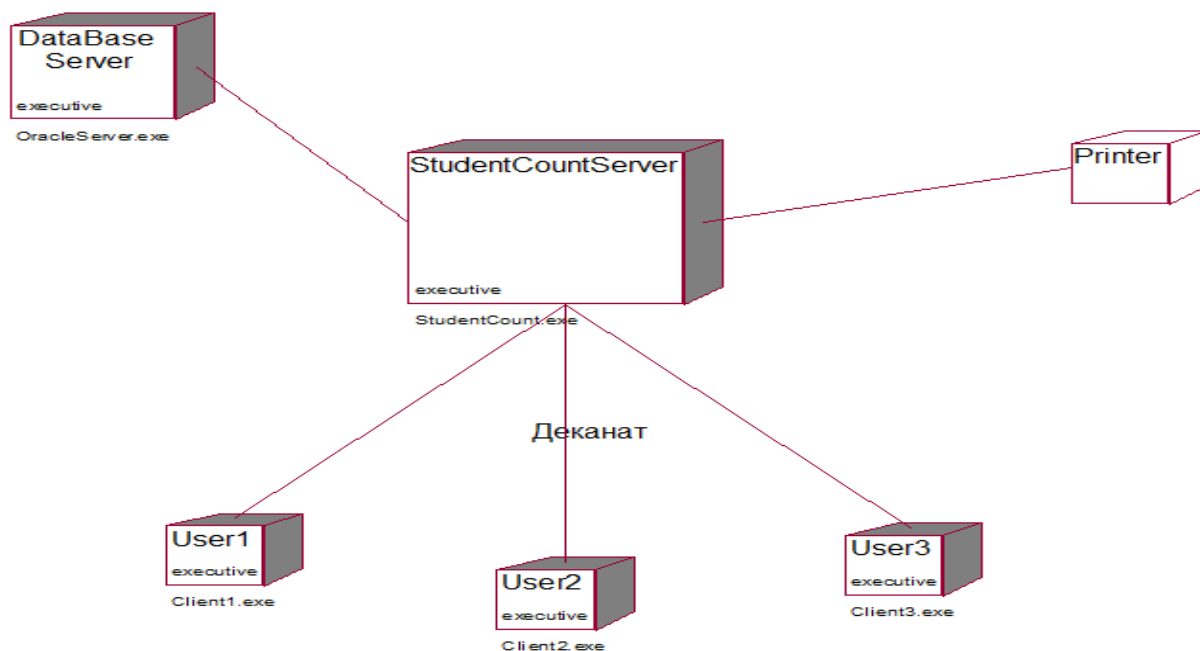
После установления связей следует сопоставить компоненты класса с самим классом. Диаграмма компонентов изображена на рисунке 16.

<sup>15</sup>Рисунок составлен автором на основе собственного исследования



**Рис. 16. Диаграмма компонентов информационной подсистемы**<sup>16</sup>

Диаграмма размещения отражает физическое расположение сети и местонахождение в ней разных компонентов. В данном проекте информационная подсистема для учебной части содержит в себе ряд подсистем, осуществляемых на некоторых физических устройствах, именуемых узлами (node). Конечный вариант диаграммы представлен на рисунке 17.



**Рис. 17. Диаграмма размещения информационной подсистемы**<sup>17</sup>

<sup>16</sup>Рисунок составлен автором на основе собственного исследования

<sup>17</sup>То же

## 2.3 Описание выбранного средства программирования

В результате обследования аппаратно-программной базы заказчика и предполагаемых условий эксплуатации данного проекта, было принято решение использовать для создания программы интегрированную среду разработки Borland Turbo Delphi.

Программный продукт Borland Turbo Delphi предназначен для RAD-разработки программного обеспечения. Следует уточнить, что это такое: rapid application development — стремительная разработка приложений, концепция создания средств разработки программных продуктов, дающая особенное внимание скорости и удобству программирования, образованию технологического процесса, позволяющего программисту предельно скоро формировать компьютерные программы. Практическое определение: RAD — это жизненный цикл процесса проектирования, разработанный для достижения наиболее значительной быстроты разработки и качества ПО, чем это вероятно при обычном подходе к проектированию. Кроме того, среда подходит для таких операционных систем, как Windows, Mac OS X, IOS и Android. Благодаря редкому комплексу незамысловатости языка и генерации машинного кода, разрешает непринужденно, и, при желании, довольно низкоуровнево взаимодействовать с операционной системой, а также с библиотеками, написанными на C/C++. Созданные программы не зависимы от стороннего ПО, как-то Microsoft.NET Framework, или Java Virtual Machine. Выделение и высвобождение памяти проверяется обычно пользовательским кодом, что, с одной стороны, ужесточает запросы к качеству кода, а с другой — делает реальным образование не простых приложений, со значительными запросами к отзывчивости (работа в реальном времени) [1, с.79]. В кросс-компиляторах для мобильных платформ заранее предусмотрен автоматическое вычисление ссылок на объекты, упрощающий задачу управления их временем жизни.

Для разработки базы данных было выбрано CASE-средство AllFusion ERwin Data Modeler. (ERwin DM) – первенствующее решение в сфере



проектирования и документирования данных, позволяющее образовывать, документировать и сопутствовать базы данных, хранилища и витрины данных.

ERwin DM предопределен для организаций, разрабатывающих и применяющих базы данных, для администраторов баз данных, системных аналитиков, проектировщиков баз данных, разработчиков, руководителей проектов. ERwin DM позволяет распоряжаться данными и сведениями в процессе корпоративных изменений, а также в обстановке нехватки квалифицированных кадров и быстроменяющихся технологий.

ERwin DM разрешает наглядно отражать сложные структуры данных. Подходящая в употреблении графическая среда AllFusionERwin DM упрощает разработку базы данных и автоматизирует массу трудоемких задач, снижая сроки формирования высококачественных и высокопроизводительных транзакционных баз данных и хранилищ данных.

Такое решение совершенствует коммуникацию на предприятиях, снабжая коллективную работу администраторов и разработчиков баз данных, неоднократное применение модели, а также наглядное представление совокупных активов данных в подходящем для понимания и обслуживания формате [10, с.113].

В заключении второй главы можно сделать следующие выводы.

В техникуме внедрена система автоматизации учебного процесса «1С: Колледж ПРОФ», которая содержит базу данных сведений, необходимых для автоматизации информационной деятельности данного предприятия, но имеющиеся возможности по анализу успеваемости студентов и посещаемости ими учебных занятий не соответствуют требованиям техникума по анализу этих показателей. В связи с этим, возникла необходимость создания отдельной информационной системы для решения данной задачи, по оценке качества обучения студентов техникума.

Результативное управление базой данных учащихся невыполнимо без системы автоматизации. Информационная система существует для ведения личных дел, учащихся техникума и работает отдельно или в составе ИС. Для

отображения процесса автоматизации были построены диаграммы в CASE средстве проектирования и разработки информационных систем и программного обеспечения для управления организациями – Rational Rose, а именно: диаграмма прецедентов, диаграмма классов, диаграмма последовательности действий и диаграмма взаимодействия. Чтобы образовать диаграммы вариантов использования, следует выбрать действующие лица, которые будут иметь отношение к работе подсистемы.

В результате обследования аппаратно-программной базы заказчика и предполагаемых условий эксплуатации данного проекта, было принято решение использовать для создания программы интегрированную среду разработки Borland Turbo Delphi. Такое решение совершенствует коммуникацию на предприятиях, снабжая коллективную работу администраторов и разработчиков баз данных, неоднократное применение модели, а также наглядное представление совокупных активов данных в подходящем для понимания и обслуживания формате.

## **3 РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

### **3.1 Описание технического задания**

Основное назначение АИС — хранение подробной информации о результатах оценки качества обучения студентов техникума и снабжение простого и скорого доступа к ней сотрудников учебной части техникума.

Такая система позволяет вести учёт результатов оценки качества обучения студентов техникума и обеспечивает лёгкий и быстрый доступ к ним сотрудников учебной части техникума.

Система предназначена для работы в отделах техникума, где существуют рабочие места, оборудованные средствами электронной вычислительной техники.

Разрабатываемая система реализовывает такие функции, как:

1. Поддержка ввода информации о показателях оценки качества обучения студентов:

- ввод записей об отделениях техникума;
- ввод записей о семестрах;
- ввод записей об изучаемых предметах;
- ввод записей о группах студентов техникума;
- ввод записей о студентах техникума;
- ввод записей о семестровых оценках студентов техникума по изучаемым ими предметам;
- ввод записей о количестве пропущенных учебных занятий студентами техникума.

2. Поддержка скорого и легкого доступа к данным:

- получение информации о хранящихся в системе данных;
- получение информации о результатах оценки качества обучения студентов техникума.

Входные данные вводятся в специальные поля ввода, заключающиеся в интерфейсе пользователя (программа). Рядом с любым полем ввода показана

подпись, показывающая, какой тип данных следует вводить в это поле и какой смысл несут такие данные. Специальных запросов к вводу не предъявляется.

Выходные данные приходят от базы данных в виде структур данных и воспроизводятся в программе в виде таблиц.

Разработка обычно проводится в три стадии:

- 1) разработка технического задания;
- 2) рабочее проектирование;
- 3) внедрение.

На этапе разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены следующие пункты работ:

- 1) разработка подпрограмм системы (осуществляется параллельно для всех подпрограмм);
- 2) разработка программной документации;
- 3) испытания системы.

На стадии внедрения должен быть выполнен этап разработки — подготовка и передача программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77 и требованием п. «Предварительный состав программной документации» настоящего технического задания.

На этапах проверок программы должны быть выполнены следующие виды работ:

- 1) разработка, согласование и утверждение программы и методики проверок;
- 2) проведение приемо-сдаточных испытаний;
- 3) корректировка программы и программной документации по итогам проверок.

На этапе подготовки и передачи программы должна быть выполнена

работа по подготовке и передаче программы и программной документации в эксплуатацию. Полное Техническое задание представлено в Приложении 1.

### 3.2 Реализация базы данных

Для того, чтобы выполнить данную задачу, следует разработать проект базы данных ИС. Таблицы — это основные объекты реляционной базы данных.

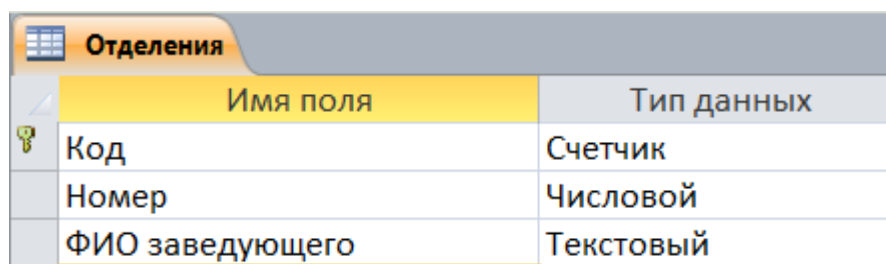
Исходя из принципа моделирования «Сущность — Связь» (от английского словосочетания ER — «Entity — Relationship»), согласно которой каждая сущность или объект модели данных должны представлять собой отдельную таблицу, следует сначала разделить исследовать предметную область деятельности структурного подразделения учебная часть на отдельные сущности.

В результате логического анализа бизнес-процессов, описанных во втором разделе данной работы, были выделены следующие сущности: отделения, группы, студенты, пропуски, оценки, семестры, предметы.

Уточняя атрибуты сущностей, приходим к выводам, что таблицы «Отделения», «Группы», «Студенты», «Пропуски», «Оценки», «Семестры» и «Предметы» должны иметь следующую логическую структуру:

Таблица «Отделения» должна содержать в себе такие атрибуты сущности, как (рисунок18):

- уникальный код для связи с другими таблицами и для однозначной идентификации записей (ключ);
- ФИО заведующего;
- номер отделения.



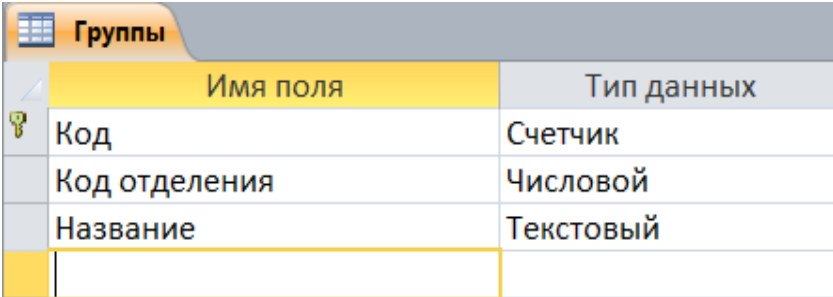
| Отделения       |            |
|-----------------|------------|
| Имя поля        | Тип данных |
| Код             | Счетчик    |
| Номер           | Числовой   |
| ФИО заведующего | Текстовый  |

Рис. 18 Структура таблицы «Отделения»<sup>18</sup>

<sup>18</sup>Рисунок составлен автором на основе собственного исследования

Таблица «Группы» должна содержать в себе такие атрибуты сущности, как (рисунок19):

- уникальный код для связи с другими таблицами и для однозначной идентификации записей (ключ);
- внешний ключ для определения принадлежности группы тому или иному отделению;
- название группы.

