



Государственное бюджетное образовательное учреждение высшего образования
Московской области

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

Колледж космического машиностроения и технологий

КУРСОВОЙ ПРОЕКТ

По МДК.01.02 «Прикладное программирование»

Тема: «Разработка приложения «Учет выполнения работ студентами»

Выполнил студент

Карандашев Н.К.

Группа П1-17

_____ (Подпись)

_____ (Дата сдачи работы)

Проверил преподаватель

Гусятинер Л.Б.

_____ (Оценка)

_____ (Подпись)

Королёв 2020 г.

Оглавление

Введение	3
Глава 1. Теоретическая часть	4
1.1 Изучение предметной области	4
1.2 Изучение существующих разработок	6
Глава 2. Проектная часть	10
2.1 Построение диаграммы прецедентов	10
2.2 Выбор инструментов	12
2.3 Проектирование сценария	14
2.4 Построение диаграммы классов	15
2.5 Описание главного модуля	16
2.6 Описание спецификаций к модулям	19
2.7 Описание модулей	20
2.8 Описание тестовых наборов модулей	25
2.9 Описание применения средств отладки	31
2.10 Анализ оптимальности использования памяти и быстродействия	32
Глава 3. Эксплуатационная часть	34
3.1 Руководство оператора	34
Заключение	45

Введение

Целью данного курсового проекта является написание программы «Учет выполнения работ студентами» преподавателей, учеников. Эта тема является актуальной на данный момент, потому что в наше время все стремятся все автоматизировать или упростить. Данная программа позволит облегчить работу сотрудникам учебного заведения, а именно администрации и преподавателям.

В данной записке будет описана программа и ее применение. Записка разделена на три главы.

В первой главе будет рассмотрена предметная область данной темы, а также приведены примеры уже существующих разработок в данной области, их описание, характеристики и отзывы пользователей.

Во второй главе будут рассмотрены инструменты и модули, которые были разработаны, структура программной части и листинги ключевых частей программных модулей.

В третьей главе будет рассмотрено руководство для пользователей, а также изображение графического интерфейса программы.

В результате этих трех глав будет полностью описана разработка, ее суть и назначения, а также сама область применения. Что полностью опишет всю разработку.

Глава 1. Теоретическая часть

1.1 Изучение предметной области

Образование – неотъемлемая и самая необходимая часть развития любой страны. С каждым годом система образования меняется, а также требует точной записи, что требует большое количество времени у персонала учебного заведения.

В процессе обучения происходит передача ученику/студенту накопленного преподавателем опыта и подготовка его к самостоятельной творческой деятельности в избранной сфере занятий, а также к обязательным знаниям по заданному предмету. От качества образования, существующего в конкретном обществе, во многом зависят темпы его экономического и политического развития, его нравственное состояние.

Так же необходимо разобрать функции образования, которые представляют из себя:

- Экономическая - формирование социально-профессиональной структуры общества, где люди способны осваивать научные и технические новшества и эффективно использовать их в профессиональной деятельности;
- Социальная - социализация личности, воспроизводство социальной структуры общества. Образование - важнейший канал социальной мобильности;
- Культурная - использование ранее накопленной культуры в целях воспитания индивида, развития его творческих способностей;
- Сберегающая - сохранение и передача культурного достояния общества из поколения в поколение.

Также для разработки программы необходимо знать следующие компоненты системы образования:

- 1) органы управления образованием;
- 2) совокупность образовательных стандартов и программ;
- 3) сеть образовательных учреждений;

- дошкольное образование;
- начальное образование;
- среднее образование ;
- среднее профессиональное образование;
- высшая школа.

4) принципы образовательной политики в РФ:

- гуманистический характер образования;
- приоритет общечеловеческих ценностей;
- право личности на свободное развитие;
- единство федерального образования при сохранении права национальных и региональных культур на своеобразие образования;
- общедоступность образования;
- адаптивность системы образования к потребностям обучающихся;
- светский характер образования в государственных учреждениях;
- свобода и плюрализм в образовании;
- демократический, государственно-общественный характер управления и самостоятельность образовательных учреждений.

Современный учебное заведение представляет собой комплекс, в управлении которым особо важна точность, оперативность и удобство, ведь изменения в журнале являются постоянными и ежедневными, постоянно происходят новые занятия, на которые должны происходить соответствующие действия. И если не обеспечить максимальную эффективность учета этих процессов, то работа всей преподавательского состава станет менее результативной.

1.2 Изучение существующих разработок

1.2.1 Образовательный портал

«Образовательный портал» - Учет выполнения работ студентами в Технологическом Университете.

Достоинства данного портала:

1. Универсальна, то есть может применяться как к преподавателям, так и к студентам;
2. Есть возможность написать личное сообщение кому-либо из преподавательского состава или из студентов.

Интерфейс «Образовательного портала»:

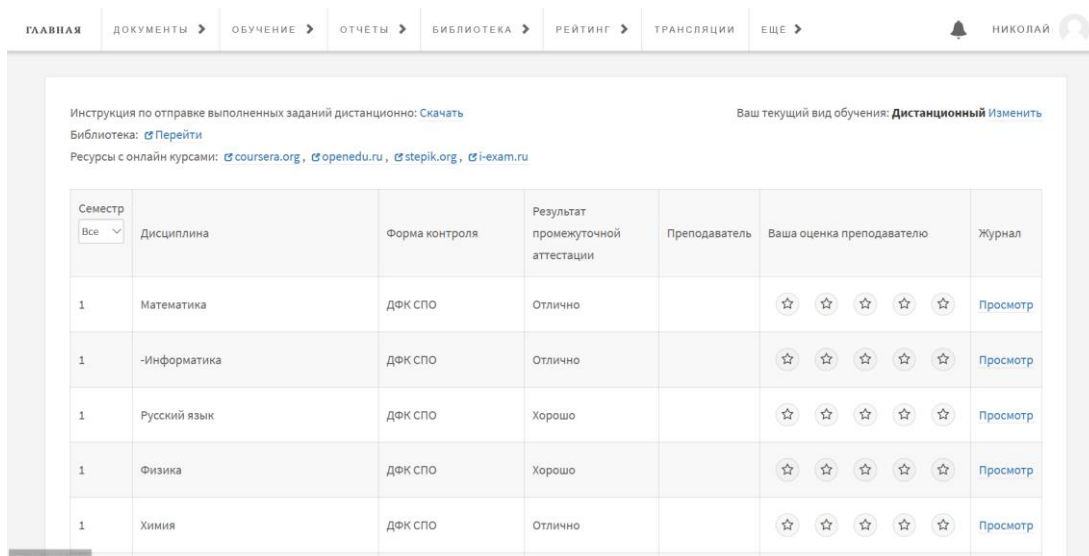


Рисунок 1. Интерфейс «Образовательного портала»

Система «Образовательный портал» затрагивает основные детали обучения.

Ссылка на «учебный портал»: https://ies.unitech-mo.ru/remote_provisionⁱ

1.2.2 Skype

Эта программа является одной из первых разработок такого рода, а следовательно является одной из самых популярных онлайн платформ для видео конференций.

Преимущества Skype представляют собой:

- Среди своих аналогов имеет самый понятный и удобный интерфейс;
- Обеспечивает многоплатформенность;
- Позволяет создавать группы с большим количеством пользователей, с общими сообщениями;
- Позволяет демонстрировать экран;
- Есть функция записи конференции, что записывает демонстрацию экрана и всех звуков в конференции, это функция очень важна во время учебных занятий и лекций.

Но в отличие от своих аналогов Skype имеет значительный минус которой представляет из себя большую затрату ресурсов на поддержку онлайн конференций, что значимо влияет на необходимые технические характеристики пользователя.

Интерфейс Skype:

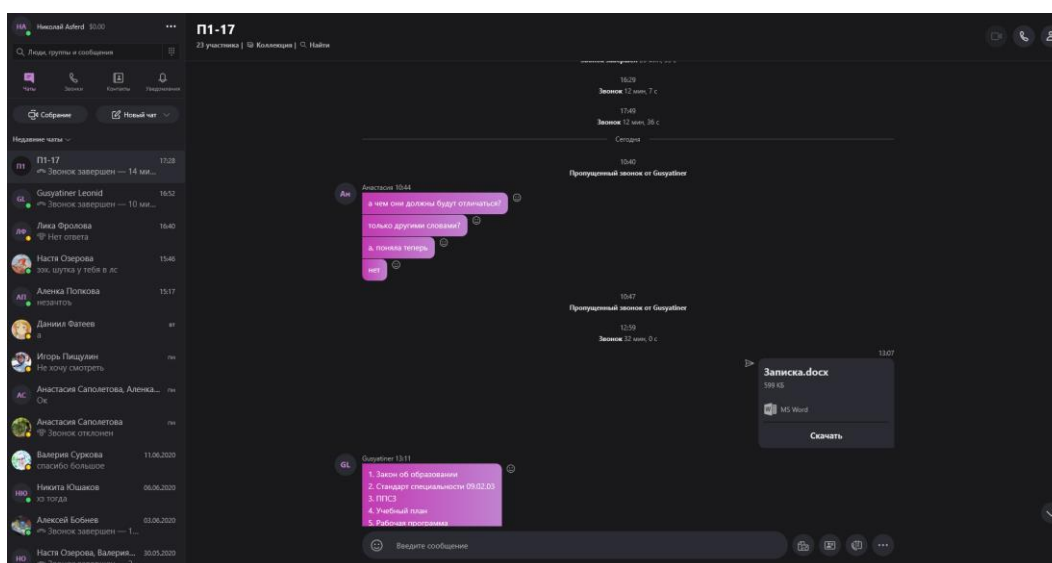


Рисунок 2. Интерфейс Skype

Ссылка на Skype: <https://www.skype.com/ru/> ii

1.2.3 Discord

Эта бесплатная программа создавалась для обмена текстовыми и голосовыми сообщениями - по сути как интернет-телефон. Центральные функции Discord являются свободными и доступными.

Discord имеет преимущества:

- Среди своих аналогов затрачивает меньше всего ресурсов на поддержку звонка;
- Так же среди своих аналогов у Discord самая продвинутая система демонстрации экрана, которая позволяет демонстрировать как сам экран, так и отдельное окно приложения, что обеспечивает безопасность личных данных пользователя;
- Обеспечивает хорошее качество звука и видео;
- Возможность создавать свои сервера, что позволяет создавать аналоги групп и классов, так же в самой группе возможно создать подгруппы, что позволяет обеспечить удобное общение со студентами и преподавателями;
- Современный и понятный интерфейс;
- Единственный среди аналогов не требует загрузки приложения.

Интерфейс Discord:

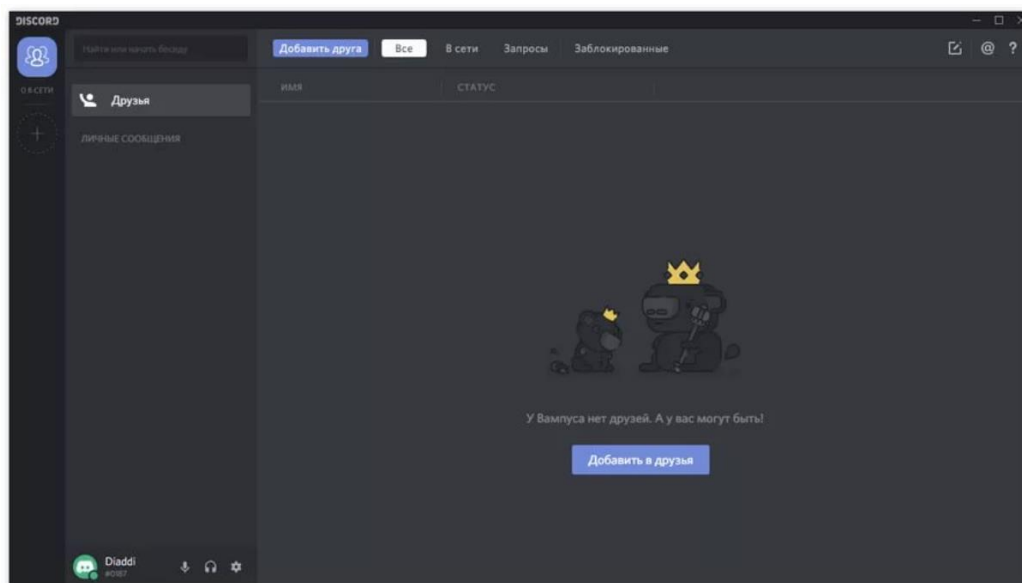


Рисунок 3. Интерфейс Discord

Ссылка на Discord: <https://discord.com/>ⁱⁱⁱ

1.2.4 Zoom

Zoom — сервис для проведения видеоконференций, онлайн-встреч и дистанционного обучения. В рейтинге среди аналогов является самым молодым.