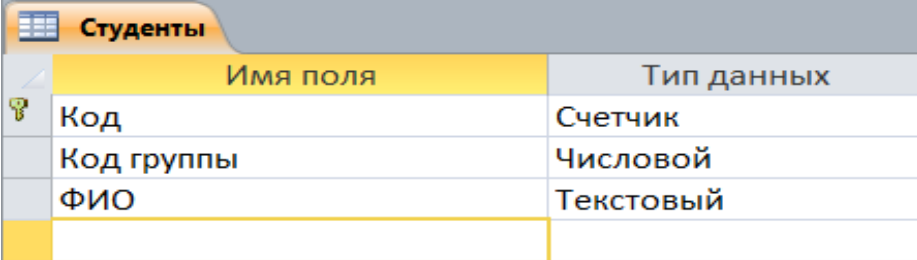


| Группы |               |            |
|--------|---------------|------------|
|        | Имя поля      | Тип данных |
| 🔑      | Код           | Счетчик    |
|        | Код отделения | Числовой   |
|        | Название      | Текстовый  |
|        |               |            |

**Рис. 19 Структура таблицы «Группы»<sup>19</sup>**

Таблица «Студенты» должна содержать в себе такие атрибуты сущности, как (рисунок20):

- уникальный код для связи с другими таблицами и для однозначной идентификации записей (ключ);
- внешний ключ для определения принадлежности студента той или иной группе;
- ФИО студента.



| Студенты |            |            |
|----------|------------|------------|
|          | Имя поля   | Тип данных |
| 🔑        | Код        | Счетчик    |
|          | Код группы | Числовой   |
|          | ФИО        | Текстовый  |
|          |            |            |

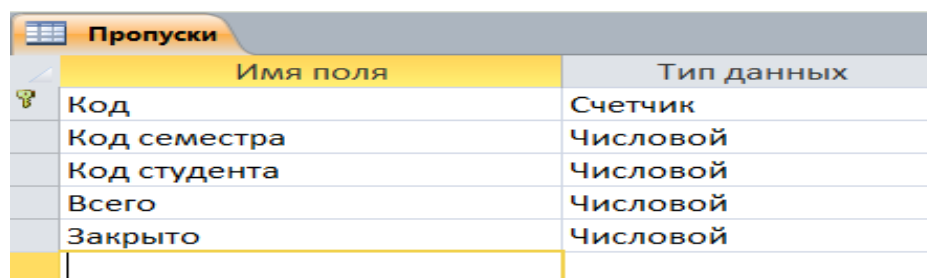
**Рис. 20 Структура таблицы «Студенты»<sup>20</sup>**

<sup>19</sup>Рисунок составлен автором на основе собственного исследования

<sup>20</sup>То же

Таблица «Пропуски» должна содержать в себе такие атрибуты сущности, как (рисунок21):

- уникальный код для связи с другими таблицами и для однозначной идентификации записей (ключ);
- внешний ключ для определения принадлежности количества пропусков тому или иному студенту;
- внешний ключ для определения принадлежности количества пропусков тому или иному семестру;
- общее количество пропусков;
- закрытое количество пропусков.



| Имя поля     | Тип данных |
|--------------|------------|
| Код          | Счетчик    |
| Код семестра | Числовой   |
| Код студента | Числовой   |
| Всего        | Числовой   |
| Закрыто      | Числовой   |

Рис. 21 Структура таблицы «Пропуски»<sup>21</sup>

Таблица «Оценки» должна содержать в себе такие атрибуты сущности, как (рисунок22):

- уникальный код для связи с другими таблицами и для однозначной идентификации записей (ключ);
- внешний ключ для определения принадлежности оценки тому или иному студенту;
- внешний ключ для определения принадлежности оценки тому или иному предмету;
- внешний ключ для определения принадлежности оценки тому или иному семестру;
- оценка.

<sup>21</sup>Рисунок составлен автором на основе собственного исследования

| Оценки       |            |
|--------------|------------|
| Имя поля     | Тип данных |
| Код          | Счетчик    |
| Код семестра | Числовой   |
| Код предмета | Числовой   |
| Код студента | Числовой   |
| Оценка       | Числовой   |

**Рис. 22 Структура таблицы «Оценки»<sup>22</sup>**

Таблица «Семестры» должна содержать в себе такие атрибуты сущности, как (рисунок23):

- уникальный код для связи с другими таблицами и для однозначной идентификации записей (ключ);
- номер семестра.

| Семестры |            |
|----------|------------|
| Имя поля | Тип данных |
| Код      | Счетчик    |
| Номер    | Числовой   |

**Рис. 23 Структура таблицы «Семестры»<sup>23</sup>**

Таблица «Предметы» должна содержать в себе такие атрибуты сущности, как (рисунок24):

- уникальный код для связи с другими таблицами и для однозначной идентификации записей (ключ);
- название предмета.

| Предметы |            |
|----------|------------|
| Имя поля | Тип данных |
| Код      | Счетчик    |
| Название | Текстовый  |

**Рис. 24 Структура таблицы «Предметы»<sup>24</sup>**

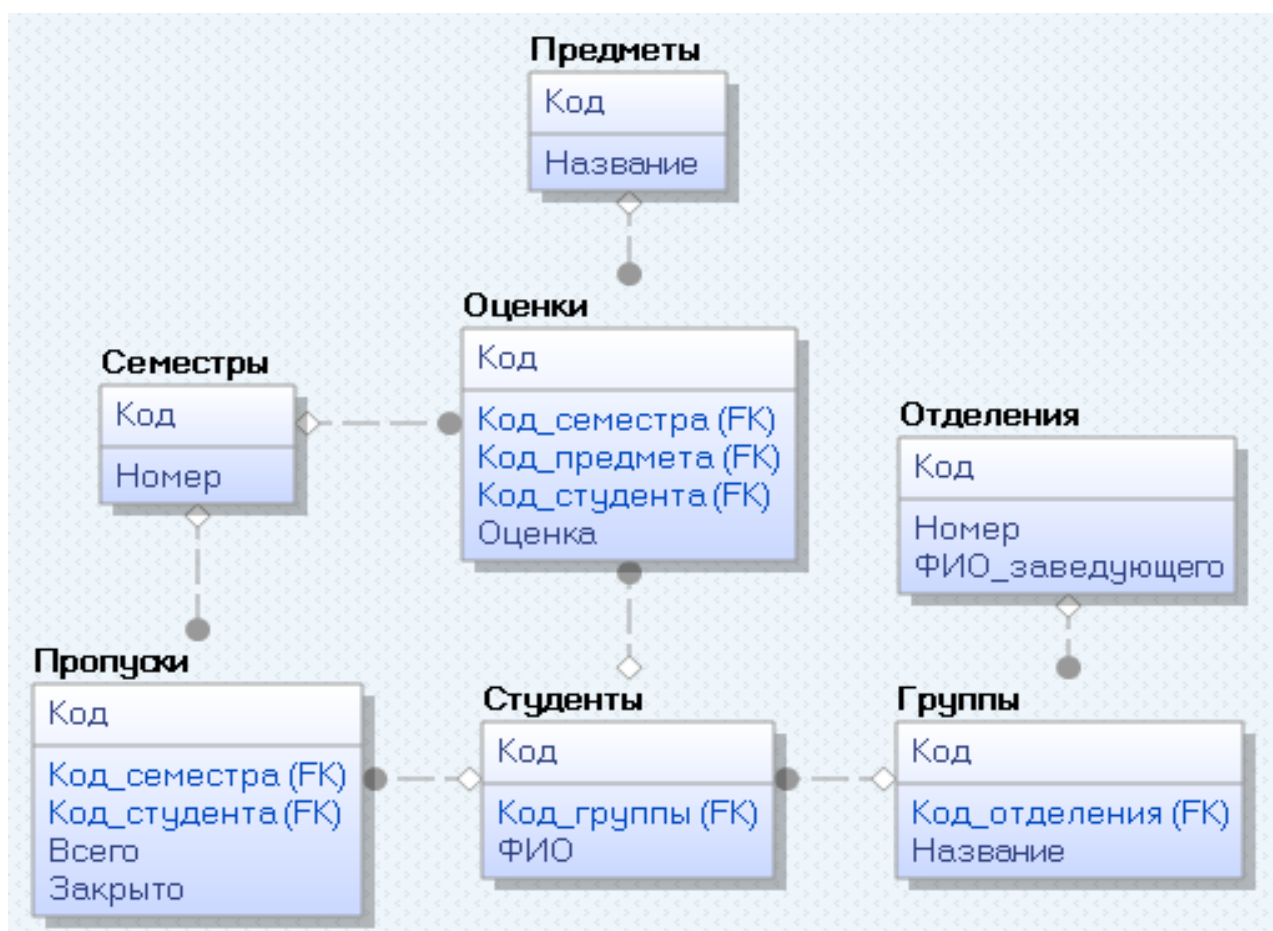
<sup>22</sup>Рисунок составлен автором на основе собственного исследования

<sup>23</sup>То же

<sup>24</sup>То же



Диаграмма «Сущность — связь», выполненная в программе CA Erwin Data Modeler показана на рисунке25.



**Рис. 25** Диаграмма «Сущность — связь»<sup>25</sup>

На основе разработанной логической модели составим физическую модель БД. Из рисунка25 видно, что:

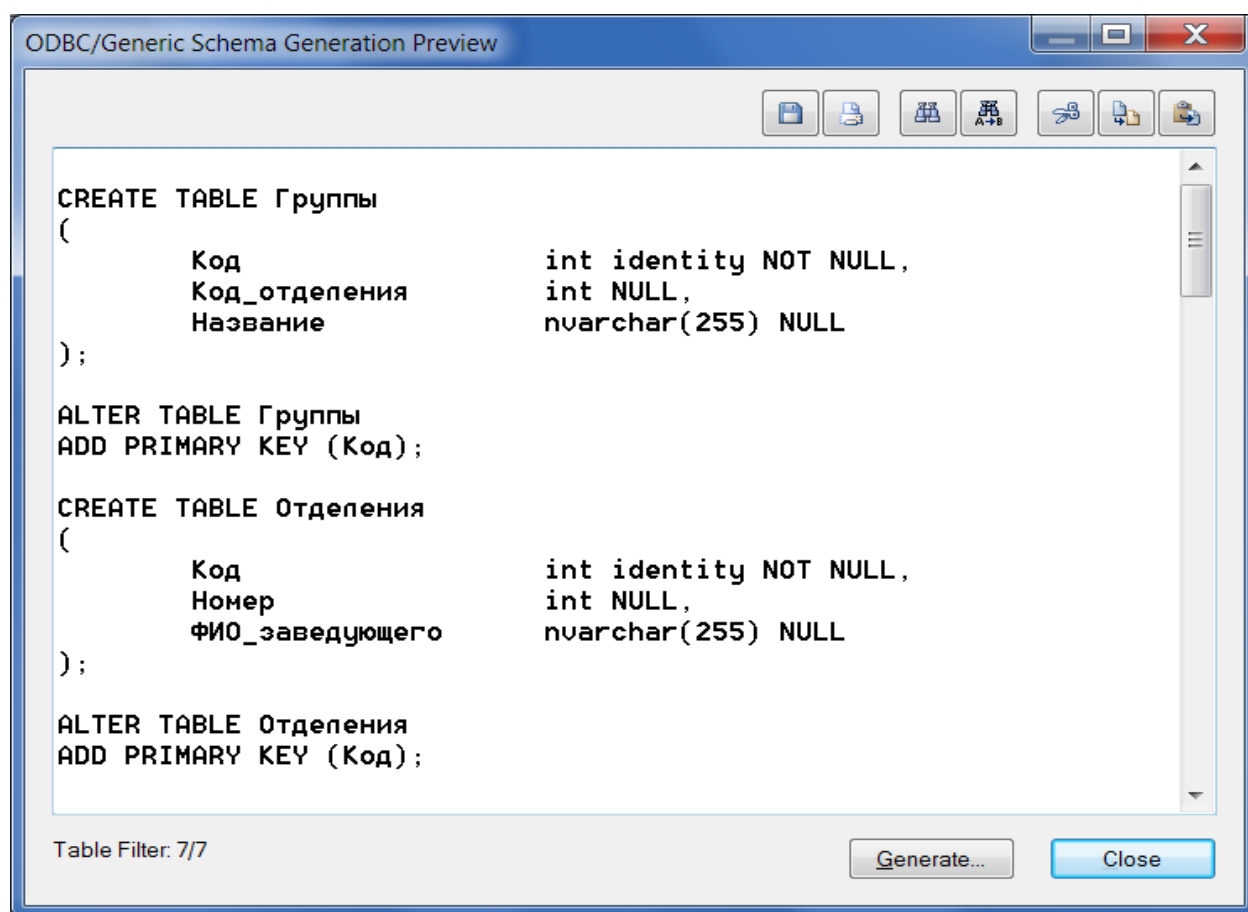
- таблица Оценки имеет внешние ключи [Код семестра] (связь типа «многие-к-одному» с полем Код таблицы Семестры), [Код предмета] (связь типа «многие-к-одному» с полем Код таблицы Предметы) и [Код студента] (связь типа «многие-к-одному» с полем Код таблицы Студенты);
- таблица Группы имеет внешний ключ [Код отделения] (связь типа «многие-к-одному» с полем Код таблицы Отделения);
- таблица Студенты имеет внешний ключ [Код группы] (связь типа «многие-к-одному» с полем Код таблицы Группы);

<sup>25</sup>Рисунок составлен автором на основе собственного исследования

«многие-к-одному» с полем Код таблицы Группы);

– а таблица Пропуски имеет внешние ключи [Код семестра] (связь типа «многие-к-одному» с полем Код таблицы Семестры) и [Код студента] (связь типа «многие-к-одному» с полем Код таблицы Студенты).

Имея физическую модель информационной системе в программе CA Erwin Data Modeler, можно сгенерировать код базы данных на языке SQL (рисунок26, Приложение 2).



**Рис. 26 Схема базы данных на языке SQL<sup>26</sup>**

Как говорилось ранее, формируемая база данных должна давать информацию об успеваемости и посещаемости учащихся техникума в разрезе семестров и учебных групп.

Для этого необходимо написать запросы на выборку данных, а именно:

<sup>26</sup>Рисунок составлен автором на основе собственного исследования

1. Запрос «Итоги семестра», который выводит информацию о названии групп, количестве студентов в них; количестве отличников, хорошистов и неуспевающих в каждой группе; об успеваемости, качестве обучения и количестве пропусков (Приложение 3).

2. Запрос «Отличники», который выводит информацию об отличниках:

```
Select nazv, fio From (  
SELECT Группы.Название as nazv, Группы.Код as kod,  
Min(Оценки.Оценка) AS minos,  
Студенты.Код as stkod, Студенты.ФИО as fio  
FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON  
Группы.Код = Студенты.[Код группы]) INNER JOIN (Семестры INNER JOIN  
Оценки ON Семестры.Код = Оценки.[Код семестра]) ON Студенты.Код =  
Оценки.[Код студента]) ON Отделения.Код = Группы.[Код отделения]  
WHERE (((Семестры.Номер)=:0) AND ((Отделения.Номер)=:1))  
GROUP BY Группы.Название, Группы.Код,  
Студенты.Код,Студенты.ФИО  
ORDER BY Студенты.Код)  
where minos=5  
groupbynazv, fio
```

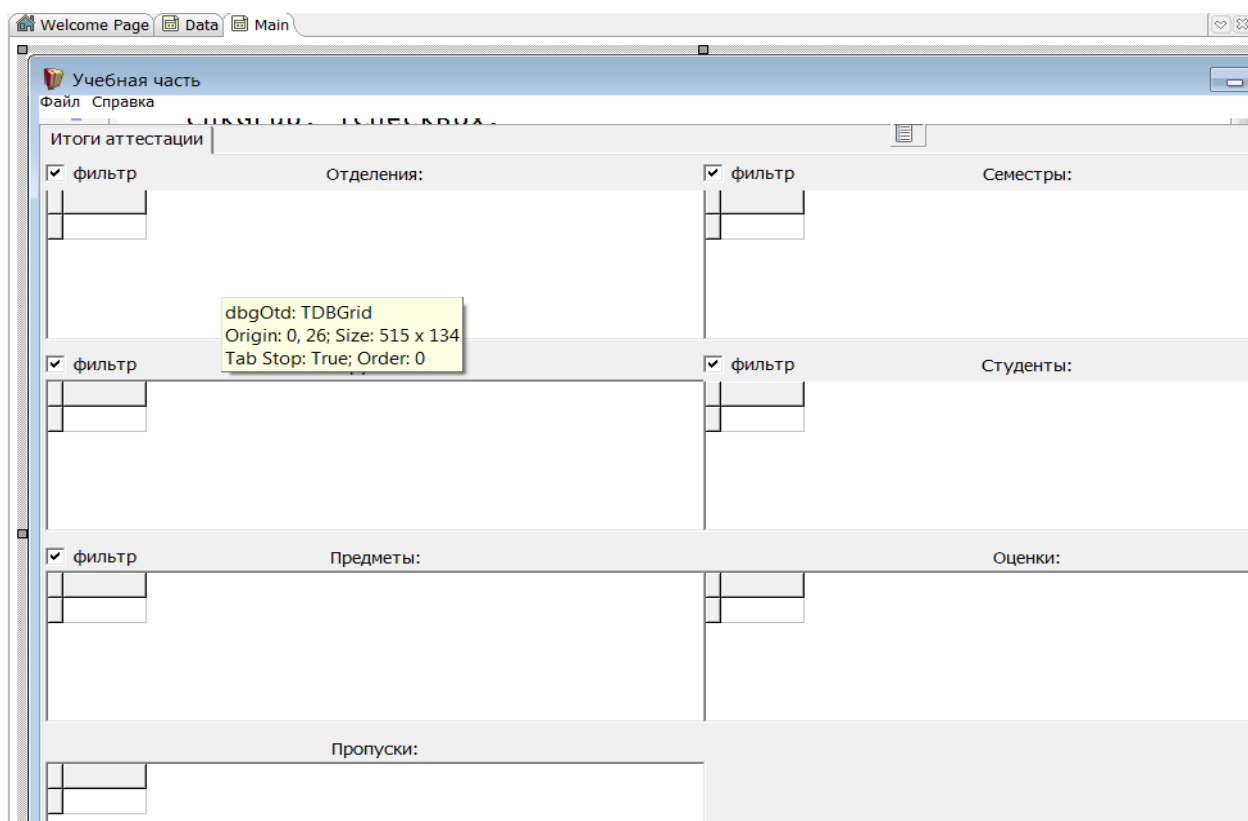
3. Запрос «Место по успеваемости», который выводит информацию об успеваемости (Приложение 4).

4. Запрос «Место по качеству», который выводит информацию о качестве обучения (Приложение 5).

5. Запрос «Место по пропускам», который выводит информацию о количестве пропущенных часов (Приложение 6).

### **3.3 Программная реализация информационной системы**

Программная реализация информационной системы осуществлялась на языке ObjectPascal в среде программирования BorlandTurboDelphi.



**Рис. 27 Размещение компонентов для доступа к таблицам базы данных<sup>27</sup>**

В программном модуле Data был добавлен код обработки событий OnCreate (рисунок 27). Данный код позволяет избежать проблемы нахождения базы данных при запуске приложения. Путь к базе данных определяется из пути к исполняемому файлу приложения.

```
procedure TdtmMain.DataModuleCreate(Sender: TObject);
Var
  i: Integer;
  s: String;
begin
  adcMain.Connected := False;
  s := ExtractFilePath(Application.ExeName);
  s := s + 'Базаданныхучебнойчасти';
  adcMain.ConnectionString := 'Provider=Microsoft.Jet.OLEDB.4.0;' +
```

<sup>27</sup>Рисунок составлен автором на основе собственного исследования

```

'Data Source=' + s + '\TFMT.MDB;Persist Security Info=False';
adcMain.Connected := True;
If FileExists(s + '\TFMT.MDB') Then
Begin
  For i := 1 To ComponentCount - 1 Do
  Begin
    If Components[i] Is TADOTable Then
    Begin
      With (Components[i] As TADOTable) Do
      Begin
        Active := True;
      End;
    End;
  End;
  If Components[i] Is TADOQuery Then
  Begin
    With (Components[i] As TADOQuery) Do
    Begin
      End;
    End;
  End;
End;
end;

```

Для компонентов TADOTable, которые подключены к таблицам, имеющим зависимые от них таблицы, были добавлены обработчики событий AfterScroll:

```

procedure TdtmMain.tblGrupAfterScroll(DataSet: TDataSet);
begin
  With DataSet Do
  Begin
    If (RecordCount < 1) Or (FieldByName('Код').AsString = '') Then

```

```

Begin
    Exit;
End;
If frmMain.chkGrup.Checked Then
Begin
    strFltStudByGrup := '[Кодгруппы] = ' +
        FieldByName('Код').AsString;
End
Else
Begin
    strFltStudByGrup := '';
End;
tblStud.Filter := strFltStudByGrup;
End;
end;

```

Это позволяет накладывать фильтр на отображение записей подчинённых таблиц. А для подчинённых таблиц были добавлены обработчики событий OnNewRecord:

```

ProcedureTdtmMain.tblOcenNewRecord(DataSet: TDataSet);
begin
WithDataSetDo
Begin
FieldByName('Кодпредмета').AsString :=
    tblPredm.FieldByName('Код').AsString;
FieldByName('Кодстудента').AsString :=
    tblStud.FieldByName('Код').AsString;
FieldByName('Кодсеместра').AsString :=
    tblSem.FieldByName('Код').AsString;
End;
end;

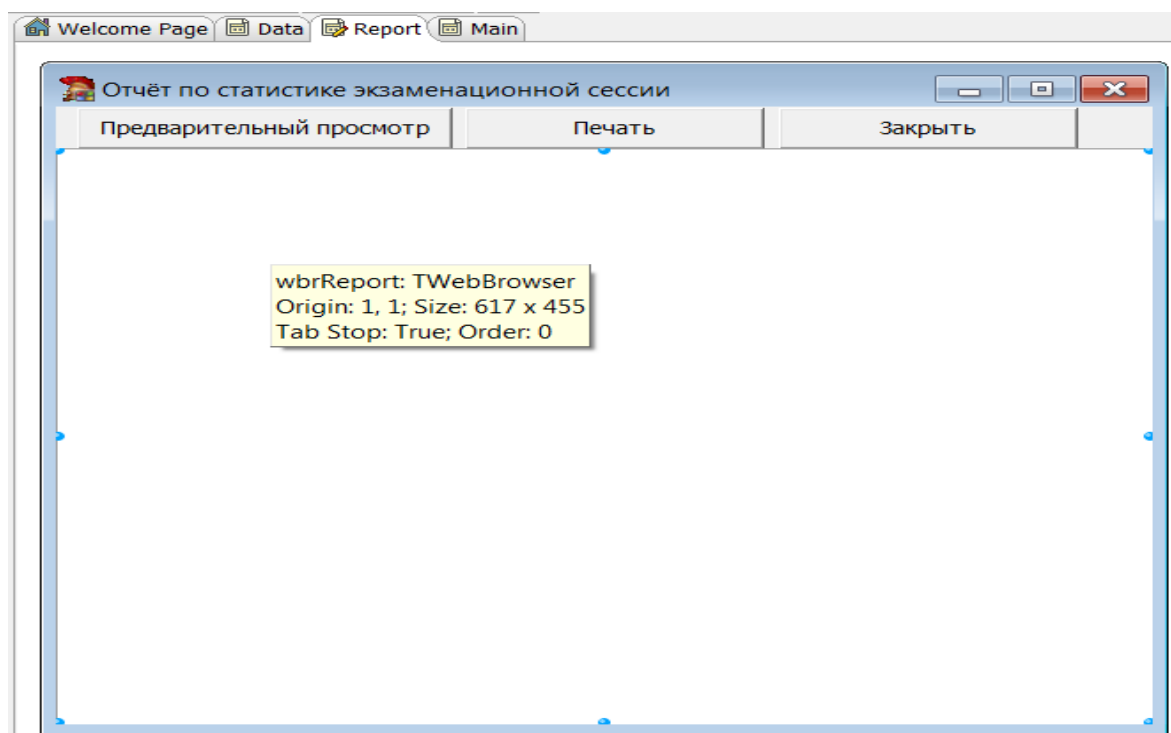
```

Это позволяет автоматизировать процесс заполнения ключевых полей.

Для заполнения вычисляемого поля была создана отдельная процедура обработки события OnCalcFields.

```
procedure TdtmMain.tblPropCalcFields(DataSet: TDataSet);  
begin  
    DataSet.FieldName('н/у').AsInteger :=  
    DataSet.FieldName('Всего').AsInteger -  
    DataSet.FieldName('Закрыто').AsInteger;  
end;
```

Для отображения и печати отчёта на форму Report был помещен компонент TWebBrowser (рисунок28).



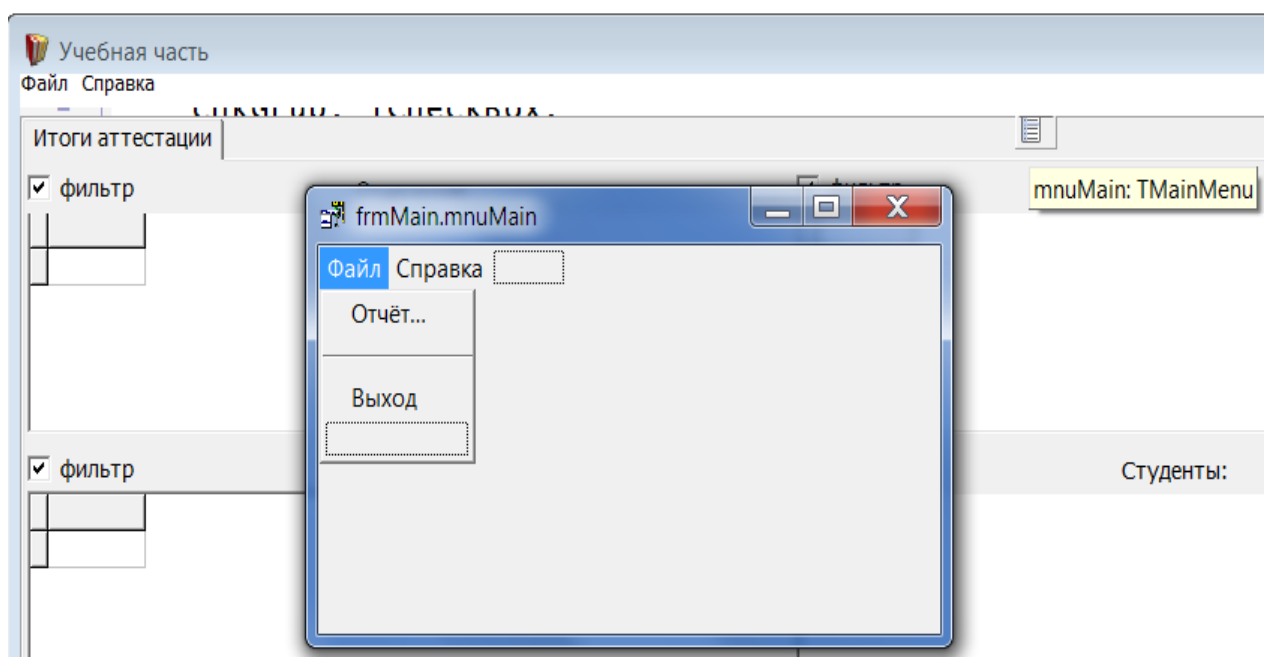
**Рис. 28 Дизайн формы отчёта<sup>28</sup>**

Для завершения формирования отчёта необходимы обработчики событий (Приложение 7).

---

<sup>28</sup>Рисунок составлен автором на основе собственного исследования

Для создания главного меню приложения на главную форму был добавлен компонент TMainMenu, в который были добавлены пункты при помощи редактора и заданы обработчики событий для вызова форм «О программе...», «Статистика экзаменационной сессии» и выхода из программы (рисунок29).



**Рис. 29 Создание главного меню программы<sup>29</sup>**

Процедура обработки события выбора пункта меню «Отчёт...» представлена в Приложении8).

### **3.4 Работа с программой**

Общий вид главного окна программного приложения показан на рисунке30.

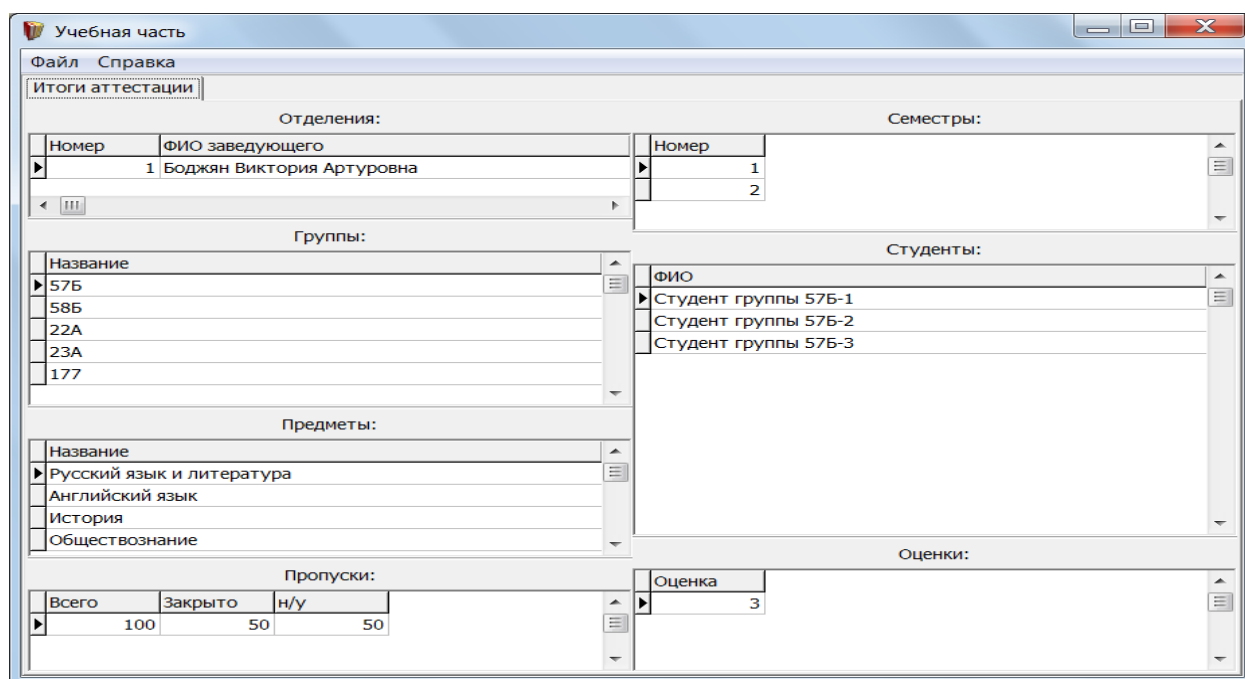
Информация на экране представлена в 7 таблицах: Отделения, Семестры, Группы, Студенты, Предметы, Пропуски и Оценки:

- в таблице Отделения перечислены отделения техникума;

<sup>29</sup>Рисунок составлен автором на основе собственного исследования



- в таблице Семестры перечислены номера учебных семестров;
- в таблице Группы перечислены группы студентов;
- в таблице Студенты перечислены студенты техникума;
- в таблице Предметы перечислены изучаемые в техникуме учебные предметы;
- в таблице Пропуски указаны общее количество пропусков, количество пропусков по уважительной причине и количество пропусков по неуважительной причине;
- в таблице Оценки указана оценка выбранного студента по выбранному предмету за указанный номер семестра.