Интерфейс Zoom:

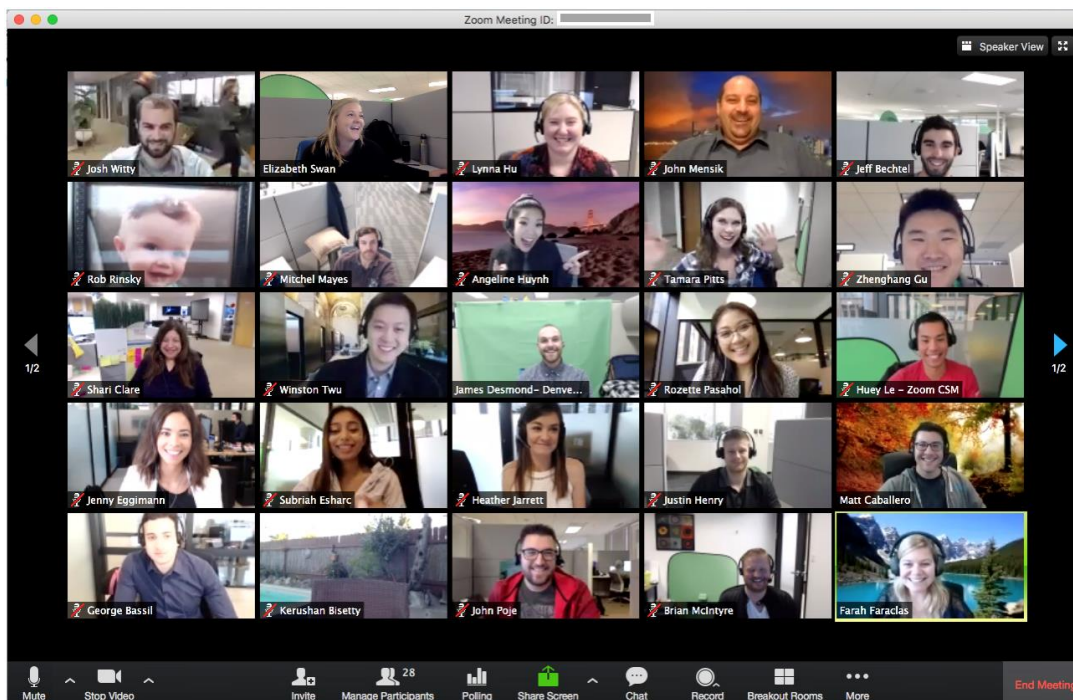


Рисунок 4. Интерфейс Zoom

Преимущества:

- Среди своих аналогов имеет возможность подключать наибольшее количество пользователей к одному звонку;
- Администратор или создатель группы может самостоятельно выключать микрофон, камеру или трансляцию экрана;
- Имеет большой функционал;
- Обеспечивает постоянное и качественное соединение;
- Среди аналогов имеет самую продвинутую систему трансляции, которая заключается в специальных фонах, который каждый пользователь может наложить что бы не показывать ничего кроме себя.

Ссылка на Zoom: <https://zoom-com.ru/?yclid=3412574462916452926iv>

Глава 2. Проектная часть

2.1 Построение диаграммы прецедентов

Были построены диаграммы прецедентов для «Преподавателя», «Студента» и «Заведующего учебной частью».

На данных диаграммах мы можем увидеть какой пользователь имеет какой функционал.

Диаграмма прецедентов «Преподавателя» (рис. 5) показывает, что преподаватель имеет множество выборов действия такие как просмотр студентов, просмотр оценок, автоматическая проверка работ, добавления дня и изменение групп, студентов последние две включают в себя изменения количества значений данной таблице.

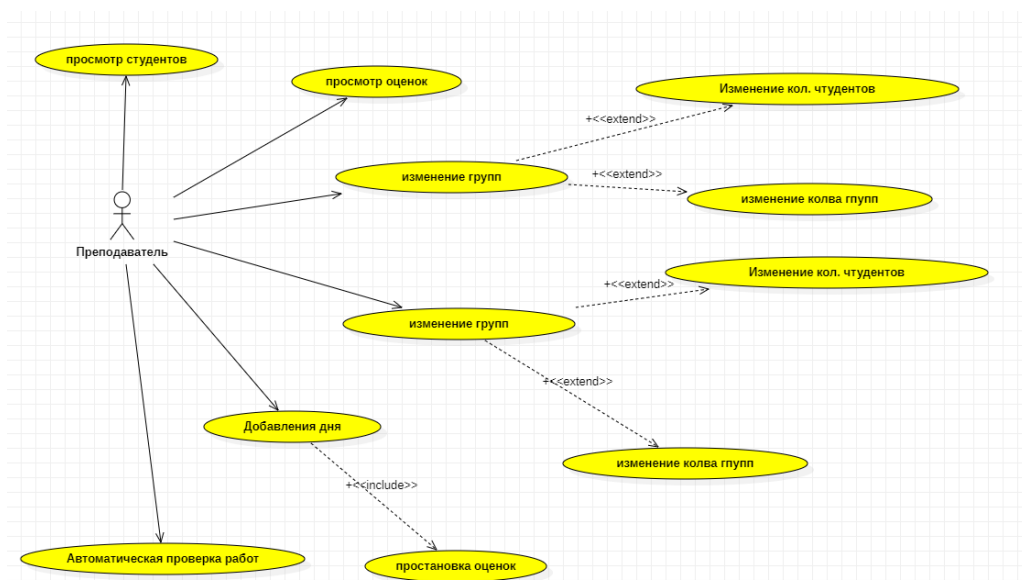


Рисунок 5. Диаграмма прецедентов «Преподавателя»

Диаграмма прецедентов «Студента» (рис. 6). Студент имеет наименьший функционал в него входит только просмотр своих оценок и просмотр расписания.

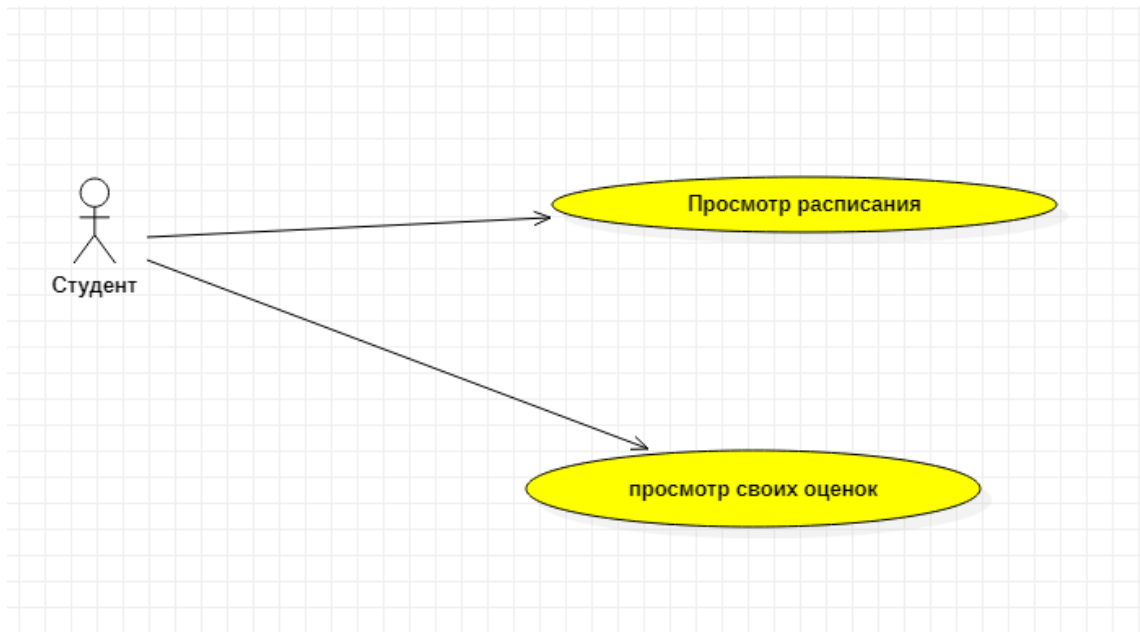


Рисунок 6. Диаграмма прецедентов «Студента»

Диаграмма прецедентов «Зав. учебной частью» (рис. 7) показывает, что заведующий учебной частью имеет право изменять количество студентов, изменять и просматривать расписание, изменять группы что включает в себя количество групп и их просмотр.

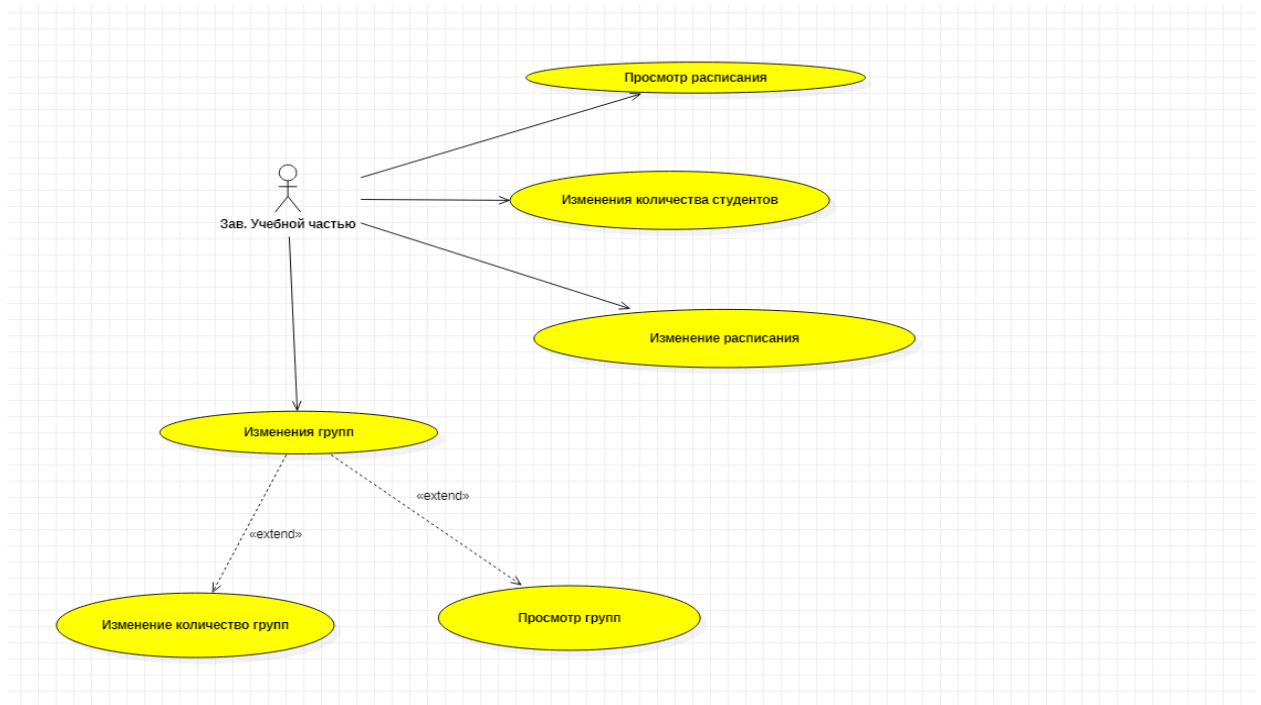


Рисунок 7. Диаграмма прецедентов «Зав. учебной частью»

2.2 Выбор инструментов

При выборе инструментов я исходил из некоторых критериев, который я считал самыми главными в разработке.

Важность критерия я выбирал из: низкая, средняя, высокая.

Таблица №1 важность критерия:

Критерий	Участие в корпоративном проекте	Простота сопровождения	Наличие библиотек связанных с выбранной Базой данных	Наличие документации на русском языке	Скорость разработки
Важность критерия	Низкая	Средняя	Высокая	Низкая	Высокая
Критерий	Простота разработки графического интерфейса	Скорость обучения			
Важность критерия	Высокая	Высокая			

Исходя из этих критериев, я сравнил 3 языка программирования от 0 до 10 баллов за критерий исходя из своих предпочтений и знаний.