**Рис. 30 Общий вид главного окна программного приложения<sup>30</sup>**

Добавление новой строки в таблицу производится нажатие клавиши «стрелка вниз» при нахождении курсора в последней строке соответствующей таблицы. Удаление строки производится при помощи нажатия сочетания клавиш Ctrl+Delete. Режим редактирования включается при помощи двойного щелчка левой клавишей мыши, либо при помощи нажатия клавиши F2.

<sup>30</sup>Рисунок составлен автором на основе собственного исследования

Порядок работы с программой следующий:

- вводятся сведения об отделениях техникума (номера и ФИО заведующих);
- вводится информация о номерах семестров;
- вводится информация о группах студентов (названия);
- вводится информация о студентах (ФИО);
- вводится информация о предметах (названия);
- вводится информация об оценке выбранного студента по выбранному предмету за указанный семестр;
- вводится информация о пропусках (общее количество пропущенных часов и количество пропущенных часов по уважительным причинам, количество пропущенных часов по неуважительным причинам рассчитывается автоматически).

После заполнения таблиц возможен вывод отчёта по статистике учебного процесса за указанный семестр путём выбора пункта меню Отчёт из меню Файл. Сгенерированный отчёт показан на рисунок<sup>31</sup>.



Отчёт по статистике экзаменационной сессии

Предварительный просмотр

Печать

Закрыть

Итоги аттестации по результатам 1 семестра 2015—2016 учебного года на отделении № 1

| № группы | К-во студентов | «5» | «4,5» | «2» | Успеваемость, % | Место | Кач-во обучения, % | Место | Пропуски «ув» | Пропуски «н/у» | Пропуски н/у на 1 человека | Место | Баллы | Итого |
|----------|----------------|-----|-------|-----|-----------------|-------|--------------------|-------|---------------|----------------|----------------------------|-------|-------|-------|
| 57Б      | 3              | 1   | 1     | 1   | 66,7            | 1     | 66,7               | 1     | 70            | 80             | 26,7                       | 7     | 9     | 2     |
| 58Б      | 3              | -   | -     | -   | -               | 7     | -                  | 7     | 234           | 242            | 80,7                       | 1     | 15    | 6     |
| 22А      | 3              | -   | -     | 2   | 33,3            | 2     | -                  | 2     | 68            | 240            | 80                         | 6     | 10    | 3     |
| 23А      | 3              | -   | -     | 3   | 0               | 6     | -                  | 6     | 244           | 218            | 72,7                       | 5     | 17    | 7     |
| 177      | 3              | -   | -     | 3   | 0               | 5     | -                  | 5     | 190           | 178            | 59,3                       | 4     | 14    | 5     |
| 039      | 3              | -   | -     | 3   | 0               | 4     | -                  | 4     | 230           | 416            | 138,7                      | 3     | 11    | 4     |
| 040      | 3              | -   | -     | 3   | 0               | 3     | -                  | 3     | 164           | 114            | 38                         | 2     | 8     | 1     |

Отличников 1 человек:

Студент группы 57Б-3 — гр. 57Б

1 место — группа 040

2 место — группа 57Б

3 место — группа 22А

Заведующая отделением

Боджан Виктория Артуровна

Рис. 31 Сгенерированный отчёт<sup>31</sup>

<sup>31</sup>Рисунок составлен автором на основе собственного исследования

### 3.5 Обоснование экономической эффективности

Для обоснования экономической эффективности разработанной информационной системы необходимо провести расчет:

- 1) стоимости часа работы ПК.
- 2) затрат на составление и отладку программы.
- 3) показателей эффективности проекта.
- 4) себестоимости программной услуги

Чтобы рассчитать издержки на разработку программы, следует составить технологический процесс разработки этой программы. Данные представлены в таблице 1.

**Таблица 1– Этапы разработки программы<sup>32</sup>**

| Этапы разработки программы     | Количество дней |
|--------------------------------|-----------------|
| 1.Подготовительный этап        | 10              |
| 2.Разработка программы         | 20              |
| 3.Отладка программы            | 5               |
| 4.Эксплуатация и сопровождение | 1               |
| Итого                          | 36              |

Проанализировав данную таблицу можно сказать о том, что этап подготовки составил 10 дней, разработка программы заняла 20 дней, за которые смогли разработать главные функции, была разработана логическая модель и составлена логика программы.

Отладку программы проводили 5 дней, а сопровождение и эксплуатация составили 1 день.

Рассчитаем по формуле стоимость часа работы на персональном компьютере (1):

---

<sup>32</sup> Таблица составлена автором на основе собственного исследования

$$S_{\text{пк}} = Z_{\text{эк}} \cdot (T \cdot n) \quad (1)$$

где  $Z_{\text{эк}}$  – затраты(издержки) на содержание и обслуживание персонального компьютера (за месяц);

$T$ - полезный фонд времени работы персонального компьютера (за месяц);

$n$ -количество машин (в данном случае 3: программисту, руководителю, дополнительному персоналу).

$$S_{\text{пк}} = 20882,21 \cdot (151,2 \cdot 3) = 46,04 \text{руб}$$

Затраты на содержание и обслуживание персонального компьютера рассчитаем следующим образом:

$$Z_{\text{эк}} = Z_{\text{п}} + C_{\text{оцотч}} + (N_{\text{рас}} + A_{\text{м}} + Z_{\text{эл}} + Z_{\text{мат}} + Z_{\text{рем}}) \cdot n \quad (2)$$

где  $Z_{\text{п}}$  – заработная плата обслуживающего персонала;

$C_{\text{оцотч}}$  - отчисления в фонды социального страхования и обеспечения;

$N_{\text{рас}}$  - накладные расходы;

$A_{\text{м}}$  – амортизационные отчисления;

$Z_{\text{рем}}$  – затраты на ремонт оборудования;

$Z_{\text{эл}}$  – затраты на электроэнергию;

$Z_{\text{мат}}$  – затраты на материалы.

$$Z_{\text{эк}} = 12000 + 3000 + (2702,7 + 625 + 229,82 + 450 + 200) \cdot 3 = 27622,56 \text{руб}$$

На основе следующим данных определяется заработная плата программиста за период написания программы, эти данные отражаются в таблице 2 «Затраты на заработную плату».

**Таблица 2 – Затраты на заработную плату<sup>33</sup>**

| Показатели                             |          | Программист | Итог затрат (руб.) |
|----------------------------------------|----------|-------------|--------------------|
| Количество часов работы                |          | 126         | -                  |
| Оплата                                 | За 1 час | 150         | -                  |
|                                        | Всего    | 18900,00 р  | 18900,00 р         |
| Дополнительная заработная плата        |          | 1890,00 р   | 1890,00 р          |
| Страховые взносы во внебюджетные фонды |          | 6237,00 р   | 6237,00 р          |
| Итого                                  |          | 27027,00 р  | 27027,00 р         |

Дополнительная заработная плата составила 10% от основной. Страховые взносы во внебюджетные фонды составили 30% от общей заработной платы. Следовательно, в итоге установим общие затраты на заработную плату, которую можно рассчитать по следующей формуле (3):

$$З_{\pi} = З_{\pi} + З_{\text{доп}} + О_{\text{гчстр}} \quad (3)$$

$$З_{\pi} = 18900 + 1890 + 6237 = 27027 \text{ руб.}$$

Накладные расходы составили 10% от основной заработной платы. Эти расходы будут тратиться на организацию производства, управление и т.п. Накладные расходы устанавливаются по следующей формуле (4):

$$Н_{\text{рас}} = (З_{\pi} * Н) / 100 \quad (4)$$

где Н - процент накладных расходов (10%)

$$Н_{\text{рас}} = (27027 * 10) / 100 = 2702,7 \text{ руб.}$$

Расчет амортизационных отчислений ведется по следующей формуле (5):

---

<sup>33</sup> Таблица составлена автором на основе собственного исследования

$$A_m = (S_{\text{пер}} / (T_{\text{экс}} * 12)) \quad (5)$$

где  $S_{\text{пер}}$  – первоначальная стоимость оборудования;

$T_{\text{экс}} * 12$  – срок эксплуатации на 12 месяцев.

$$A_m = 15000 / (2 * 12) = 625$$

Затраты на электроэнергию можно рассчитать следующим образом (6):

$$З_{\text{эл}} = P * T * Z \quad (6)$$

где  $P$  - мощность оборудования;

$T$  - общее время работы оборудования;

$Z$  - цена одного киловатта/час (3,8 руб.)

$$З_{\text{эл}} = 0.48 * 126 * 3.8 = 229.82 \text{ руб.}$$

Месячный полезный фонд работы на персональном компьютере (на определенный период) определим по такой формуле, как (7):

$$П_{\text{фТ}} = T_{\text{реж}} - (T_{\text{теж}} * P_{\text{ем}}) / 100 \quad (7)$$

где  $T_{\text{реж}}$  - режимный фонд времени;

$P_{\text{ем}}$  - процент плановых потерь на профилактику и ремонт (10%)

$$П_{\text{фТ}} = 176 - (176 * 10) / 100 = 158$$

$$T_{\text{реж}} = N_{\text{раб}} * t_{\text{раб}} \quad (8)$$

где  $N_{\text{раб}}$  – количество рабочих дней в месяце;

$t_{\text{раб}}$  – продолжительность рабочего дня (в часах).

$$T_{\text{реж}} = 22 * 8 = 176$$

Рассчитаем затраты на составление и отладку программы.

Стоимость решения задачи на персональном компьютере определим следующей формулой (9):

$$S_{\text{реш}} = Z_{\text{п.кон}} + Z_{\text{на отладку}} * S_{\text{пк}} + S_{\text{мат}} + H_{\text{рас}} \quad (9)$$

где  $Z_{\text{п.кон}}$  – заработная плата консультанта;

$Z_{\text{на отладку}}$  – затраты времени на отладку программы и решения задачи на ПК;

$S_{\text{эвм}}$  – стоимость часа работы на персональном компьютере, которую определим по этой формуле (1);

$S_{\text{мат}}$  – стоимость материалов, используемых при составлении и отладке программы на ПК;

$H_{\text{рас}}$  – накладные расходы можно рассчитать данным образом (10):

$$S_{\text{реш}} = 0 + 30 * 46,04 + 450 + 2702,7 = 4533,9 \text{ руб.}$$

Расчет затрат времени на отладку программы и решения задачи на персональном компьютере определим следующим образом (10):

$$Z_{\text{на отладку}} = S * d \quad (10)$$

где  $S$  – средняя занятость в течении дня;

$d$  – количество дней для отладки программы на персональном компьютере

$$Z_{\text{на отладку}} = 6 * 5 = 30 \text{ часов}$$

Рассчитаем себестоимость программной услуги. Такой программный продукт можно реализовать в единичном экземпляре или тиражирован и продан клиентам. Как правило, расчет экономической эффективности использования разработки производят для ее клиентов.

Существенным фактором, воздействующим на процесс создания цены, выражается конкуренция на рынке, необходимость учета которой абсолютно ясна. Чтобы повысить конкурентоспособность продукта на рынке, возможно придется снизить его цену. Также повысить конкурентоспособность можно путем улучшения качества товара или услуги и его выгодные для потребителя отличительные признаки по сравнению со схожими продуктами конкурентов.

Самым ответственным моментом для разработчика, если смотреть с позиции экономики, проявляется процесс образования цены. Бесспорно, что программные продукты и услуги являются довольно специфическим товаром с массой характерных им особенностей. Большинство их особенностей выражаются и в методах расчетов цены на них. На разработку программного продукта или услуги средней сложности часто требуются крайне небольшие средства. Тем не менее, при этом он может предоставить экономический эффект, существенно превосходящий эффект от применения довольно дорогих по стоимости систем.

Нужно отметить, что у программных продуктов почти не имеется процесса физического старения и износа. Для них существенные затраты перепадает на разработку образца, тогда как процесс тиражирования – это относительно не сложная и не дорогая процедура копирования магнитных носителей и сопровождающей документации. Следственно, такой товар не располагает, по сути, рыночной стоимостью, вырабатываемой на базе общественно надобных затрат труда.

Цена на программные продукты определяется на единицу программной продукции с учетом совокупности ее поставки.

Её цена, обычно основывается на базе нормативной себестоимости производства и прибыли следующим образом (11):



$$Ц_{п}=C+N_3 \quad (11)$$

где С - себестоимость единицы продукции, руб.;

$N_3$ - добавка к цене (10-15% от себестоимости).

Следует рассчитать себестоимость программы с помощью суммирования всех затрат, данные изобразим в таблице 3 «Сводная таблица затрат».

**Таблица 3 – Сводная таблица затрат<sup>34</sup>**

|                                                            |             |
|------------------------------------------------------------|-------------|
| Стоимость решения задачи на ПК:                            | 29433,90р.  |
| $Z_{п.конс}$ — заработная плата консультанта               | 0           |
| $S_{ПК}$ — стоимость часа работы ПК                        | 46,04р.     |
| $S_{мат}$ — стоимость материалов                           | 450р.       |
| $N_{рас}$ — накладные расходы                              | 2702,7р.    |
| $Z_{на отладку}$ - затраты времени на решения задачи на ПК | 30          |
| $S$ — средняя занятость в течение дня                      | 8           |
| $d$ — количество дней на разработку программы на ПК        | 20          |
| $Z_{плата}$ - зарплата программиста                        | 27 027,00р. |
| Основная заработная плата                                  | 18 900,00р. |
| Дополнительная заработная плата (10%)                      | 1 890,00р.  |
| Отчисление в страховые фонды (30%)                         | 6 237,00р.  |
| $Z_{диз}$ - затраты на руководителя                        | 6 000,00р.  |

В результате для создания программы понадобилось затратить 29 433,90р. на разработку. Рассчитаем затраты на разработку программного обеспечения  $C_{РП}$ , которые формируются из затрат на его проектирование  $C_{ПК}$  и затрат на программирование СПР (12):

$$C_{РП} = C_{ПК} + C_{ПР} \quad (12)$$

Расчет затрат на проектирование программного обеспечения. Затраты на разработку алгоритмического и информационного обеспечения формируются следующим образом (13):

<sup>34</sup> Таблица составлена автором на основе собственного исследования

$$C_{\text{ПК}} = Z_{\text{ПР}} + M_{\text{Э}} + (\Pi_{\text{НР}} \cdot Z_{\text{ПР}}) / (100\%) \quad (13)$$

где  $Z_{\text{ПР}}$  - затраты на оплату труда проектировщика, руб.;

$M_{\text{Э}}$  - затраты на эксплуатационные нужды (носители информации, бумага, заправка картриджа принтера, работа в глобальной сети Интернет по поиску литературы и др.), руб.;

$\Pi_{\text{НР}}$  - процент накладных расходов, %.