Таблица №2 сравнение языков программирования по критериям:

Критерий/Язык программирования	C++	C#	Object Pascal
Участие в корпоративном проекте	7	10	7
Простота сопровождения	7	6	8
Наличие библиотек	3	4	10
Наличие документации на русском языке	8	7	8
Скорость разработки	8	6	8
Простота разработки графического интерфейса	7	8	10
Скорость обучения	10	5	9
Итого баллов	50	46	60

По результатам сравнения мной был выбран язык программирования Object Pascal.

2.3 Проектирование сценария

Данная программа ориентирована на такой сценарий (рис. 8).

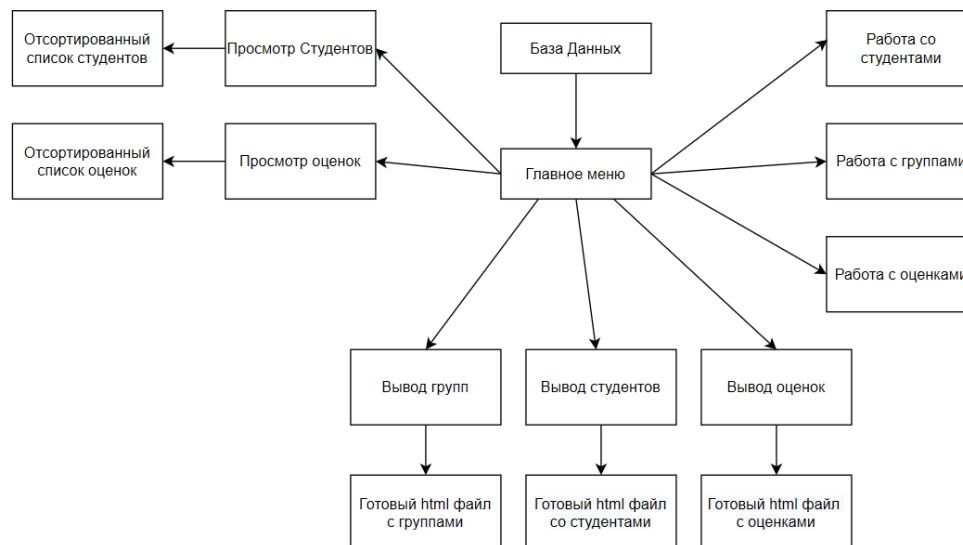


Рисунок 8. Сценарий программы

После запуска программы пользователь имеет несколько вариантов действий:

1. Просмотр студентов;
2. Просмотр оценок;
3. Работа со студентами;
4. Работа с оценками;
5. Работа с группами;
6. Вывод групп;
7. Вывод студентов;
8. Вывод оценок.

При выборе вывода любой таблицы пользователь будет получать готовый html файл с готовыми к просмотру таблицами.

При выборе просмотра любой таблицы пользователь будет получать готовую к просмотру соответствующую таблицу.

2.4 Построение диаграммы классов

В данном параграфе представлены все классы, которые используются в программе, а также их взаимосвязи (рис. 9).

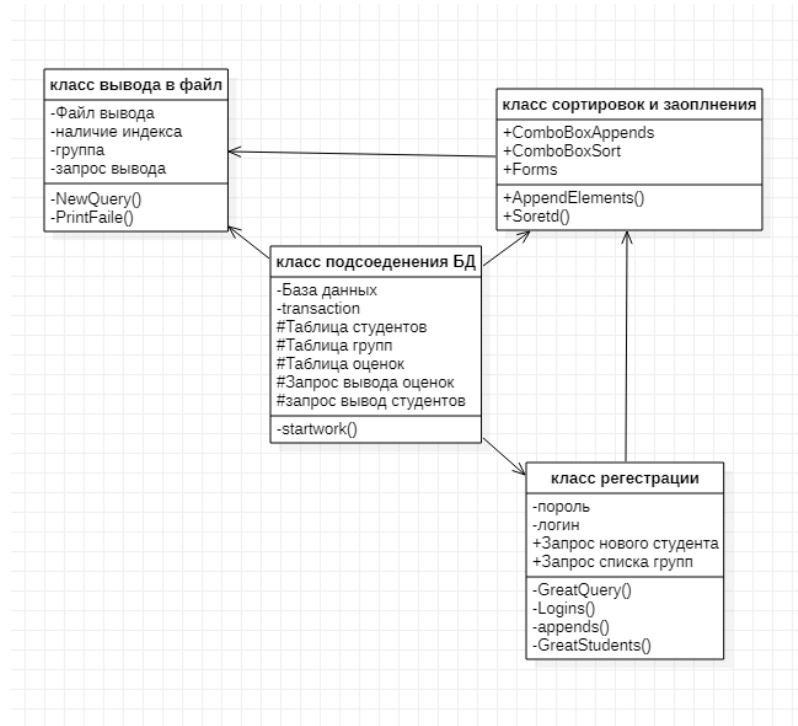


Рисунок 9. Диаграмма классов

2.5 Описание главного модуля

В главный модуль разработки входит класс подсоединение базы данных, который отвечает за взаимосвязь со всеми компонентами базы данных, а также за их включение.

Данный модуль запускается автоматически при запуске программы что обеспечивает качественное соединение и отсутствие ошибок во время работы.

Данный модуль состоит из одного кода, который приведен в листинге №1.

Листинг № 1:

```
unit maindata;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, IBDatabase, IBTable, IBCustomDataSet, IBQuery,
  db, StdCtrls, Dialogs;
type
  //список элементов формы
  TMainDatas = class(TDataModule)
    OpenDialog1: TOpenDialog;
    QDatDATAS: TIBStringField;
    QDstudIDGROUPS: TIBIntegerField;
    QDstudIDSTUDENTS: TIBIntegerField;
    QDstudNAMEGROUPS: TIBStringField;
    QDstudSECONDNAME: TIBStringField;
    QueryOcDATAS: TIBStringField;
    QueryOcHAVEWORK: TIBStringField;
    QueryOcIDOC: TIBIntegerField;
    QueryOcNAMEGROUPS: TIBStringField;
    QueryOcOCENKA: TIBStringField;
    QueryOcSECONDNAME: TIBStringField;
    SqDat: TDataSource;
    SQstud: TDataSource;
    QDat: TIBQuery;
    QDstud: TIBQuery;
    SourceOc: TDataSource;
    QueryOc: TIBQuery;
    Workstd: IBTable;
    SourceWorkStd: TDataSource;
    LookOcenkiDATAS: TIBStringField;
    LookOcenkiHAVEWORK: TIBStringField;
    LookOcenkiIDOC: TIBIntegerField;
    LookOcenkiNAMEGROUPS: TIBStringField;
    LookOcenkiSECONDNAME: TIBStringField;
    LookOcenkiVALUE: TIBIntegerField;
    SourceNameGR: TDataSource;
    QuarryNameGR: TIBQuery;
    MainStudentFATHERNAME: TIBStringField;
    MainStudentIDGROUPS: TIBIntegerField;
    MainStudentIDSTUDENTS: TIBIntegerField;
    MainStudentNAME: TIBStringField;
```



```

MainStudentNAMEINGIT: TIBStringField;
MainStudentSECONDNAME: TIBStringField;
QueryStudentsFATHERNAME: TIBStringField;
QueryStudentsIDSTUDENTS: TIBIntegerField;
QueryStudentsNAME: TIBStringField;
QueryStudentsNAMEGROUPS: TIBStringField;
QueryStudentsNAMEINGIT: TIBStringField;
QueryStudentsSECONDNAME: TIBStringField;
SourceStudent: TDataSource;
MainStudent: TIBTable;
SourceQSt: TDataSource;
QueryStudents: TIBQuery;
MainGroupsIDGROUPS: TIBIntegerField;
MainGroupsNAMEGROUPS: TIBStringField;
SourceGroupe: TDataSource;
MainGroups: TIBTable;
IdTrans: TIBTransaction;
IDData: TIBDatabase;
WorkstdDATAS: TIBStringField;
WorkstdHAVEWORK: TIBStringField;
WorkstdIDOC: TIBIntegerField;
WorkstdIDSTUD: TIBIntegerField;
WorkstdOCENKA: TIBStringField;
WorkWithOcDATAS: TIBStringField;
WorkWithOcHAVEWORK: TIBStringField;
WorkWithOcIDGROUPS: TIBIntegerField;
WorkWithOcIDOC: TIBIntegerField;
WorkWithOcIDSTUDENTS: TIBIntegerField;
WorkWithOcVALUE: TIBStringField;
procedure DataModuleCreate(Sender: TObject);
private
public
end;
var
    MainDatas: TMainDatas;
Implementation uses WorkStudentsForms;
{$R *.lfm}
{ TMainDatas }
procedure TMainDatas.DataModuleCreate(Sender: TObject);
begin //поочередно включаем все необходимые нам компоненты
if (opendialog1.Execute) then begin
    IDData.DatabaseName:='127.0.0.1:' +
ExtractFileDir (OpenDialog1.FileName)+'\MYSTUDENTS.FDB';
//записываем путь в дату
    IDData.Connected:=true; //подключаем
    IdTrans.StartTransaction; //начинаем транзакцию
    MainGroups.open; //далее открываем все запросы и таблицы
    MainStudent.Open;
    workstd.open;
end;
end;
end.

```

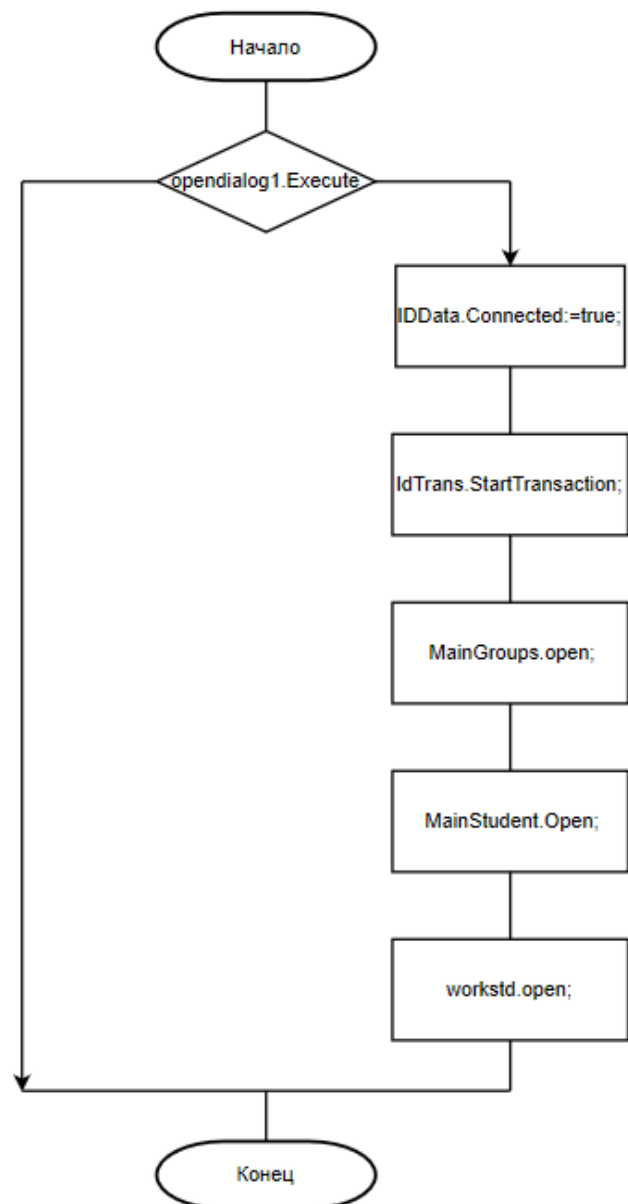


Рисунок 10. Блок-схема главного модуля

2.6 Описание спецификаций к модулям

В данной главе будут описаны публичные члены модулей курсового проекта.

В главном модуле публичным объектом является MainDatas который представляет из себя главную форму модуля, на который закреплены объекты, предназначенные для взаимосвязи с базой данных.

В этом объекте так же присутствует процедура DataModuleCreate, которая подсоединяет базу данных к программе.

В модуле отвечающим за заполнение выпадающих списков, и сортировку соответствующих таблиц так же присутствуют публичные методы, которые будут описаны в следующем параграфе здесь они будут только перечислены. Так же в модуле имеется публичный объект LookinOC, который включает в себя все компоненты формы.

Публичные процедуры:

- FormCreate – заполняет выпадающий список необходимыми элементами из базы данных;
- ComboBox1Change – сортирует таблицу по выбранному элементу из ComboBox, который является частью класса LookinOc.

2.7 Описание модулей

В данной главе мы разберем три главных модуля:

- Центральный модуль;

Данный модуль представляет из себя набор обязательных элементов базы данных таких как таблицы, запросы и транзакции. И центральную функцию, которая представляет из себя запуск и настройку базы данных.

- Модуль работы с базами данных;

Модуль представляет из себя несколько графических форм, каждая из которых обращается в центральный модуль за своей таблицей из базы данных для дальнейшей работы с ней и подключает свою навигационную систему этой таблице что обеспечивает удобную работу. Также этот модуль для удобства ввода имеет скрипт для подстановки соответствующих значений в выпадающие списки.

- Модуль просмотра таблиц.

Модуль представляет из себя несколько графических форм каждая из которых обращается в центральный модуль за своим запросом из базы данных для дальнейшей его вывода, так же подключает навигацию к полученному запросу. Так же этот модуль имеет два скрипта которые представляют из себя:

1. Подстановку в выпадающий список соответствующее значения;
2. При выборе значения из списка производит изменение скрипта запроса и выводит новую отсортированную таблицу.

Листинг №2:

```

unit LookOcenki;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, DBGrids,
  StdCtrls;
type
  { TLookinOc }
  TLookinOc = class(TForm)
    ComboBox1: TComboBox;
    Label1: TLabel;
    lookoc: TDBGrid;
    procedure ComboBox1Change(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
  public
  end;
var
  LookinOc: TLookinOc;
implementation
  uses maindata ;
  {$R *.lfm}
  { TLookinOc }
  procedure TLookinOc.FormCreate(Sender: TObject);
  begin
    begin
      maindata.MainDatas.Qdat.close;
      maindata.MainDatas.Qdat.open;
      maindata.MainDatas.QueryOc.close;
      maindata.MainDatas.QueryOc.open;
      while not(maindata.MainDatas.Qdat.eof) do //проходим по таблице с
//датами
        begin
          ComboBox1.Items.Add(maindata.MainDatas.Qdat.FieldName('datas').AsString); //добавляем в комбобокс соответствующее значение из запроса
            maindata.MainDatas.Qdat.next; //сдвиг указателя
        end;
    end;
  end;
  procedure TLookinOc.ComboBox1Change(Sender: TObject);
  begin
    maindata.MainDatas.QueryOc.close; //закрываем таблицу
    if not (ComboBox1.Text = 'Все') then //проверка на включение всех
      maindata.MainDatas.QueryOc.SQL[3]:=' and ocenki.datas='+ chr(39)
+ComboBox1.Text+ chr(39) //добавляем соответствующий кусок кода
    else
      maindata.MainDatas.QueryOc.SQL[3]:=''; //иначе удаляем
      maindata.MainDatas.QueryOc.open;
    end;
  end;
end.

```

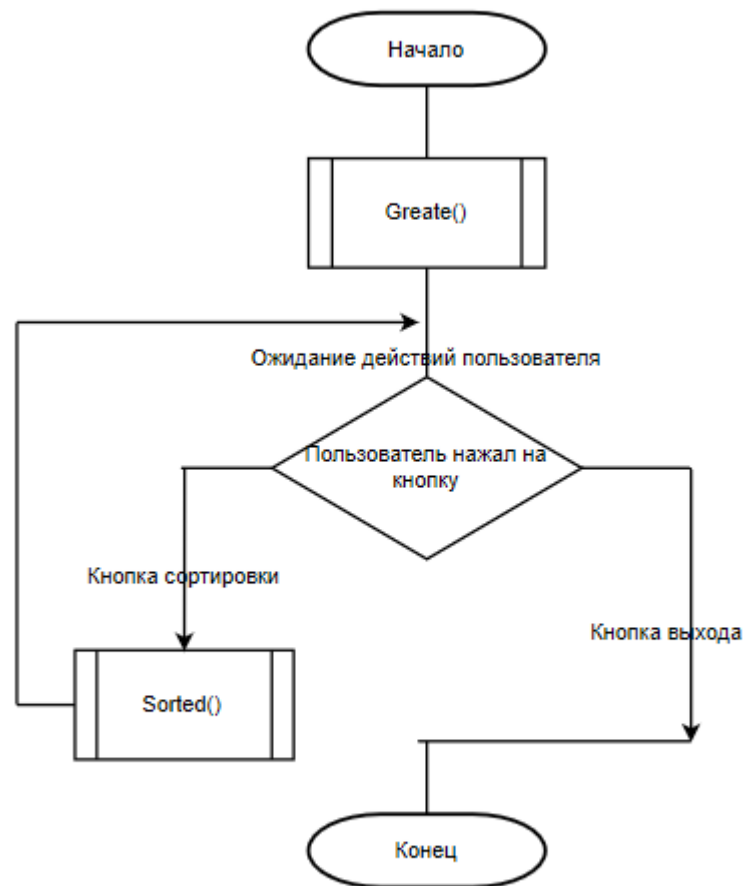


Рисунок 11. Блок-схема главного модуля

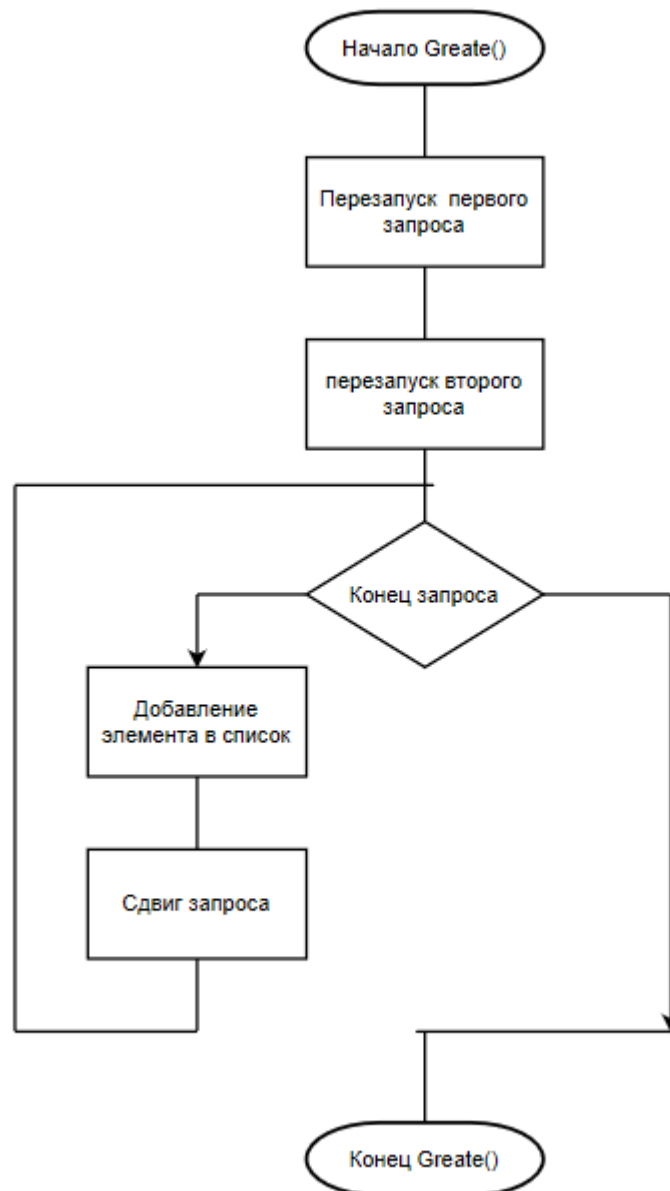


Рисунок 12. Блок-схема функции `Greate()`

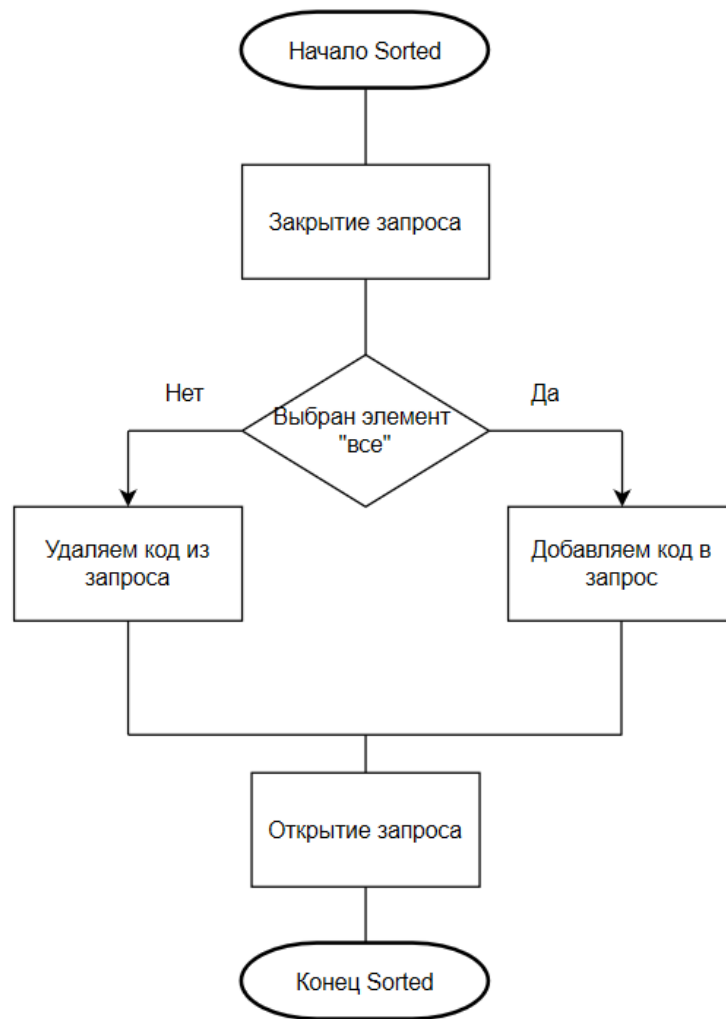


Рисунок 13. Блок-схема функции Sorted()

2.8 Описание тестовых наборов модулей

В этом параграфе будут продемонстрированы результаты тестирования методом «Черного ящика».

Тест 1. Переход в окно «Работа с оценками».

Действия: нажать на кнопку «Работа с оценками».

Ожидаемый результат: переход в окно «Работа с оценками».

Результат теста:

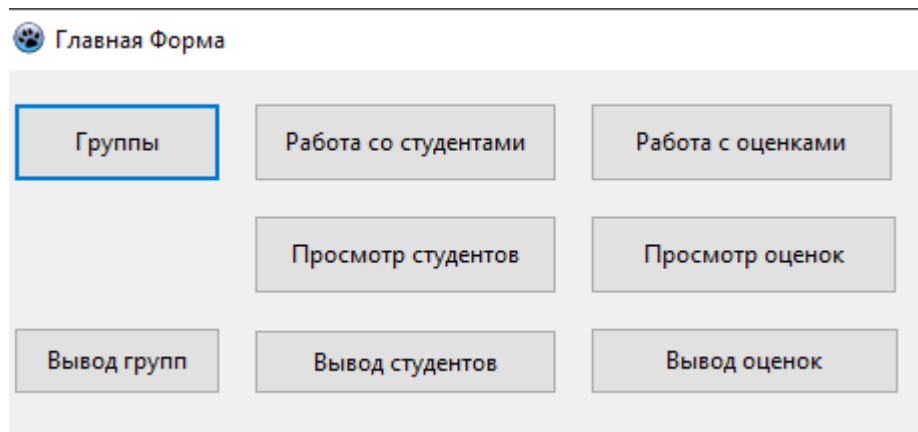


Рисунок 14. Главное меню

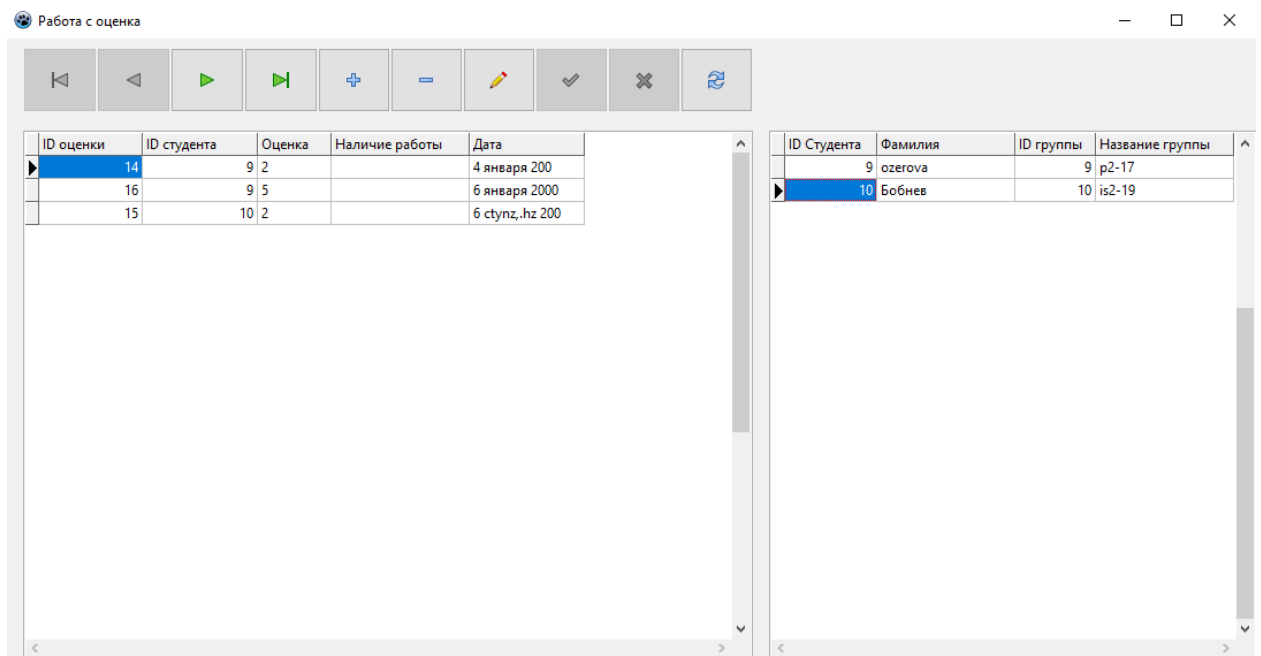


Рисунок 15. Работа с оценками

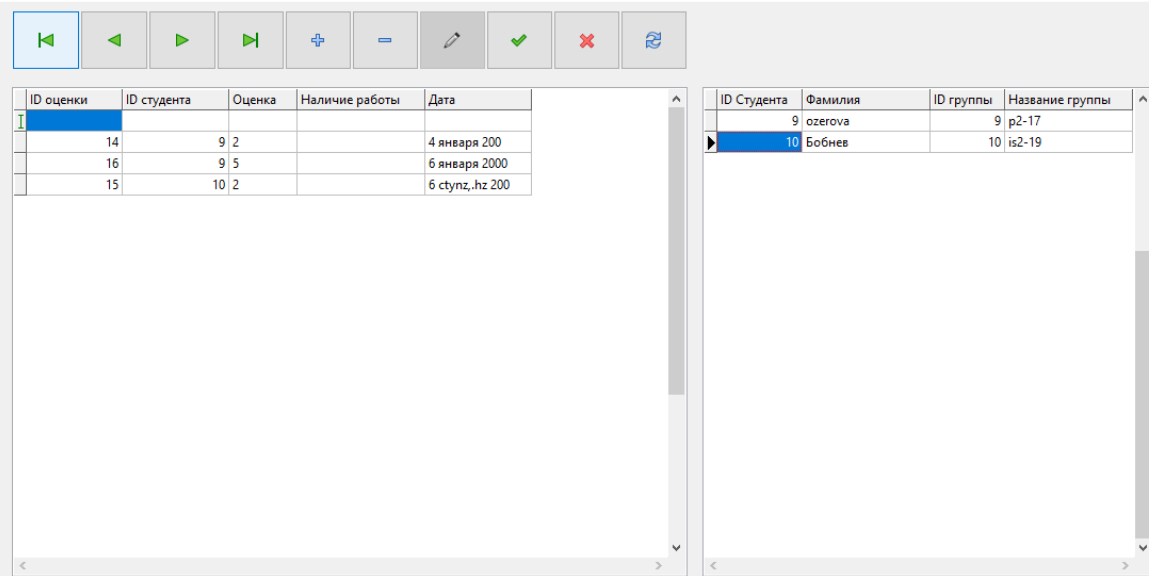
Тест 2. Добавления позиций в таблицу.

Действия: нажать на кнопку добавления на панели навигации, в появившейся новой строке таблицы ввести необходимую информацию и нажать на кнопку обновление.

Ожидаемый результат: Добавление строки в таблицу, добавление в нее информации.

Результат теста:

Работа с оценкой

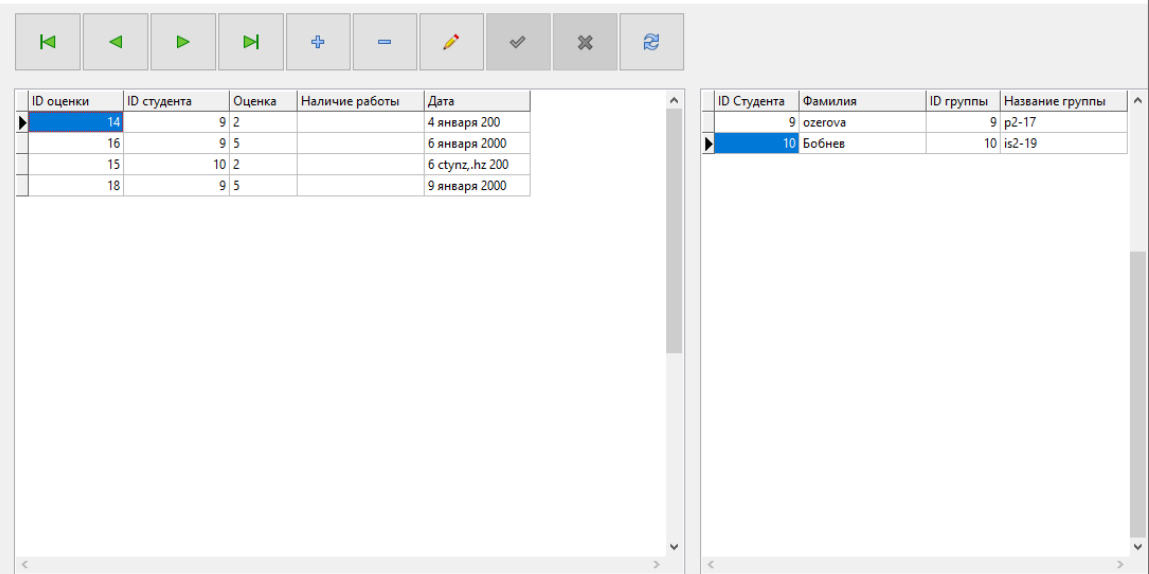


ID оценки	ID студента	Оценка	Наличие работы	Дата
14	9	2		4 января 200
16	9	5		6 января 2000
15	10	2		6 стынз, hz 200

ID Студента	Фамилия	ID группы	Название группы
9	ozegova	9	p2-17
10	Бобнев	10	is2-19

Рисунок 16. Начало ввода

Работа с оценкой



ID оценки	ID студента	Оценка	Наличие работы	Дата
14	9	2		4 января 200
16	9	5		6 января 2000
15	10	2		6 стынз, hz 200
18	9	5		9 января 2000

ID Студента	Фамилия	ID группы	Название группы
9	ozegova	9	p2-17
10	Бобнев	10	is2-19

Рисунок 17. Полученная новая таблица

Тест 3. Удаление позиции из таблицы.

Действия: через панель навигации выбрать вторую строку, нажать на кнопку удаления в панели навигации.

Ожидаемый результат: удаление второй строки.

Результат теста:

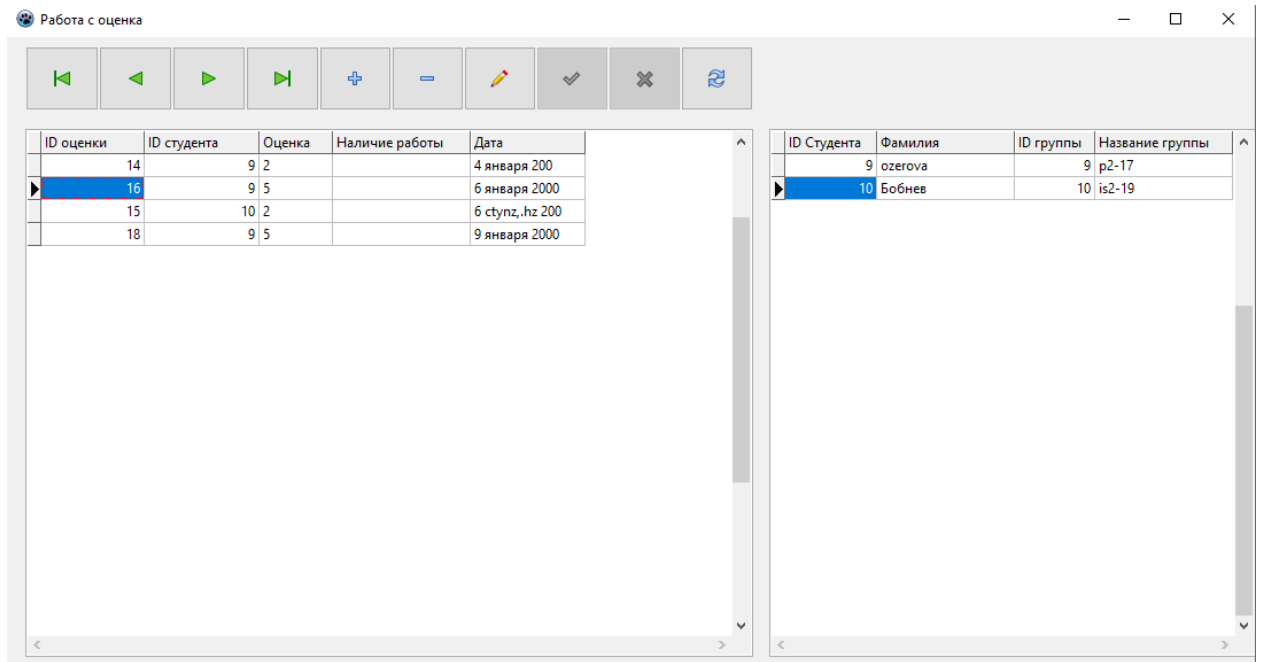


Рисунок 18. Постановка на 2 строку

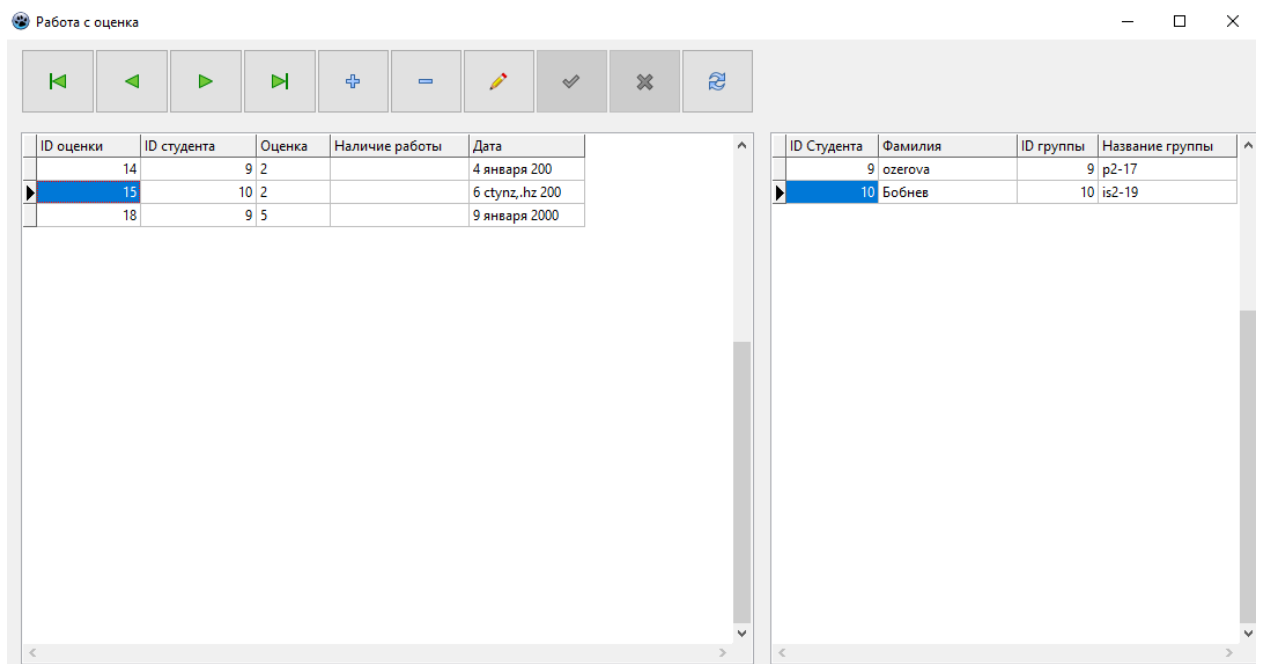


Рисунок 19. Новая таблица после удаления

Тест 4: Выход в главное меню.