Затраты на эксплуатационные нужды можно рассчитать по данной формуле (14):

$$M_{\text{Э}} = C_{\text{Д}} \cdot N_{\text{Д}} + C_{\text{Б}} \cdot N_{\text{Б}} + C_{\text{К}} + C_{\text{ИН}} \cdot N_{(\text{ч.ИН})} \quad (14)$$

где  $C_{\text{Д}}$  - стоимость одного диска CD-ROM, руб.;

$N_{\text{Д}}$  - количество использованных дисков, шт.;

$C_{\text{Б}}$  - стоимость 1 пачки бумаги, руб.;

$N_{\text{Б}}$  - количество использованной бумаги, пачка;

$C_{\text{К}}$  - стоимость заправки картриджа принтера, руб.;

$C_{\text{ИН}}$  - стоимость работы одного часа в глобальной сети Интернет, руб.;

$N_{\text{ч.ИН}}$  - количество часов работы в глобальной сети Интернет по поиску литературы, руб.

Количество часов, затраченное на работу в глобальной сети Интернет по поиску литературы, вычисляется по формуле (15):

$$N_{(\text{ч.ИН})} = N_{\text{Ч}} \cdot N_{\text{Д}} \cdot N_{\text{Н}} \quad (15)$$

где  $N_{\text{Ч}}$  - количество часов работы в глобальной сети Интернет в день, час;

$N_{\text{Д}}$  - количество рабочих дней в неделе, дни;

$N_{\text{Н}}$  - количество недель, недели.

Разрабатывая программное обеспечение, поиск литературы в глобальной сети Интернет проводился каждый день по 2 часа во время прохождения

преддипломной практики, при количестве дней в рабочей неделе – 5, количестве недель, нужных для поиска литературы– 1. Значит, количество часов работы в сети Интернет, согласно формуле (15), составляет:

$$N_{\text{ч.ин}}=3\cdot5\cdot1=15 \text{ часов}$$

Затраты на эксплуатационные нужды, при количестве применяемых дисков – 1 по цене – 15 руб./шт., количестве применяемых пачек бумаги – 2 по цене – 150 руб./пачка, стоимости заправки картриджа принтера – 200 руб., стоимость работы одного часа в глобальной сети Интернет – 15 руб., составят, согласно формуле (14),

$$M_{\text{э}}=15\cdot1+150\cdot2+200+15\cdot15=740 \text{ рублей}$$

Затраты на разработку алгоритмического и информационного обеспечения, интерфейса пользователя, при оплате труда проектировщика – 1500 руб./неделя, количестве рабочих недель – 6, проценте накладных расходов – 10%, составят (по данной формуле (13)):

$$C_{\text{ПК}}=1500\cdot6+740+(10\cdot1500\cdot6)/100= 10640 \text{ рублей}$$

Расчет затрат на программирование программного обеспечения. Затраты на программирование устанавливаются следующим образом (16):

$$C_{\text{ПР}} = Z_{\text{ПР}} + M_{\text{э}} + Z_{\text{отл}} \frac{P_{\text{НР}} \cdot Z_{\text{ПР}}}{100\%} \quad (16)$$

где  $Z_{\text{ПР}}$  - затраты на оплату труда программиста, руб.;

$M_{\text{э}}$  - затраты на эксплуатационные нужды (носители информации), руб.;

$Z_{\text{отл}}$  - затраты на отладку программы, руб.;

$P_{НР}$  - процент накладных расходов, %.

Затраты на эксплуатационные нужды формируются из покупки 1 диска по цене 10 руб./шт.

Затраты на отладку устанавливаются следующим образом (17):

$$Z_{отл} = t_{отл} \cdot C_{мн} \quad (17)$$

где  $t_{отл}$  - время, требуемое для отладки программы, час;

$C_{мн}$  - стоимость работ по отладке, руб./день.

Стоимость работ по отладке определяется по формуле (18):

$$C_{мн} = Z_{мн} \cdot n \cdot (1 + Q_{осн}) \quad (18)$$

где  $Z_{осн}$  - основная заработная плата разработчика, руб./день;

$N$  - число разработчиков программного продукта, чел.;

$Q_{осн}$  - коэффициент, принимающий во внимание отчисления на социальные нужды по действующему законодательству (0.26);

$r$  - среднее число рабочих дней, израсходованных на отладку модуля, день.

Время, необходимое для отладки, можно рассчитать следующим образом (19):

$$t_{отл} = \frac{qe(1+p)}{4k} \quad (19)$$

где  $q$  - полагаемое время отладки, день;

$e$  - коэффициент сложности программы;

$p$  - коэффициент коррекции программы;

$k$  - коэффициент квалификации разработчика.

Применяя (17), (18) и (19), затраты на отладку программы, при

возможном времени отладки – 10 дней, коэффициенте сложности программы – 1.5, коэффициенте коррекции программы – 0.2, коэффициенте квалификации разработчика – 0.8, стоимости 1 рабочего дня разработчика – 250 руб., числе разработчиков – 1 составит:

$$З_{отл}=(10 \cdot 1.5 \cdot (1+0.2))/(4 \cdot 0.8) \cdot 250 \cdot 1 \cdot (1+0.26)= 1772 \text{ рубля}$$

Затраты на программирование программного обеспечения, при оплате труда программиста – 1500 руб./неделя, продолжительности построения – 6 недели и проценте накладных расходов – 10 %, составят по формуле (17):

$$С_{пр}=1500 \cdot 6+10+1772+(10 \cdot 1500 \cdot 6)/100= 11682 \text{ рубля}$$

Из (19) вытекает, что затраты на разработку программы составят:

$$С_{рп}=10640+11682= 22322 \text{ рубля}$$

Расчет затрат на внедрение программного обеспечения. Внедрение программного обеспечения не намеривает добавочной покупки компьютеров или иного оборудования. Для функционирования программного обеспечения не нужно специального программного обеспечения. Следовательно, затраты на внедрение программного обеспечения равны нулю.

## ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе изложены теоретические и практические аспекты создания автоматизированных систем обработки данных.

Разработаны программные модули на языке программирования Turbo Pascal в среде программирования Turbo Delphi, которые позволяют автоматизировать процесс занесения информации в созданную базу данных, а также автоматизировать процесс образования отчётов для печати.

Разработаны информационные системы учета успеваемости и посещаемости студентов (на примере ГБПОУ КК «Туапсинский гидрометеорологический техникум»).

Данная разработка выгодно отличается от них тем, что имеет низкую стоимость и возможность произвольной адаптации под конкретные нужды заказчика.

Данная разработка может применяться в различных сферах практической деятельности, связанных с предметной областью данной информационной системы и базы данных, а именно с деятельностью структурного подразделения учебная часть.

В результате выполнения бакалаврской работы можно сделать следующие **выводы**:

В техникуме внедрена система автоматизации учебного процесса 1С: Колледж ПРОФ, которая содержит базу данных сведений, необходимых для автоматизации информационной деятельности данного предприятия, но имеющиеся возможности по анализу успеваемости студентов и посещаемости ими учебных занятий не соответствуют требованиям техникума по анализу этих показателей. В связи с этим, возникла необходимость создания отдельной информационной системы для решения данной задачи, по оценке качества обучения студентов техникума.

Результативное управление базой данных учащихся невыполнимо без системы автоматизации. Информационная система существует для ведения

личных дел, учащихся техникума и работает отдельно или в составе ИС. Для отображения процесса автоматизации были построены диаграммы в CASE средстве проектирования и разработки информационных систем и программного обеспечения для управления организациями – Rational Rose, а именно: диаграмма прецедентов, диаграмма классов, диаграмма последовательности действий и диаграмма взаимодействия. Для создания диаграммы вариантов применения, следует выбрать действующие лица, которые будут иметь отношение к работе подсистемы.

В результате обследования аппаратно-программной базы заказчика и предполагаемых условий эксплуатации данного проекта, было принято решение использовать для создания программы интегрированную среду разработки Borland Turbo Delphi. Такое решение совершенствует коммуникацию на предприятиях, снабжая коллективную работу администраторов и разработчиков баз данных, неоднократное применение модели, а также наглядное представление совокупных активов данных в подходящем для понимания и обслуживания формате.

Умелое сочетание данных технологий позволяет создавать эффективные приложения для обработки данных.

Внедрение программного обеспечения не намеривает добавочной покупки компьютеров или иного оборудования. Для функционирования программного обеспечения не нужно специального программного обеспечения. Следовательно, затраты на внедрение программного обеспечения равны нулю.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Климова Л.М. Delphi 7. Основы программирования. Решение типовых задач. Самоучитель. — Москва, КУДИЦ-Образ, 2012. — 480 с.
2. Шупруга В.П. Delphi 2006 на примерах (+ CD-ROM). — Москва, БХВ-Петербург, 2012. — 528 с.
3. Культин Н.К. Delphi в задачах и примерах (+ CD-ROM). — Москва, БХВ-Петербург, 2012. — 288 с.
4. Осипов Д.В. Delphi. Профессиональное программирование. — Москва, Символ-Плюс, 2013. — 1056 с.
5. Любавин С.А. Turbo Delphi для новичков и не только. — Москва, ИТ Пресс, 2012. — 320 с.
6. Абрамян М.Э. Практикум по программированию на языке Паскаль. 4-е изд. — Ростов-на-Дону: изд-во «ЦВВР», 2013. — 187с.
7. Абрамян М.Э., Михалкович С.С. Основы программирования на языке Паскаль: Скалярные типы данных, управляющие операторы, процедуры и функции. — Ростов-на-Дону: изд-во «ЦВВР», 2012. — 198 с.
8. Аляев Ю.В., Козлов О.А. Алгоритмизация и языки программирования Pascal, C++, Visual Basic. — М.: Финансы и статистика, 2013. — 235 с.
9. Амелина Н.И., Королева Е.М. Методические указания для студентов механико-математического факультета «Задачи по программированию». Часть 2. — Ростов-на-Дону: УПЛ РГУ, 2014. — 31 с.
10. Голицына О.Л., Попов И.И. Основы алгоритмизации и программирования: учебное пособие. — М.: Форум: Инфра-М, 2014. — 148 с.
11. Грейди Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ /Пер. с англ. — 2-е изд.— М.: Бином, 2014. — 324 с.
12. Гусева А.И. Учимся программировать: PASCAL 7.0. Задачи и методы их решения. — М.: Диалог–МИФИ, 2012. — 256 с.



13. Дагене В.А., Григас Г.К., Аугутис К.Ф. 100 задач по программированию. — М.: Просвещение, 2013. — 255 с.
14. Джеймс Фокселл. Освой самостоятельно Visual Basic.NET за 24 часа. — М.: Вильямс, 2012. — 183 с.
15. Дэн Кларк. Объектно-ориентированное программирование в Visual Basic .NET. — СПб.: Питер, 2013. — 128 с.
16. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. — М.: Наука, 2014. — 224 с.
17. Зак Д.О. Самоучитель Visual Basic.Net. — Киев: BHV; СПб.: Питер, 2013. — 215 с.
18. Кевин Янк. PHP и MySQL. От новичка к профессионалу. — М.: Эксмо, 2013. — 208 с.
19. Касьянов В.Н., Сабельфельд В.К. Сборник заданий по практикуму на ЭВМ. — М.: Наука, 2014. — 272 с.
20. Кетков Ю.А., Кетков А.П. Практика программирования: Visual Basic, C++ Builder. — СПб.: BHV, 2014. — 149 с.
21. Колтунов И.И. Информационные и дистанционные технологии в образовании. — ИНФРА-М, 2012. — 215 с.
22. Заботина Н.Н. Проектирование информационных систем. — ИНФРА-М, 2012. — 138 с.
23. Элизабет Робсон, Эрик Фримен. Изучаем HTML, XHTML и CSS 2-е изд. — СПб.: Питер, 2014. — 179 с.
24. Царев В.В., Кантарович А.А., Черимш В.В. Оценка конкурентоспособности предприятий. — М.: ЮНИТИ-ДАНА, 2012. — 799 с.
25. Голицына О.Л., Максимов Н.В., Попов И. И. Базы данных: учебное пособие 2-е изд., испр. и доп. — М.: Форум: ИНФРА-М, 2013. — 231 с.

## **ПРИЛОЖЕНИЯ**

### **Приложение 1**

#### **Техническое задание**

##### **1.1. Наименование АИС**

Наименование — «Учебная часть».

##### **1.2. Краткая характеристика области применения программы**

Программа предназначена к применению в учебной части техникума.

#### **2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ**

##### **2.1. Основание для проведения разработки**

Основанием для проведения разработки является необходимость автоматизации учёта результатов оценки качества обучения студентов техникума и обеспечения лёгкого доступа сотрудников учебной части техникума к информации о них.

##### **2.2. Наименование и условное обозначение темы разработки**

Наименование темы разработки — «Разработка автоматизированной информационной системы учебной части».

Условное обозначение темы разработки (шифр темы) — «А.В.00001».

#### **3. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**

##### **3.1. Назначение программы**

Основное назначение АИС — хранение подробной информации о результатах оценки качества обучения студентов техникума и обеспечение лёгкого и быстрого доступа к ней сотрудников учебной части техникума.

Данная система позволяет вести учёт результатов оценки качества обучения студентов техникума и обеспечивает лёгкий и быстрый доступ к ним сотрудников учебной части техникума.

##### **3.2. Область применения программы**

Система предназначена для работы в отделах техникума, где существуют рабочие места, оборудованные средствами электронной вычислительной техники.

#### **4. ТРЕБОВАНИЯ К ПРОГРАММЕ**

##### **4.1. Требования к функциональным характеристикам**

###### **4.1.1. Состав выполняемых функций**

В результате проведения работ по данному техническому заданию должна быть создана система, выполняющая следующие функции:

1. Поддержка ввода информации о показателях оценки качества обучения студентов:

- ввод записей об отделениях техникума;
- ввод записей о семестрах;
- ввод записей об изучаемых предметах;
- ввод записей о группах студентов техникума;
- ввод записей о студентах техникума;
- ввод записей о семестровых оценках студентов техникума по изучаемым ими предметам;
- ввод записей о количестве пропущенных учебных занятий студентами техникума.

2. Поддержка быстрого и простого доступа к данным:

- получение информации о хранящихся в системе данных;
- получение информации о результатах оценки качества обучения студентов техникума.

###### **4.1.2. Требования к организации входных данных**

Входные данные вводятся в специальные поля ввода, содержащиеся в интерфейсе пользователя (программа). Рядом с каждым полем ввода указана подпись, говорящая о том, какой тип данных нужно вводить в это поле и какой смысл должны нести эти данные. Особых требований к вводу не предъявляется.

###### **4.1.3. Требования к организации выходных данных**

Выходные данные приходят от базы данных в виде структур данных и отображаются в программе в виде таблиц.

###### **4.1.4. Требования к временным характеристикам**

## Продолжение приложения 1

Время отклика программы не должно превышать стандартное значение не более минуты.