Действия: нажать на кнопку закрытие окна.

Ожидаемый результат: выход в главное меню.

Результат теста:

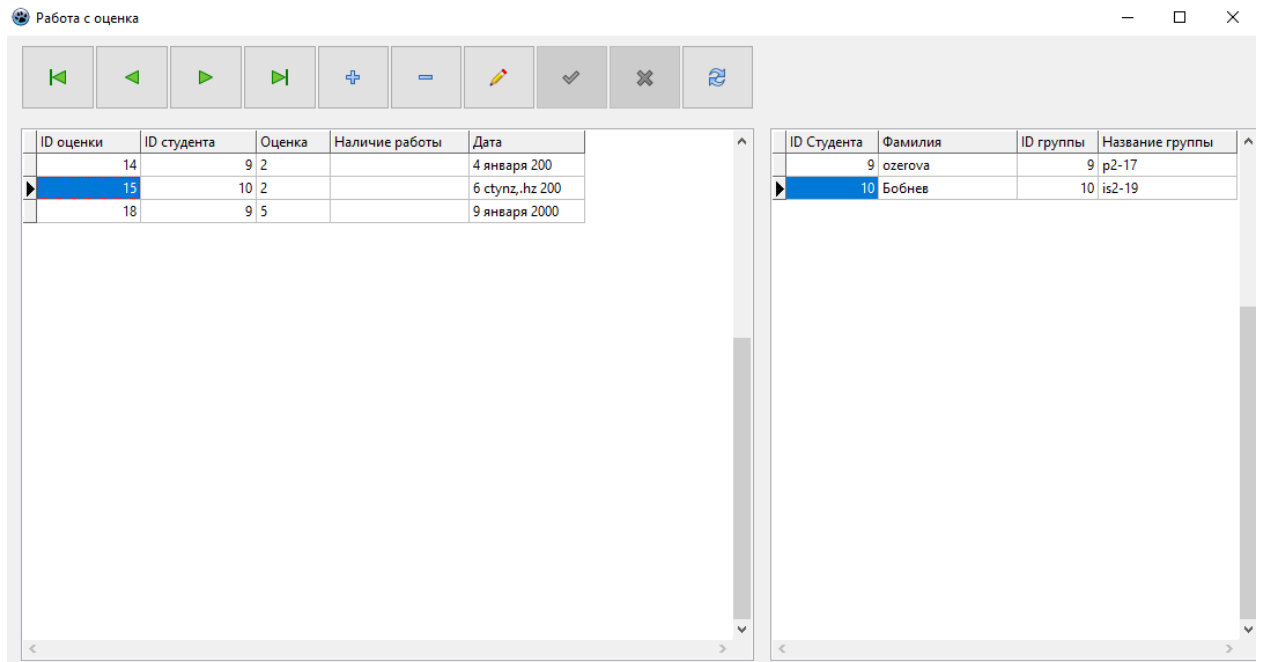


Рисунок 20. Окно "Работа с оценками"

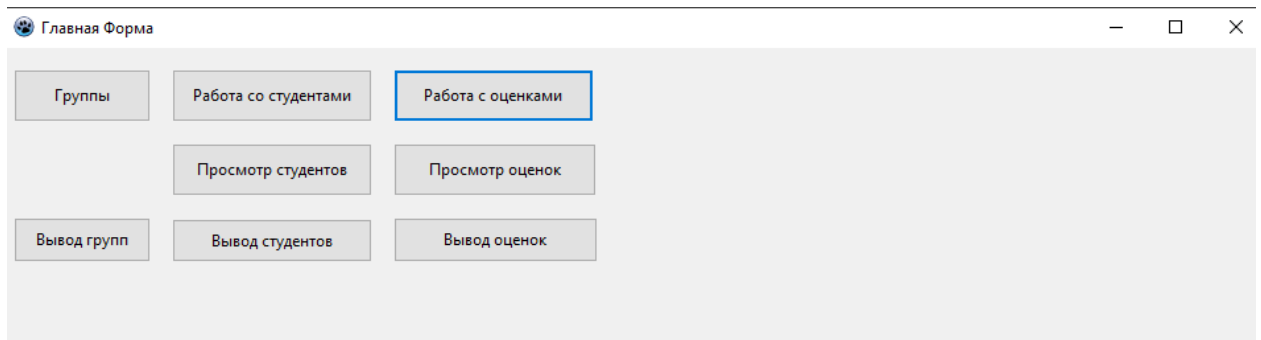


Рисунок 21. Главное меню

Тест 5: Переход в окно «Просмотр студентов».

Действия: нажать на кнопку «Просмотр студентов».

Ожидаемый результат: открытие окна «Просмотр студентов».

Результат теста:

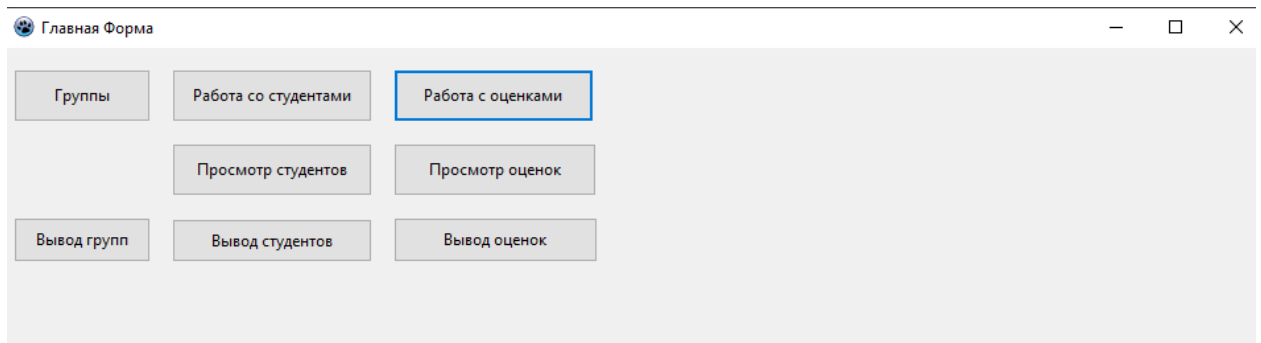


Рисунок 22. Главное меню

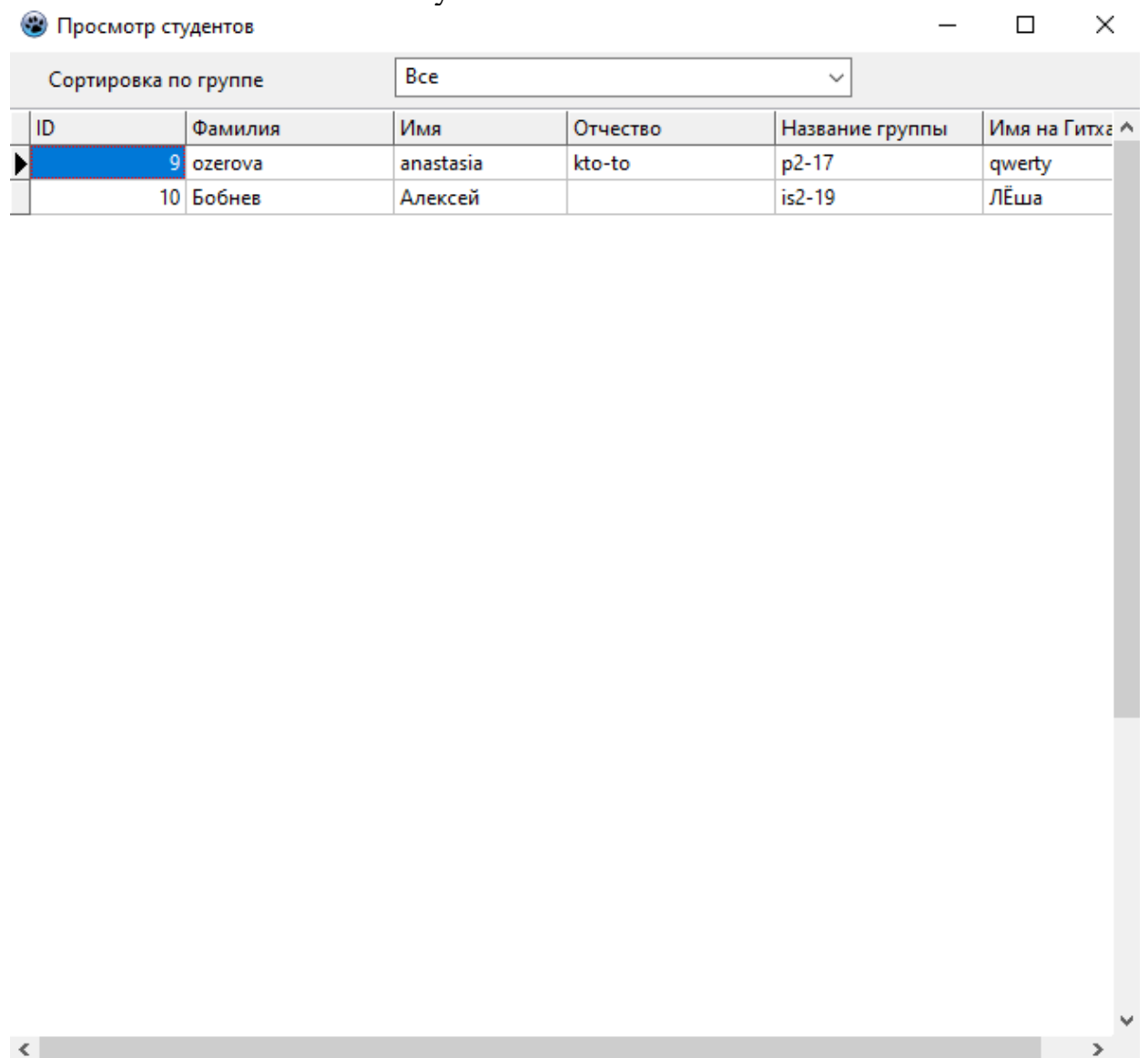


Рисунок 23. Таблица "Просмотр студентов"

Тест 6: Сортировка по группе.

Действия: нажать на выпадающий список, выбрать p2-17.

Ожидаемый результат: получение таблицы, состоящей из студентов из группы p2-17.

Результат теста:

Просмотр студентов

Сортировка по группе: Все

ID	Фамилия	Имя	Отчество	Название группы	Имя на Гитхабе
9	ozeroва	anastasia	kto-to	p2-17	qwerty
10	Бобнев	Алексей		is2-19	Лёша

Рисунок 24. Таблица "Просмотр студентов"

Просмотр студентов

Сортировка по группе: p2-17

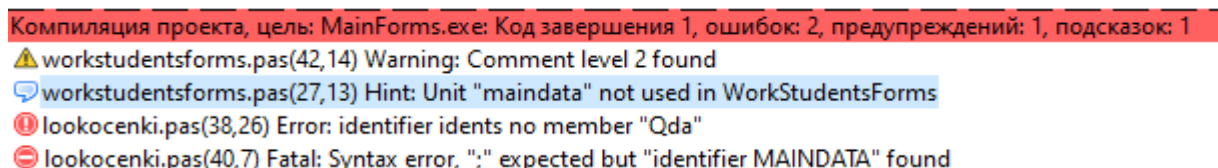
ID	Фамилия	Имя	Отчество	Название группы	Имя на Гитхабе
9	ozeroва	anastasia	kto-to	p2-17	qwerty

Рисунок 25. Сортировка

2.9 Описание применения средств отладки

В этом параграфе показано умение использовать средства отладки. В ходе написания курсового проекта при попытке запустить программу были получены следующие ошибки.

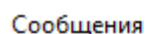
До отладки программы были ошибки с неизвестными именами.



```
Компиляция проекта, цель: MainForm.exe: Код завершения 1, ошибок: 2, предупреждений: 1, подсказок: 1
⚠ workstudentsforms.pas(42,14) Warning: Comment level 2 found
💬 workstudentsforms.pas(27,13) Hint: Unit "maindata" not used in WorkStudentsForms
❗ lookocenki.pas(38,26) Error: identifieridents no member "Qda"
❗ lookocenki.pas(40,7) Fatal: Syntax error, ";" expected but "identifier MAINDATA" found
```

Рисунок 26. Сообщение об ошибках до исправления

После проверки скрипта lookocenki а точнее 38 и 40 строк. Были исправлены найденные ошибки, в результате чего при следующем запуске программы не было получена ни одна ошибка.



Сообщения

Рисунок 27. Сообщение об ошибках после исправления

2.10 Анализ оптимальности использования памяти и быстродействия

В данном разделе будет проведен анализ оптимальности использования памяти и быстродействия программы.

Список принятых оптимальных решений:

1. Было принято решение об использовании многотабличных запросов по индексам, что позволило сократить время поиска так как был использован двоичный или бинарный поиск, так же это значимо сыграло на объеме памяти, так как вместо нескольких весомых палей использовалось только одно поле индивидуального номера;

2. Размер файла MainForms.exe достаточно большой. Было решено оптимизировать компиляцию, тем самым уменьшив размер файла.



Рисунок 28. Размер файла до оптимизации

Для уменьшения размера файла необходимо отключить «Генерировать отладочную информацию» и включить «Вырезать символы из исполняемого файла».

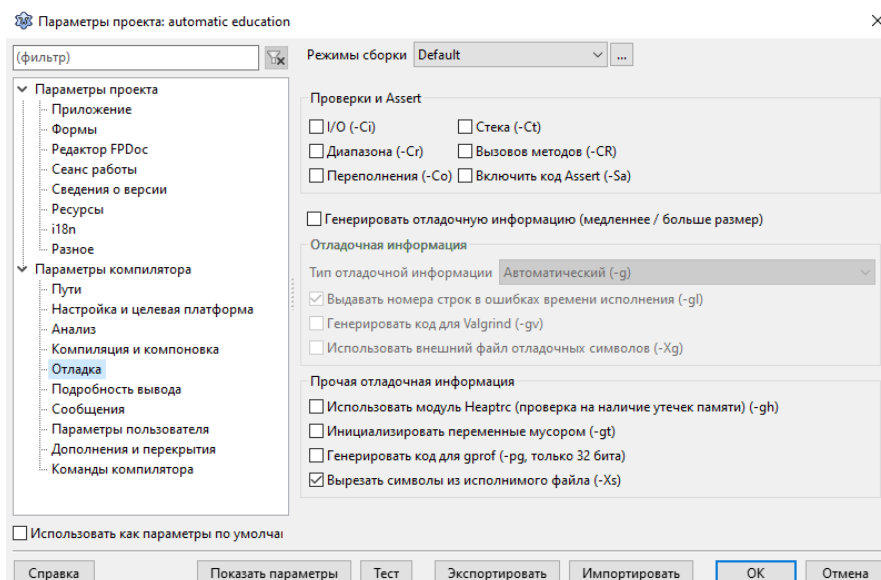


Рисунок 29. Вкладка "Отладка"

Так же необходимо включить «Умная компоновка» в разделе «Стиль модуля» и «Компоновка».

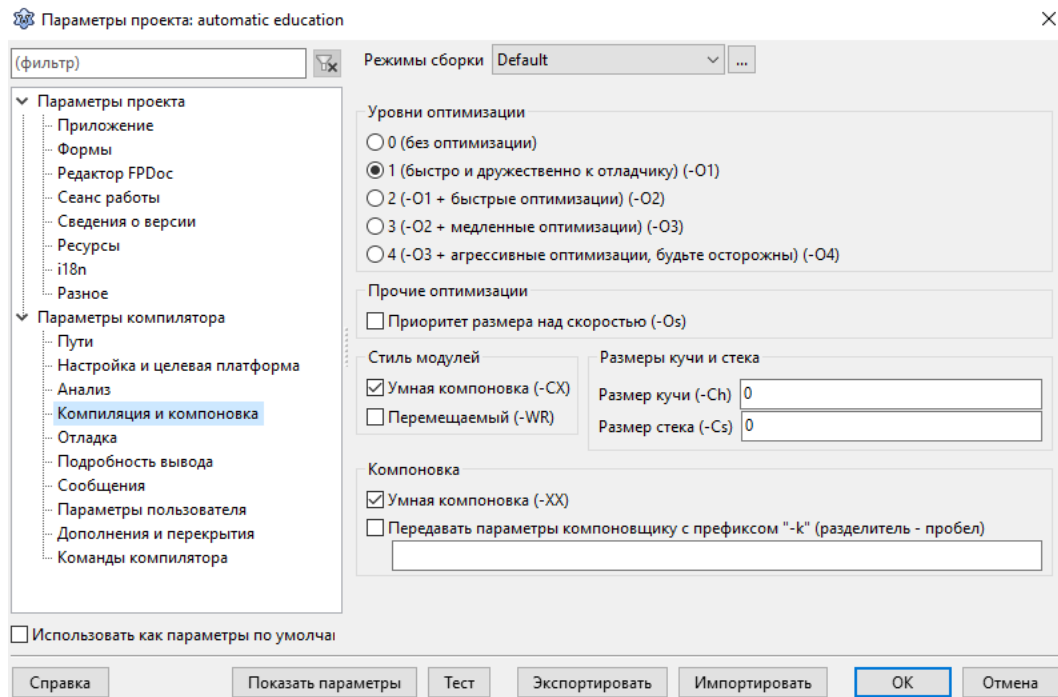


Рисунок 30. Вкладка "Компиляция и компоновка"

После проведенной оптимизации компиляции можно заметить существенную разницу в размере файла. До оптимизации размер exe-файла составлял 31468 КБ, а после 4347 КБ.

	MainForms	21.06.2020 16:30	Приложение	4 347 КБ
--	-----------	------------------	------------	----------

Рисунок 31. Размер файла после оптимизации

Глава 3. Эксплуатационная часть

3.1 Руководство оператора

Аннотация

В данном документе приведено руководство оператора по применению и эксплуатации программы «automatic education», предназначенной для облегчения работы преподавателей и помощи студентам.

В данном программном документе в разделе «назначение программы» будут указаны сведения о назначении программы и информация, необходимая для понимания функциональности программы и ее эксплуатации.

В разделе «Условия выполнения программы» указаны условия, необходимые для выполнения программы (минимальный состав аппаратных и программных средств и т.п.).

В разделе «Выполнение программы» указана необходимая последовательность действий оператора, обеспечивающая загрузку, запуск, выбор базы данных, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузку и управляет выполнением программы, а также ответы программы на данные команды.

В разделе «Сообщение оператору» приведены тексты сообщений, выдаваемых в ходе выполнения программы, описание их содержания и соответствующие действия оператора (действия оператора в случае сбоя, возможности повторного, запуска программы и т.п.).

Оформление программного документа «Руководство оператора» произведено по требованиям ЕСПД (ГОСТ 19.101-77 ¹⁾, ГОСТ 19.103-77 ²⁾,

¹⁾ ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов

²⁾ ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов

ГОСТ 19.104-78* ³⁾, ГОСТ 19.105-78* ⁴⁾, ГОСТ 19.106-78* ⁵⁾, ГОСТ 19.505-79* ⁶⁾, ГОСТ 19.604-78* ⁷⁾).

1 Назначение программы

1.1. Функциональное назначение программы

Специальное программное обеспечение «automatic education» используется для отслеживания работ студентов, а также для автоматической проверки выполнения задания.

1.2. Эксплуатационное назначение программы

Специальное программное обеспечение «automatic education» может эксплуатироваться на объектах любого масштаба в сфере образования для облегчения работы преподавателей.

1.3. Состав функций

1.3.1. Функция открытия окна таблиц

Эта функция позволяет перейти из главного меню в окно работы с таблицей.

1.3.2. Функция добавления элемента в таблицу

Эта функция позволяет добавлять нужную информацию в таблицу.

1.3.3. Функция удаления элемента из таблицы

Эта функция позволяет удалить выбранный элемент в таблице.

1.3.4. Функция возвращения в главное меню

Эта функция позволяет вернуться в главное меню.

1.3.5. Функция открытия окна просмотра

Эта функция позволяет перейти из главного меню в окно просмотра таблицы.

1.3.6. Функция сортировки

³⁾ ГОСТ 19.104-78* ЕСПД. Основные надписи

⁴⁾ ГОСТ 19.105-78* ЕСПД. Общие требования к программным документам

⁵⁾ ГОСТ 19.106-78* ЕСПД. Общие требования к программным документам, выполненным печатным способом

⁶⁾ ГОСТ 19.505-79* ЕСПД. Руководство оператора. Требования к содержанию и оформлению

⁷⁾ ГОСТ 19.604-78* ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом

Эта функция позволяет отсортировать элементы таблицы по выбранному критерию.

2. Условия выполнения программы

2.1. Минимальные системные требования

ОС: Windows Vista/7/8/10

Процессор: 2.0 GHz Dual Core Processor

Оперативная память: 1 GB ОЗУ

Видеокарта: GeForce 8800 или эквивалентная

DirectX: Версии 9.0с

Место на диске: 750 MB

2.2. Требование к персоналу (пользователю)

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

3. Выполнение программы

3.1. Загрузка и запуск программы

Запустите программу «automatic education», откроется окно главного меню:

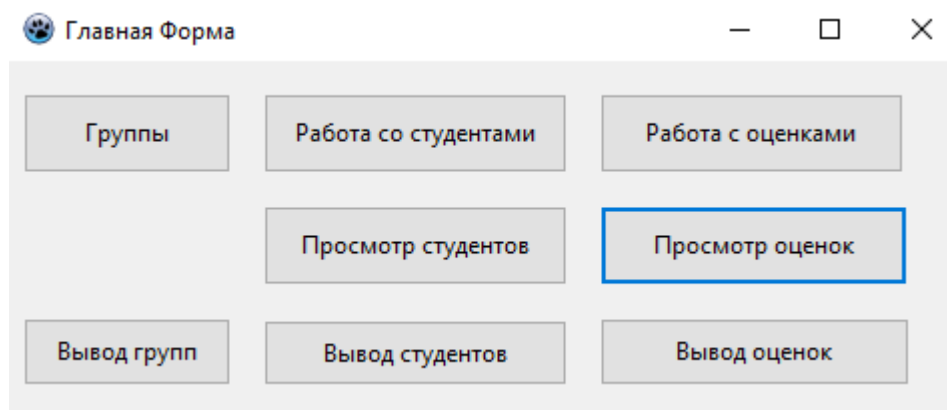


Рисунок 32. Главное окно

Кнопки меню имеют следующие функции:

- «Группы», «Работа со студентами», «Работа с оценкам» – открывают соответствующие таблицы для работы с ними;
- «Просмотр оценок», «Просмотр групп» – открывают соответствующие таблицы для просмотра;
- «Вывод групп», «Вывод студентов», «Вывод оценок» – обрабатывает таблицы и выводит их в файл html.

3.2. Выполнение программ

3.2.1. Выполнение функции открытия окна таблиц.

Выберите пункт меню «Работа с таблицей», в данном случае «Работа со студентами», после этого откроется окно с таблицей.