### 4.2. Требования к надежности

#### 4.2.1. Требования к обеспечению надежного (устойчивого) функционирования системы

Надежное (устойчивое) функционирование системы должно быть обеспечено выполнением совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- а) организацией бесперебойного питания технических средств;
- б) регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;
- в) регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов;
- г) необходимым уровнем квалификации сотрудников профильных подразделений.
- д) целостность сетевой инфраструктуры, обеспечивающей связь между рабочими станциями, на которых функционирует система.
- е) периодическое резервное копирование баз данных и хранение информации на съёмных носителях

#### 4.2.2. Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы, при условии соблюдения условий эксплуатации технических и программных средств.

## Продолжение приложения 1

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

### 4.2.3. Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

### 4.3. Условия эксплуатации

#### 4.3.1. Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

#### 4.3.2. Требования к видам обслуживания

См. Требования к обеспечению надежного (устойчивого) функционирования программы.

#### 4.3.3. Требования к численности и квалификации персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц — системный программист и конечный пользователь программы — оператор.

Системный программист должен иметь минимум среднее техническое образование.

В перечень задач, выполняемых системным программистом, должны входить:

- а) задача поддержания работоспособности технических средств;
- б) задачи установки (инсталляции) и поддержания работоспособности системных программных средств — операционной системы;

## Продолжение приложения 1

- в) задача установки (инсталляции) подпрограмм системы;
- г) установка и настройка СУБД.

Изначально обязанности, перечисленные в пунктах (в) и (г) берёт на себя разработчик.

Конечный пользователь программы (оператор) должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

Персонал должен быть аттестован минимум на II квалификационную группу по электробезопасности (для работы с конторским оборудованием).

### 4.4. Требования к составу и параметрам технических средств

В состав технических средств должен входить IBM-совместимый персональный компьютер (ПЭВМ), включающий в себя:

- а) процессор Pentium-4 или аналогичный с тактовой частотой 1700 МГц, не менее;
- б) оперативную память объемом, 256 МБ, не менее;
- в) жесткий диск объемом 100 Гб, и выше;
- г) оптический манипулятор типа «мышь»;

### 4.5. Требования к информационной и программной совместимости

#### 4.5.1. Требования к информационным структурам и методам решения

Требования к информационным структурам на входе и выходе, а также к методам решения не предъявляются.

#### 4.5.2. Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть реализованы на языке ObjectPascal. В качестве интегрированной среды разработки программы предлагается использовать Delphi.

#### 4.5.3. Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены локализованной версией операционной системы Windows.

#### 4.5.4. Требования к защите информации и программ

Требования к защите информации и программ не предъявляются.

### 4.6. Специальные требования

Специальные требования к программе не предъявляются.

## 5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

### 5.1. Предварительный состав программной документации

Состав программной документации должен включать в себя:

- 1) техническое задание;
- 2) спецификация;
- 3) текст программы;
- 4) описание системы и её подпрограмм;
- 5) систему тестирования и методики испытаний;
- 6) пояснительная записка;
- 7) ведомость эксплуатационных документов;
- 8) формуляр;
- 9) описание применения;
- 10) руководство системного программиста;
- 11) руководство программиста;
- 12) руководство оператора.

### 5.2. Специальные требования к программной документации

Специальные требования к программной документации не предъявляются.

## 6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

### 6.1. Ориентировочная экономическая эффективность

Ориентировочная экономическая эффективность рассчитывается по стандартной методике.

### 6.2. Предполагаемая годовая потребность

Предполагаемое число использования системы 240 в год — в течение рабочих дней.

### 6.3. Экономические преимущества разработки

## Продолжение приложения 1

АИС «Учебная часть» позволяет структурировать данные о качестве обучения студентов техникума, а также обеспечить быстрый доступ к данной информации сотрудникам учебной части техникума. Данные функции системы позволяют повысить скорость обработки информации о параметрах качества обучения студентов техникума. Это поможет повысить эффективность работы техникума и уменьшить трудозатраты работников учебной части техникума.

### 7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

#### 7.1. Стадии разработки

Разработка должна быть проведена в три стадии:

- 1) разработка технического задания;
- 2) рабочее проектирование;
- 3) внедрение.

#### 7.2. Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

- 1) разработка подпрограмм системы (ведётся параллельно для каждой подпрограммы);
- 2) разработка программной документации;
- 3) испытания системы.

На стадии внедрения должен быть выполнен этап разработки — подготовка и передача программы.

#### 7.3. Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

- 1) постановка задачи;
- 2) определение и уточнение требований к техническим средствам;
- 3) определение требований к системе и определение основных подсистем;



## **Продолжение приложения 1**

4) определение стадий, этапов и сроков разработки программы и документации на неё;

5) выбор языков программирования;

6) согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77 и требованием п. «Предварительный состав программной документации» настоящего технического задания.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1) разработка, согласование и утверждение программы и методики испытаний;

2) проведение приемо-сдаточных испытаний;

3) корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию.

### **7.4. Исполнители**

Руководитель разработки

Ответственный исполнитель

Исполнитель

## **8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ**

### **8.1. Виды испытаний**

Приемо-сдаточные испытания программы должны проводиться согласно разработанной и согласованной схеме «Программы и методики испытаний».

Ход проведения приемо-сдаточных испытаний документируется в

протоколе проведения испытаний.

**8.2. Общие требования к приемке работы**

После проведения испытаний в полном объеме, на основании «Протокола испытаний» утверждают «Свидетельство о приемке» и производят запись в программном документе «Формуляр».

## Код базы данных на языке SQL

```
CREATE TABLE Группы
```

```
(
    Код          int identity NOT NULL,
    Код_отделения int NULL,
    Название     nvarchar(255) NULL
);
```

```
ALTER TABLE Группы
```

```
ADD PRIMARY KEY (Код);
```

```
CREATE TABLE Отделения
```

```
(
    Код          int identity NOT NULL,
    Номер        int NULL,
    ФИО_заведующего nvarchar(255) NULL
);
```

```
ALTER TABLE Отделения
```

```
ADD PRIMARY KEY (Код);
```

```
CREATE TABLE Оценки
```

```
(
    Код          int identity NOT NULL,
    Код_семестра int NULL,
    Код_предмета int NULL,
    Код_студента int NULL,
    Оценка       int NULL
);
```

);

ALTER TABLE Оценки

ADD PRIMARY KEY (Код);

CREATE TABLE Предметы

(

Код int identity NOT NULL,

Название nvarchar(255) NULL

);

ALTER TABLE Предметы

ADD PRIMARY KEY (Код);

CREATE TABLE Пропуски

(

Код int identity NOT NULL,

Код\_семестра int NULL,

Код\_студента int NULL,

Всего int NULL,

Закрыто int NULL

);

ALTER TABLE Пропуски

ADD PRIMARY KEY (Код);

CREATE TABLE Семестры

(

Код int identity NOT NULL,

Номер int NULL

);

ALTER TABLE Семестры

ADD PRIMARY KEY (Код);

CREATE TABLE Студенты

(

Код int identity NOT NULL,

Код\_группы int NULL,

ФИО nvarchar(255) NULL

);

ALTER TABLE Студенты

ADD PRIMARY KEY (Код);

ALTER TABLE Группы

ADD CONSTRAINT Группы\_FK00 FOREIGN KEY (Код\_отделения)  
REFERENCES Отделения (Код)  
ON DELETE CASCADE;

ALTER TABLE Оценки

ADD CONSTRAINT Оценки\_FK00 FOREIGN KEY (Код\_предмета)  
REFERENCES Предметы (Код)  
ON DELETE CASCADE;

ALTER TABLE Оценки

ADD CONSTRAINT Оценки\_FK01 FOREIGN KEY (Код\_семестра)  
REFERENCES Семестры (Код)

ONDELETECASCADE;

ALTER TABLE Оценки

ADD CONSTRAINT Оценки\_FK02 FOREIGN KEY (Код\_студента)  
REFERENCES Студенты (Код)  
ON DELETE CASCADE;

ALTER TABLE Пропуски

ADD CONSTRAINT Пропуски\_FK00 FOREIGN KEY (Код\_семестра)  
REFERENCES Семестры (Код)  
ON DELETE CASCADE;

ALTER TABLE Пропуски

ADD CONSTRAINT Пропуски\_FK01 FOREIGN KEY (Код\_студента)  
REFERENCES Студенты (Код)  
ON DELETE CASCADE;

ALTER TABLE Студенты

ADD CONSTRAINT Студенты\_FK00 FOREIGN KEY (Код\_группы)  
REFERENCES Группы (Код)  
ON DELETE CASCADE;

### Главный запрос

```

SELECT      r.Название,      IIf(r.kolvo>0,r.kolvo,'-')      AS      kolvo,
IIf(r.ocenok5>0,r.ocenok5,'-') AS ocenok5, IIf(r.ocenok54>0,r.ocenok54,'-') AS
ocenok54,      IIf(r.ocenok2>0,r.ocenok2,'-')      AS      ocenok2,
IIf(r.kolvo>0,IIf(Round((r.kolvo-r.ocenok2)/r.kolvo*100,0)=(r.kolvo-
r.ocenok2)/r.kolvo*100,(r.kolvo-r.ocenok2)/r.kolvo*100,round((r.kolvo-
r.ocenok2)/r.kolvo*100,1)),'-')      AS      usp,      IIf(r.kolvo>0      and
r.ocenok5+r.ocenok54>0,IIf(Round((r.ocenok5+r.ocenok54)/r.kolvo*100,0)=(r.oceno
k5+r.ocenok54)/r.kolvo*100,(r.ocenok5+r.ocenok54)/r.kolvo*100,round((r.ocenok5+
r.ocenok54)/r.kolvo*100,1)),'-') AS kach, IIf(r.zakr>0,r.zakr,'-') AS zakr,
IIf(r.nu>0,r.nu,'-') AS nu, IIf(r.kolvo>0 and r.nu>0,IIf(Round(r.nu / r.kolvo,0)=r.nu /
r.kolvo,r.nu / r.kolvo,round(r.nu / r.kolvo,1)),'-') AS prop

```

```

FROM (SELECT
g.Название,
g.Код,
(SELECT Count(Студенты.Код)
FROMГруппыINNERJOINСтудентыONГруппы.Код
Студенты.[Кодгруппы]
WHEREГруппы.Код=g.кодGROUPBYГруппы.Код) as
kolvo,
(Select count(minoc) From (
SELECT      Группы.Название      as      nazv,      Группы.Код      as      kod,
Min(Оценки.Оценка) AS minoc,
Студенты.Код as stkod
FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN
Оценки ON Семестры.Код = Оценки.[Кодсеместра]) ON Студенты.Код =

```

### Продолжение приложения 3

```
Оценки.[Кодстудента]) ON Отделения.Код = Группы.[Кодотделения]
WHERE (((Семестры.Номер)=:0) AND ((Отделения.Номер)=:1))
GROUP BY Группы.Название, Группы.Код, Студенты.Код
ORDER BY Студенты.Код)
where minoc=5 and kod=g.код
group by nazv, kod, minoc) As
ocenok5,
(Select count(minoc) From (
SELECT    Группы.Название    as    nazv,    Группы.Код    as    kod,
Min(Оценки.Оценка) AS minoc,
Студенты.Код as stkod
FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN
Оценки ON Семестры.Код = Оценки.[Кодсеместра]) ON Студенты.Код =
Оценки.[Кодстудента]) ON Отделения.Код = Группы.[Кодотделения]
WHERE (((Семестры.Номер)=:2) AND ((Отделения.Номер)=:3))
GROUP BY Группы.Название, Группы.Код, Студенты.Код
ORDER BY Студенты.Код)
where minoc=4 and kod=g.код
group by nazv, kod, minoc) As
ocenok54,
(Select count(minoc) From (
SELECT    Группы.Название    as    nazv,    Группы.Код    as    kod,
Min(Оценки.Оценка) AS minoc,
Студенты.Код as stkod
FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN
Оценки ON Семестры.Код = Оценки.[Кодсеместра]) ON Студенты.Код =
Оценки.[Кодстудента]) ON Отделения.Код = Группы.[Кодотделения]
```



### Продолжение приложения 3

```
WHERE (((Семестры.Номер)=:4) AND ((Отделения.Номер)=:5))
GROUP BY Группы.Название, Группы.Код, Студенты.Код
ORDER BY Студенты.Код)
where minoc=2 and kod=g.код
group by nazv, kod, minoc) Asocenok2,
(SELECT Sum(Пропуски.Всего)
FROM      ((ОтделенияINNERJOINГруппыONОтделения.Код      =
Группы.[Кодотделения])
INNERJOINСтудентыONГруппы.Код      =      Студенты.[Кодгруппы])
INNERJOIN      (СеместрыINNERJOINПропускиONСеместры.Код      =
Пропуски.[Кодсеместра]) ONСтуденты.Код = Пропуски.[Кодстудента]
WHERE      (((Семестры.Номер)=:6)      AND      ((Отделения.Номер)=:7))
andгруппы.код=g.код
GROUP BY Группы.Код) AS
vsego,
(SELECTSum(Пропуски.Закрыто)
FROM      ((Отделения INNER JOIN Группы ON Отделения.Код =
Группы.[Код отделения])
INNER JOIN Студенты ON Группы.Код = Студенты.[Код группы])
INNER JOIN (Семестры INNER JOIN Пропуски ON Семестры.Код =
Пропуски.[Код семестра]) ON Студенты.Код = Пропуски.[Код студента]
WHERE      (((Семестры.Номер)=:8) AND ((Отделения.Номер)=:9)) and
группы.код=g.код
GROUP BY Группы.Код) AS
zakr,
vsego-zakr AS
nu
FROM
группы AS g) AS r;
```

**Запрос «Место по успеваемости»**

```

SELECT Название, usp from (
    SELECT      r.Название,      IIf(r.kolvo>0,r.kolvo,'-')      AS      kolvo,
    IIf(r.ocenok5>0,r.ocenok5,'-') AS ocenok5, IIf(r.ocenok54>0,r.ocenok54,'-') AS
ocenok54,      IIf(r.ocenok2>0,r.ocenok2,'-')      AS      ocenok2,
    IIf(r.kolvo>0,IIf(Round((r.kolvo-r.ocenok2)/r.kolvo*100,
0)=(r.kolvo-
r.ocenok2)/r.kolvo*100,(r.kolvo-r.ocenok2)/r.kolvo*100,round((r.kolvo-
r.ocenok2)/r.kolvo*100,1)),'-')      AS      usp,      IIf(r.kolvo>0      and
r.ocenok5+r.ocenok54>0,IIf(Round((r.ocenok5+r.ocenok54)/r.kolvo*100,0)=(r.oceno
k5+r.ocenok54)/r.kolvo*100,(r.ocenok5+r.ocenok54)/r.kolvo*100,round((r.ocenok5+
r.ocenok54)/r.kolvo*100,1)),'-') AS kach, IIf(r.zakr>0,r.zakr,'-') AS zakr,
IIf(r.nu>0,r.nu,'-') AS nu, IIf(r.kolvo>0 and r.nu>0,IIf(Round(r.nu / r.kolvo,0)=r.nu /
r.kolvo,r.nu / r.kolvo,round(r.nu / r.kolvo,1)),'-') AS prop

```

```

FROM (SELECT
g.Название,
g.Код,
(SELECT Count(Студенты.Код)
FROM Группы INNER JOIN Студенты ON Группы.Код =
Студенты.[Кодгруппы]
WHERE Группы.Код=g.код GROUP BY Группы.Код) as
kolvo,
(Select count(minoc) From (
SELECT      Группы.Название      as      nazv,      Группы.Код      as      kod,
Min(Оценки.Оценка) AS minoc,
Студенты.Код as stkod
FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN

```

#### Продолжение приложения 4

ОценкиONСеместры.Код = Оценки.[Кодсеместра]) ONСтуденты.Код =  
Оценки.[Кодстудента]) ONОтделения.Код = Группы.[Кодотделения]

WHERE (((Семестры.Номер)=:0) AND ((Отделения.Номер)=:1))

GROUP BY Группы.Название, Группы.Код, Студенты.Код

ORDER BY Студенты.Код)

where minoc=5 and kod=g.код

group by nazv, kod, minoc) As

ocenok5,

(Select count(minoc) From (

SELECT Группы.Название as nazv, Группы.Код as kod,

Min(Оценки.Оценка) AS minoc,

Студенты.Код as stkod

FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON  
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN  
Оценки ON Семестры.Код = Оценки.[Кодсеместра]) ON Студенты.Код =  
Оценки.[Кодстудента]) ON Отделения.Код = Группы.[Кодотделения]

WHERE (((Семестры.Номер)=:2) AND ((Отделения.Номер)=:3))

GROUP BY Группы.Название, Группы.Код, Студенты.Код

ORDER BY Студенты.Код)

where minoc=4 and kod=g.код

group by nazv, kod, minoc) As

ocenok54,

(Select count(minoc) From (

SELECT Группы.Название as nazv, Группы.Код as kod,

Min(Оценки.Оценка) AS minoc,

Студенты.Код as stkod

FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON  
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN  
Оценки ON Семестры.Код = Оценки.[Кодсеместра]) ON Студенты.Код =

#### Продолжение приложения 4

```
Оценки.[Кодстудента]) ON Отделения.Код = Группы.[Кодотделения]
WHERE (((Семестры.Номер)=:4) AND ((Отделения.Номер)=:5))
GROUP BY Группы.Название, Группы.Код, Студенты.Код
ORDER BY Студенты.Код)
where minoc=2 and kod=g.код
group by nazv, kod, minoc) As
ocenok2,
(SELECT Sum(Пропуски.Всего)
FROM      ((ОтделенияINNERJOINГруппыONОтделения.Код      =
Группы.[Кодотделения])
INNERJOINСтудентыONГруппы.Код      =      Студенты.[Кодгруппы])
INNERJOIN      (СеместрыINNERJOINПропускиONСеместры.Код      =
Пропуски.[Кодсеместра]) ONСтуденты.Код = Пропуски.[Кодстудента]
WHERE      (((Семестры.Номер)=:6)      AND      ((Отделения.Номер)=:7))
andгруппы.код=g.код
GROUP BY Группы.Код) AS
vsego,
(SELECTSum(Пропуски.Закрыто)
FROM      ((Отделения INNER JOIN Группы ON Отделения.Код =
Группы.[Код отделения])
INNER JOIN Студенты ON Группы.Код = Студенты.[Код группы])
INNER JOIN (Семестры INNER JOIN Пропуски ON Семестры.Код =
Пропуски.[Код семестра]) ON Студенты.Код = Пропуски.[Код студента]
WHERE      (((Семестры.Номер)=:8)      AND      ((Отделения.Номер)=:9)) and
группы.код=g.код
GROUP BY Группы.Код) AS
zacr,
vsego-zacr AS nu
FROM группы AS g) AS r)order by uspdesc;
```

**Запрос «Место по качеству»**

```

SELECT Название, kach from (
    SELECT      r.Название,      IIf(r.kolvo>0,r.kolvo,'-')      AS      kolvo,
    IIf(r.ocenok5>0,r.ocenok5,'-') AS ocenok5, IIf(r.ocenok54>0,r.ocenok54,'-') AS
ocenok54,      IIf(r.ocenok2>0,r.ocenok2,'-')      AS      ocenok2,
    IIf(r.kolvo>0,IIf(Round((r.kolvo-r.ocenok2)/r.kolvo*100,
0)=(r.kolvo-
r.ocenok2)/r.kolvo*100,(r.kolvo-r.ocenok2)/r.kolvo*100,round((r.kolvo-
r.ocenok2)/r.kolvo*100,1)),'-')      AS      usp,      IIf(r.kolvo>0      and
r.ocenok5+r.ocenok54>0,IIf(Round((r.ocenok5+r.ocenok54)/r.kolvo*100,0)=(r.oceno
k5+r.ocenok54)/r.kolvo*100,(r.ocenok5+r.ocenok54)/r.kolvo*100,round((r.ocenok5+
r.ocenok54)/r.kolvo*100,1)),'-') AS kach, IIf(r.zakr>0,r.zakr,'-') AS zakr,
IIf(r.nu>0,r.nu,'-') AS nu, IIf(r.kolvo>0 and r.nu>0,IIf(Round(r.nu / r.kolvo,0)=r.nu /
r.kolvo,r.nu / r.kolvo,round(r.nu / r.kolvo,1)),'-') AS prop
    FROM (SELECT
        g.Название,
        g.Код,
        (SELECT Count(Студенты.Код)
        FROM Группы INNER JOIN Студенты ON Группы.Код =
Студенты.[Кодгруппы]
        WHERE Группы.Код=g.код GROUP BY Группы.Код) as
        kolvo,
        (Select count(minoc) From (
            SELECT      Группы.Название      as      nazv,      Группы.Код      as      kod,
            Min(Оценки.Оценка) AS minoc,
            Студенты.Код as stkod
            FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN
Оценки ON Семестры.Код = Оценки.[Кодсеместра]) ON Студенты.Код =

```

## Продолжение приложения 5

```
Оценки.[Кодстудента]) ON Отделения.Код = Группы.[Кодотделения]
WHERE (((Семестры.Номер)=:0) AND ((Отделения.Номер)=:1))
GROUP BY Группы.Название, Группы.Код, Студенты.Код
ORDER BY Студенты.Код)
where minoc=5 and kod=g.код
group by nazv, kod, minoc) As
ocenok5,
(Select count(minoc) From (
SELECT    Группы.Название    as    nazv,    Группы.Код    as    kod,
Min(Оценки.Оценка) AS minoc,
Студенты.Код as stkod
FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN
Оценки ON Семестры.Код = Оценки.[Кодсеместра]) ON Студенты.Код =
Оценки.[Кодстудента]) ON Отделения.Код = Группы.[Кодотделения]
WHERE (((Семестры.Номер)=:2) AND ((Отделения.Номер)=:3))
GROUP BY Группы.Название, Группы.Код, Студенты.Код
ORDER BY Студенты.Код)
where minoc=4 and kod=g.код
group by nazv, kod, minoc) As
ocenok54,
(Select count(minoc) From (
SELECT    Группы.Название    as    nazv,    Группы.Код    as    kod,
Min(Оценки.Оценка) AS minoc,
Студенты.Код as stkod
FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN
Оценки ON Семестры.Код = Оценки.[Кодсеместра]) ON Студенты.Код =
Оценки.[Кодстудента]) ON Отделения.Код = Группы.[Кодотделения]
```

## Продолжение приложения 5

```
WHERE (((Семестры.Номер)=:4) AND ((Отделения.Номер)=:5))
GROUP BY Группы.Название, Группы.Код, Студенты.Код
ORDER BY Студенты.Код)
where minoc=2 and kod=g.код
group by nazv, kod, minoc) As
ocenok2,
(SELECT Sum(Пропуски.Всего)
FROM      ((ОтделенияINNERJOINГруппыONОтделения.Код      =
Группы.[Кодотделения])
INNERJOINСтудентыONГруппы.Код      =      Студенты.[Кодгруппы])
INNERJOIN      (СеместрыINNERJOINПропускиONСеместры.Код      =
Пропуски.[Кодсеместра]) ONСтуденты.Код = Пропуски.[Кодстудента]
WHERE      (((Семестры.Номер)=:6)      AND      ((Отделения.Номер)=:7))
andгруппы.код=g.код
GROUP BY Группы.Код) AS
vsego,
(SELECTSum(Пропуски.Закрыто)
FROM      ((Отделения INNER JOIN Группы ON Отделения.Код =
Группы.[Код отделения])
INNER JOIN Студенты ON Группы.Код = Студенты.[Код группы])
INNER JOIN (Семестры INNER JOIN Пропуски ON Семестры.Код =
Пропуски.[Код семестра]) ON Студенты.Код = Пропуски.[Код студента]
WHERE      (((Семестры.Номер)=:8)      AND      ((Отделения.Номер)=:9)) and
группы.код=g.код
GROUP BY Группы.Код) AS
zakr,
vsego-zakr AS
nu
FROM группы AS g) AS r)order by uspdesc;
```

**Запрос «Место по пропускам»**

```

SELECT Название, prop from (
    SELECT      r.Название,      IIf(r.kolvo>0,r.kolvo,'-')      AS      kolvo,
    IIf(r.ocenok5>0,r.ocenok5,'-') AS ocenok5, IIf(r.ocenok54>0,r.ocenok54,'-') AS
ocenok54,      IIf(r.ocenok2>0,r.ocenok2,'-')      AS      ocenok2,
    IIf(r.kolvo>0,IIf(Round((r.kolvo-r.ocenok2)/r.kolvo*100,
0)=(r.kolvo-
r.ocenok2)/r.kolvo*100,(r.kolvo-r.ocenok2)/r.kolvo*100,round((r.kolvo-
r.ocenok2)/r.kolvo*100,1)),'-')      AS      usp,      IIf(r.kolvo>0      and
r.ocenok5+r.ocenok54>0,IIf(Round((r.ocenok5+r.ocenok54)/r.kolvo*100,0)=(r.oceno
k5+r.ocenok54)/r.kolvo*100,(r.ocenok5+r.ocenok54)/r.kolvo*100,round((r.ocenok5+
r.ocenok54)/r.kolvo*100,1)),'-') AS kach, IIf(r.zakr>0,r.zakr,'-') AS zakr,
IIf(r.nu>0,r.nu,'-') AS nu, IIf(r.kolvo>0 and r.nu>0,IIf(Round(r.nu / r.kolvo,0)=r.nu /
r.kolvo,r.nu / r.kolvo,round(r.nu / r.kolvo,1)),'-') AS prop

```

```

FROM (SELECT
g.Название,
g.Код,
(SELECT Count(Студенты.Код)
FROM Группы INNER JOIN Студенты ON Группы.Код =
Студенты.[Кодгруппы]
WHERE Группы.Код=g.код GROUP BY Группы.Код) as
kolvo,
(Select count(minoc) From (
SELECT      Группы.Название      as      nazv,      Группы.Код      as      kod,
Min(Оценки.Оценка) AS minoc,
Студенты.Код as stkod
FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN

```



## Продолжение приложения 6

Оценки.Оценка ON Семестры.Код = Оценки.[Кодсеместра]) ON Студенты.Код =  
Оценки.[Кодстудента]) ON Отделения.Код = Группы.[Кодотделения]

WHERE (((Семестры.Номер)=:0) AND ((Отделения.Номер)=:1))

GROUP BY Группы.Название, Группы.Код, Студенты.Код

ORDER BY Студенты.Код)

where minoc=5 and kod=g.код

group by nazv, kod, minoc) As

ocenok5,

(Select count(minoc) From (

SELECT Группы.Название as nazv, Группы.Код as kod,

Min(Оценки.Оценка) AS minoc,

Студенты.Код as stkod

FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON  
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN  
Оценки ON Семестры.Код = Оценки.[Кодсеместра]) ON Студенты.Код =  
Оценки.[Кодстудента]) ON Отделения.Код = Группы.[Кодотделения]

WHERE (((Семестры.Номер)=:2) AND ((Отделения.Номер)=:3))

GROUP BY Группы.Название, Группы.Код, Студенты.Код

ORDER BY Студенты.Код)

where minoc=4 and kod=g.код

group by nazv, kod, minoc) As

ocenok54,

(Select count(minoc) From (

SELECT Группы.Название as nazv, Группы.Код as kod,

Min(Оценки.Оценка) AS minoc,

Студенты.Код as stkod

FROM Отделения INNER JOIN ((Группы INNER JOIN Студенты ON  
Группы.Код = Студенты.[Кодгруппы]) INNER JOIN (Семестры INNER JOIN  
Оценки ON Семестры.Код = Оценки.[Кодсеместра]) ON Студенты.Код =

## Продолжение приложения 6

```

Оценки.[Кодстудента]) ON Отделения.Код = Группы.[Кодотделения]
WHERE (((Семестры.Номер)=:4) AND ((Отделения.Номер)=:5))
GROUP BY Группы.Название, Группы.Код, Студенты.Код
ORDER BY Студенты.Код)
where minoc=2 and kod=g.код
group by nazv, kod, minoc) As
ocenok2,
(SELECT Sum(Пропуски.Всего)
FROM      ((ОтделенияINNERJOINГруппыONОтделения.Код      =
Группы.[Кодотделения])
INNERJOINСтудентыONГруппы.Код      =      Студенты.[Кодгруппы])
INNERJOIN      (СеместрыINNERJOINПропускиONСеместры.Код      =
Пропуски.[Кодсеместра]) ONСтуденты.Код = Пропуски.[Кодстудента]
WHERE      (((Семестры.Номер)=:6)      AND      ((Отделения.Номер)=:7))
andгруппы.код=g.код
GROUP BY Группы.Код) AS
vsego,
(SELECTSum(Пропуски.Закрыто)
FROM      ((Отделения INNER JOIN Группы ON Отделения.Код =
Группы.[Код отделения])
INNER JOIN Студенты ON Группы.Код = Студенты.[Код группы])
INNER JOIN (Семестры INNER JOIN Пропуски ON Семестры.Код =
Пропуски.[Код семестра]) ON Студенты.Код = Пропуски.[Код студента]
WHERE      (((Семестры.Номер)=:8)      AND      ((Отделения.Номер)=:9)) and
группы.код=g.код
GROUP BY Группы.Код) AS
zacr,
vsego-zacr AS nu
FROM группы AS g) AS r)order by usp;

```

## Обработчики событий для подготовки формирования отчёта

```
procedure TfrmReport.cmdCloseClick(Sender: TObject);
begin
  Close;
end;
```

```
procedure TfrmReport.cmdPrintClick(Sender: TObject);
begin
  wbrReport.ExecWB(6, 1);
end;
```

```
procedure TfrmReport.cmdPrintPreviewClick(Sender: TObject);
begin
  wbrReport.ExecWB(7, 0);
end;
```

```
procedure TfrmReport.Execute(s: String);
begin
  strURL := s;

  wbrReport.Navigate('about:blank');

  ShowModal;
end;
```

```
procedure TfrmReport.wbrReportDocumentComplete(ASender: TObject;
  const pDisp: IDispatch; var URL: OleVariant);
```

```
Var
  d: OLEVariant;
```

```
t: String;
t1: String;
t2: String;
t3: String;
```

```
i: Integer;
```

```
j: Integer;
```

```
c: Integer;
```

```

begin
  If URL = 'about:blank' Then
  Begin
    d := wbrReport.Document;

    d.Body.InnerHTML := strURL;

    With dtmMain Do
    Begin
      qryTemp.SQL.Clear;

      t := 'Select * From Временная Order By Баллы';

      qryTemp.SQL.Add(t);

      qryTemp.Open;

      qryTemp.First;

      c := qryTemp.Recordset.RecordCount;

      i := 1;

      While Not qryTemp.EOF Do
      Begin
        For j := 1 To c Do
        Begin
          t := qryTemp.FieldName('Группа').AsString;

          If d.GetElementByID('gtd' + IntToStr(j)).InnerHTML = t Then
          Begin
            If i = 1 Then
            Begin
              t1 := t;
            End;

            If i = 2 Then
            Begin
              t2 := t;
            End;

            If i = 3 Then
            Begin

```

```

        t3 := t;
    End;

    d.GetElementByID('td' + IntToStr(j)).InnerHTML :=
        IntToStr(i);
    End;
End;

i := i + 1;

qryTemp.Next;
End;

t := '<p>';
t := t + '1 <i>место — группа ' + t1;
t := t + '</i></p>';

t := t + '<p>';
t := t + '2 <i>место — группа ' + t2;
t := t + '</i></p>';

t := t + '<p>';
t := t + '3 <i>место — группа ' + t3;
t := t + '</i></p>';

d.GetElementByID('mst').InnerHTML := t;

t := '</font>';

t := t + '<table width="100%">';

t := t + '<tr><th width="50%"><font size="4" face="Arial">';
t := t + 'Заведующая отделением</font></th>' +
'<th><font size="4" face="Arial">' +
    dtmMain.tblOtd.FieldName('ФИОзаведующего').AsString +
'</font></th></tr>';
t := t + '</table>';

d.Body.InnerHTML := d.Body.InnerHTML + t;
qryTemp.Close;
End;
End;
end;
```

## Программный код генерации отчёта

```
procedure TfrmMain.mnuMainFileReportClick(Sender: TObject);
```

```
Var
```

```
  s: String;
```

```
  t: String;
```

```
  y: Word;
```

```
  m: Word;
```

```
  d: Word;
```

```
  i: Integer;
```

```
  c: Integer;
```

```
  intMestoUsp: Integer;
```

```
  intMestoKach: Integer;
```

```
  intMestoProp: Integer;
```

```
begin
```

```
  With dtmMain Do
```

```
  Begin
```

```
    //{
```

```
    dtmMain.qryMain.Parameters.ParamByName('0').Value :=
```

```
    dtmMain.tblSem.FieldName('Homep').AsString;
```

```
    dtmMain.qryMain.Parameters.ParamByName('1').Value :=
```

```
dtmMain.tblOtd.FieldName('Homep').AsString;
```

```
dtmMain.qryMain.Parameters.ParamByName('2').Value :=  
dtmMain.tblSem.FieldName('Homep').AsString;
```

```
dtmMain.qryMain.Parameters.ParamByName('3').Value :=  
dtmMain.tblOtd.FieldName('Homep').AsString;
```

```
dtmMain.qryMain.Parameters.ParamByName('4').Value :=  
dtmMain.tblSem.FieldName('Homep').AsString;
```

```
dtmMain.qryMain.Parameters.ParamByName('5').Value :=  
dtmMain.tblOtd.FieldName('Homep').AsString;
```

```
dtmMain.qryMain.Parameters.ParamByName('6').Value :=  
dtmMain.tblSem.FieldName('Homep').AsString;
```

```
dtmMain.qryMain.Parameters.ParamByName('7').Value :=  
dtmMain.tblOtd.FieldName('Homep').AsString;
```

```
dtmMain.qryMain.Parameters.ParamByName('8').Value :=  
dtmMain.tblSem.FieldName('Homep').AsString;
```

```
dtmMain.qryMain.Parameters.ParamByName('9').Value :=  
dtmMain.tblOtd.FieldName('Homep').AsString;
```

```
//}
```

```
dtmMain.qryMestoUsp.Parameters.ParamByName('0').Value :=  
dtmMain.tblSem.FieldName('Homep').AsString;
```

```
dtmMain.qryMestoUsp.Parameters.ParamByName('1').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoUsp.Parameters.ParamByName('2').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoUsp.Parameters.ParamByName('3').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoUsp.Parameters.ParamByName('4').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoUsp.Parameters.ParamByName('5').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoUsp.Parameters.ParamByName('6').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoUsp.Parameters.ParamByName('7').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoUsp.Parameters.ParamByName('8').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoUsp.Parameters.ParamByName('9').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoKach.Parameters.ParamByName('0').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```



```
dtmMain.qryMestoKach.Parameters.ParamByName('1').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoKach.Parameters.ParamByName('2').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoKach.Parameters.ParamByName('3').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoKach.Parameters.ParamByName('4').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoKach.Parameters.ParamByName('5').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoKach.Parameters.ParamByName('6').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoKach.Parameters.ParamByName('7').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoKach.Parameters.ParamByName('8').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoKach.Parameters.ParamByName('9').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoProp.Parameters.ParamByName('0').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

## Продолжение приложения 8

```
dtmMain.qryMestoProp.Parameters.ParamByName('1').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoProp.Parameters.ParamByName('2').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoProp.Parameters.ParamByName('3').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoProp.Parameters.ParamByName('4').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoProp.Parameters.ParamByName('5').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoProp.Parameters.ParamByName('6').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoProp.Parameters.ParamByName('7').Value :=  
dtmMain.tblOtd.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoProp.Parameters.ParamByName('8').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
dtmMain.qryMestoProp.Parameters.ParamByName('9').Value :=  
dtmMain.tblSem.FieldByName('Homep').AsString;
```

```
//dtmMain.qryMain.SQL.Add(s);
```

```
//dtmMain.qryMain.Parameters.Clear;

//qryMain.Prepared := True;

//qryMain.SQL.Clear;

//qryMain.SQL.Add(t);

qryMain.Open;

qryMain.First;

s := '<font face="Arial"><p align="center"><b><font size="5">';

DecodeDate(Now, y, m, d);

s := s + 'Итоги аттестации по результатам ' +
tblSem.FieldName('Код').AsString + ' семестра ';

If m < 9 Then
Begin
s := s + IntToStr(y - 1) + '—' + IntToStr(y);
End
Else
Begin
s := s + IntToStr(y) + '—' + IntToStr(y + 1);
End;

s := s + ' учебного года на отделении № ' +
```

## Продолжение приложения 8

```
tblOtd.FieldName('Homep').AsString + '</font></b></p>';
```

```
s := s + '<font size="1" face="Times New Roman"><table border="1"
bordercolor="#000000";
```

```
s := s +
' Style="border-collapse:collapse;border-style:solid";
```

```
s := s +
' cellpadding="0" cellspacing="0" width="100%";
```

```
s := s +
'>;
```

```
s := s + '<tr>;
```

```
s := s + '<td align="center">№ группы</td>;
```

```
s := s + '<td align="center">К-востудентов</td>;
```

```
s := s + '<td align="center">«5»</td>;
```

```
s := s + '<td align="center">«4,5»</td>;
```

```
s := s + '<td align="center">«2»</td>;
```

```
s := s + '<td align="center">Успеваемость, %</td>;
```

```
s := s + '<td align="center">Место</td>;
```

```
s := s + '<td align="center">Кач-вообучения, %</td>;
```

```
s := s + '<td align="center">Место</td>;
```

```
s := s + '<td align="center">Пропуски «ув»</td>;
```

```
s := s + '<td align="center">Пропуски «н/у»</td>;
```

```
s := s + '<td align="center">Пропуски н/у на 1 человека</td>;
```

```
s := s + '<td align="center">Место</td>;
```

```

s := s + '<td align="center">Баллы</td>';
s := s + '<td align="center">Итог</td>';

s := s + '</tr>';

c := 1;

qryTemp.SQL.Clear;

t := 'Delete From Временная';

qryTemp.SQL.Add(t);

qryTemp.ExecSQL;

While Not qryMain.EOF Do
Begin
  s := s + '<tr align="center"><td><font size="3" face="Times New
Roman">' +
  '<span id="gtd' + IntToStr(c) + '" name="gtd' + IntToStr(c) + '">' +
  qryMain.FieldName('Название').AsString + '</span></font></td>';

  s := s + '<td><font size="3" face="Times New Roman">' +
  qryMain.FieldName('kolvo').AsString + '</font></td>';

  s := s + '<td><font size="3" face="Times New Roman">' +
  qryMain.FieldName('ocenok5').AsString + '</font></td>';

  s := s + '<td><font size="3" face="Times New Roman">' +

```

```
qryMain.FieldName('ocenok54').AsString + '</font></td>';
```

```
s := s + '<td><font size="3" face="Times New Roman">' +  
qryMain.FieldName('ocenok2').AsString + '</font></td>';
```

```
s := s + '<td><font size="3" face="Times New Roman">' +  
qryMain.FieldName('usp').AsString + '</font></td>';
```

```
qryMestoUsp.Open;
```

```
qryMestoUsp.First;
```

```
intMestoUsp := 1;
```

```
While Not qryMestoUsp.EOF Do
```

```
Begin
```

```
  If qryMestoUsp.FieldName('Название').AsString =
```

```
  qryMain.FieldName('Название').AsString Then
```

```
  Begin
```

```
    Break;
```

```
  End;
```

```
  intMestoUsp := intMestoUsp + 1;
```

```
  qryMestoUsp.Next;
```

```
End;
```

```
qryMestoUsp.Close;
```

## Продолжение приложения 8

```
s := s + '<td><font size="3" face="Times New Roman">' +  
  IntToStr(intMestoUsp) + '</font></td>';
```

```
s := s + '<td><font size="3" face="Times New Roman">' +  
  qryMain.FieldByName('kach').AsString + '</font></td>';
```

```
qryMestoKach.Open;
```

```
qryMestoKach.First;
```

```
intMestoKach := 1;
```

```
While Not qryMestoKach.EOF Do
```

```
  Begin
```

```
    If qryMestoKach.FieldByName('Название').AsString =  
      qryMain.FieldByName('Название').AsString Then
```

```
      Begin
```

```
        Break;
```

```
      End;
```

```
    intMestoKach := intMestoKach + 1;
```

```
    qryMestoKach.Next;
```

```
  End;
```

```
qryMestoKach.Close;
```

```
s := s + '<td><font size="3" face="Times New Roman">' +  
  IntToStr(intMestoKach) + '</font></td>';
```

```
s := s + '<td><font size="3" face="Times New Roman">' +  
qryMain.FieldByName('zakr').AsString + '</font></td>';
```

```
s := s + '<td><font size="3" face="Times New Roman">' +  
qryMain.FieldByName('nu').AsString + '</font></td>';
```

```
s := s + '<td><font size="3" face="Times New Roman">' +  
qryMain.FieldByName('prop').AsString + '</font></td>';
```

```
qryMestoProp.Open;
```

```
qryMestoProp.First;
```

```
intMestoProp := 1;
```

```
While Not qryMestoProp.EOF Do
```

```
Begin
```

```
  If qryMestoProp.FieldByName('Название').AsString =
```

```
  qryMain.FieldByName('Название').AsString Then
```

```
  Begin
```

```
    Break;
```

```
  End;
```

```
  intMestoProp := intMestoProp + 1;
```

```
  qryMestoProp.Next;
```

```
End;
```

```
qryMestoProp.Close;
```



```
s := s + '<td><font size="3" face="Times New Roman">' +  
    IntToStr(intMestoProp) + '</font></td>';  
  
s := s + '<td><font size="3" face="Times New Roman">' +  
    IntToStr(intMestoUsp + intMestoKach + intMestoProp) + '</font></td>';  
  
qryTemp.SQL.Clear;  
  
t := 'Insert Into Временная (Группа, Баллы) Values (''  
    + qryMain.FieldName('Название').AsString + ''', ' +  
    IntToStr(intMestoUsp + intMestoKach + intMestoProp) + ')';  
  
qryTemp.SQL.Add(t);  
  
qryTemp.ExecSQL;  
  
s := s + '<td><font size="3" face="Times New Roman">' +  
    '<span id="td' + IntToStr(c) + '" name="td' +  
    IntToStr(c) + '"></span>' + '</font></td>';  
  
c := c + 1;  
  
qryMain.Next;  
End;  
  
qryMain.Close;  
  
s := s + '</table></font>';
```

## Продолжение приложения 8

```
dtmMain.qryOcenki5.Parameters.ParamByName('0').Value :=  
dtmMain.tblSem.FieldByName('Номер').AsString;
```

```
dtmMain.qryOcenki5.Parameters.ParamByName('1').Value :=  
dtmMain.tblOtd.FieldByName('Номер').AsString;
```

```
qryOcenki5.Open;
```

```
s := s + '<b><font size="5">Отличников ' +  
IntToStr(qryOcenki5.Recordset.RecordCount) + ' человек:</font></b>';
```

```
s := s + '<table width="100%">';
```

```
qryOcenki5.First;
```

```
i := 0;
```

```
While Not qryOcenki5.EOF Do
```

```
Begin
```

```
  i := i + 1;
```

```
  If i Mod 2 = 0 Then
```

```
    Begin
```

```
      s := s + '<td><i>' +
```

```
        qryOcenki5.FieldByName('fio').AsString + ' — гр. ' +
```

```
        qryOcenki5.FieldByName('nazv').AsString +
```

```
        '</i></td></tr>';
```

```
    End
```

```
  Else
```

Begin

```
s := s + '<tr align="center"><td><i>' +
qryOcenki5.FieldByName('fio').AsString + ' — гр. ' +
qryOcenki5.FieldByName('nazv').AsString +
'</i></td>';
```

End;

```
qryOcenki5.Next;
```

End;

```
qryOcenki5.Close;
```

```
s := s + '</table>';
```

```
//<font face="Monotype Corsiva">
```

```
s := s + '<font face="Monotype Corsiva" size="5">';
```

```
s := s + '<span id="mst" name="mst">';
```

```
{
```

```
s := s + '<p>';
```

```
s := s + '1 <i>место — группа<span id="spn1" name="spn1">';
```

```
s := s + '</i></p>';
```

```
s := s + '<p>';
```

```
s := s + '2 <i>место — группа<span id="spn2" name="spn2">';
```

```
s := s + '</i></p>';
```

```
s := s + '<p>';  
s := s + '3 <i>место — группа<span id="spn3" name="spn3">';  
  s := s + '</i></p>';  
  //}  
  
  //{  
s := s + '</font>';
```