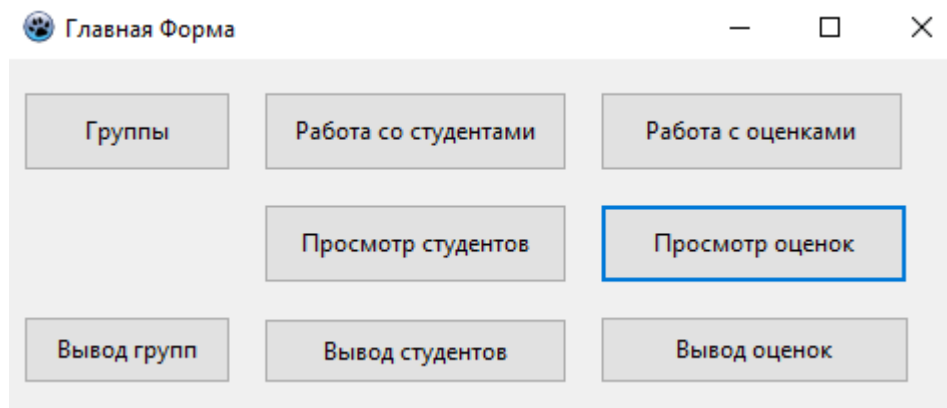


Рисунок 33. Главное окно

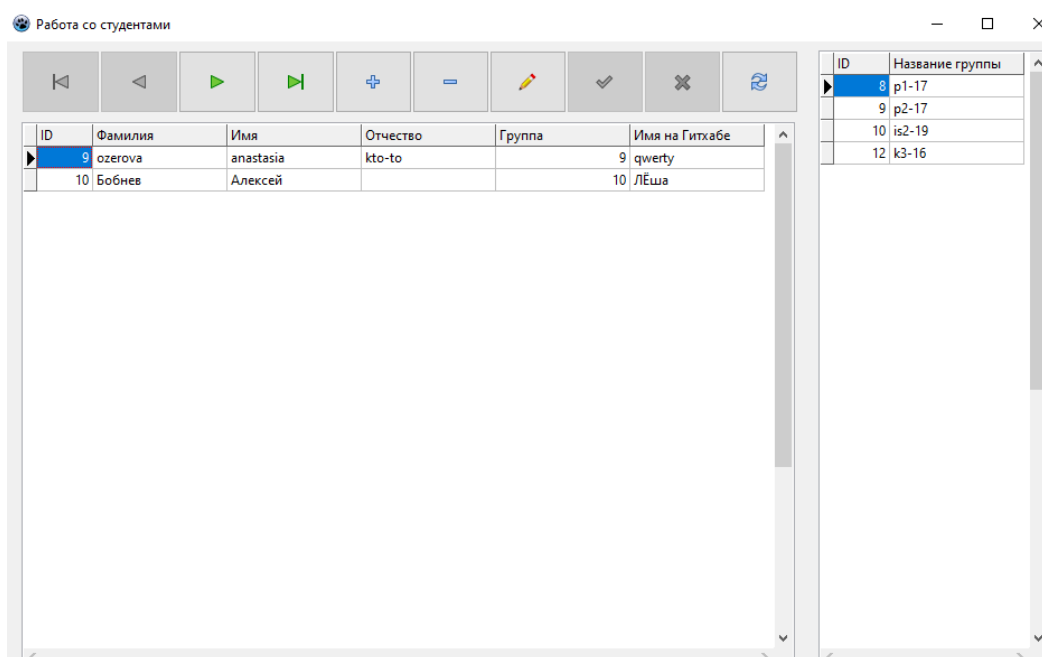


Рисунок 34. Окно " Работа со студентами"

3.2.2. Функция добавления элемента в таблицу

На панели навигации выбрать кнопку добавления, после чего заполнить появившуюся строку данными, если этого не сделать, то программа вызовет ошибку, так же на панели навигации необходимо нажать подтверждение, без этого данные не будут сохранены в базе данных.

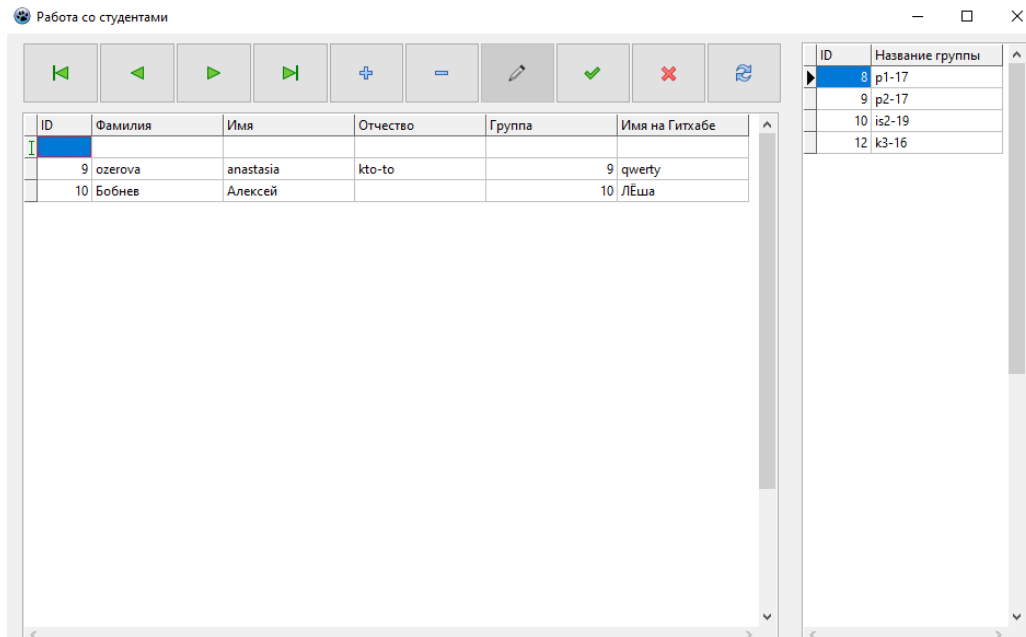


Рисунок 35. После нажатия на кнопку добавления

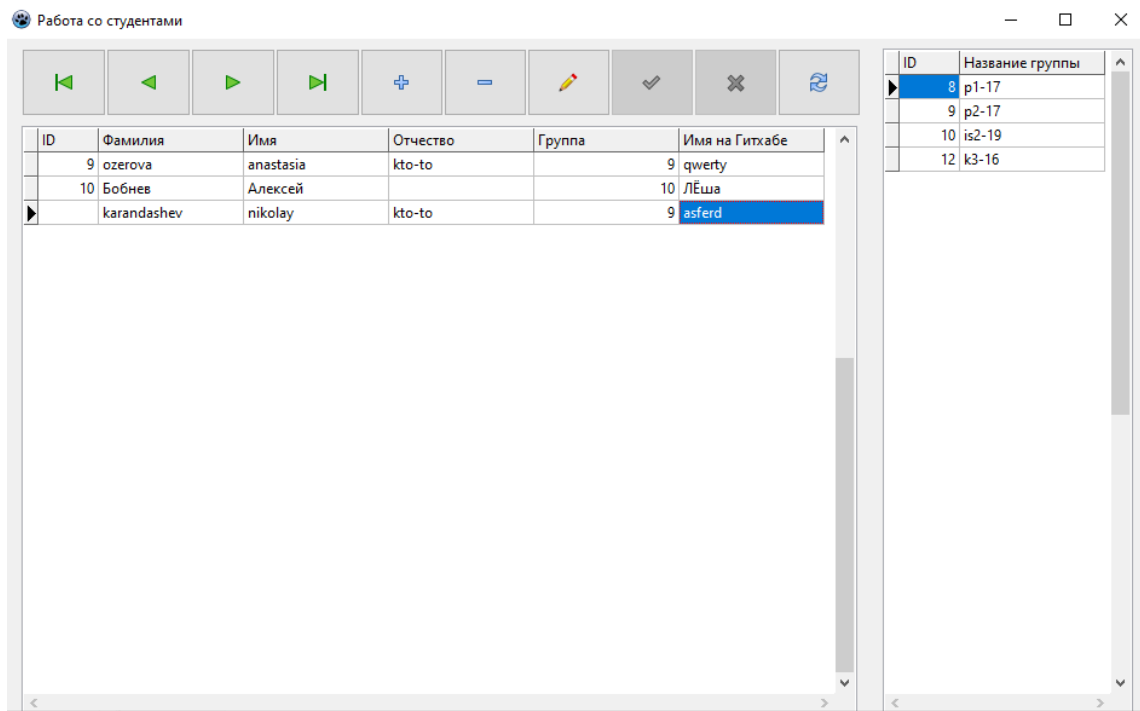


Рисунок 36. Окончание добавления

3.2.3. Функция удаления элемента из таблицы.

На панели навигации, используя кнопки передвижения, выбрать необходимую запись, после чего на этой же панели навигации нажать кнопку удаления.

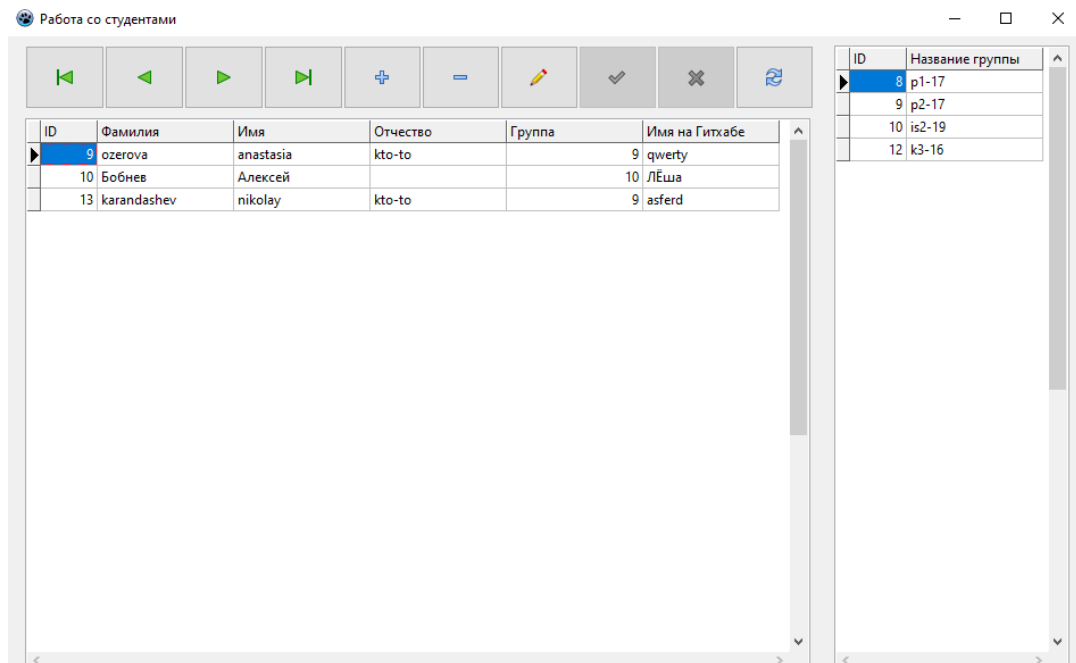


Рисунок 37. По умолчанию указатель на первый элемент

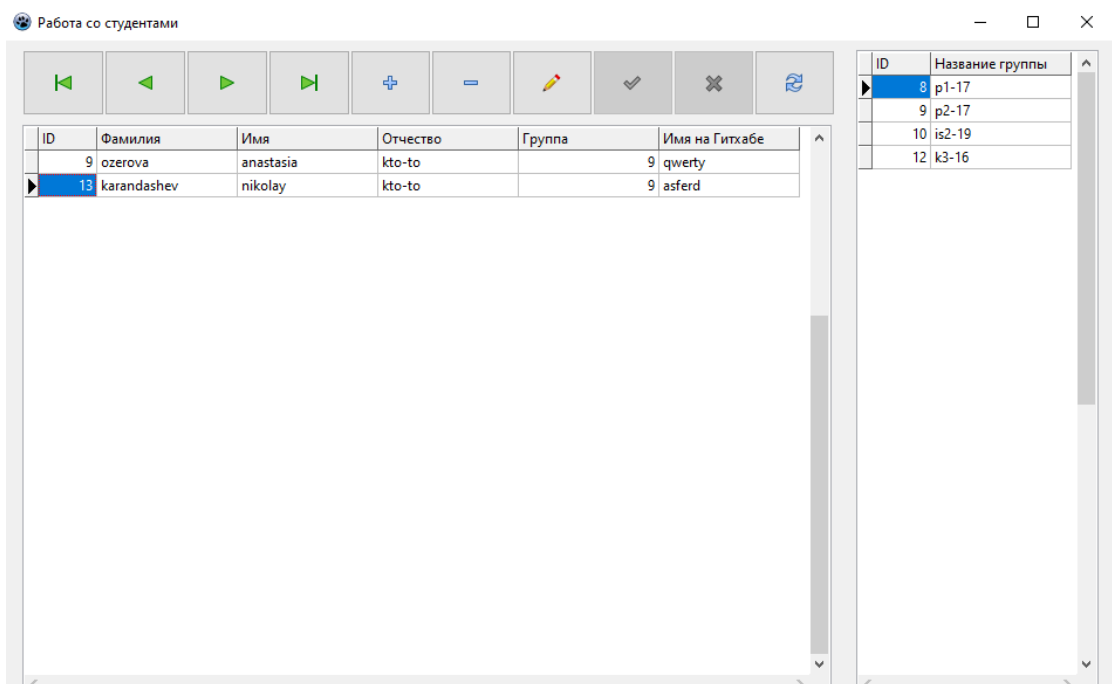


Рисунок 38. Результат удаления

3.2.4. Функция возвращения в главное меню

Чтобы вернуться в главное меню, необходимо закрыть окно, используя кнопку в верхнем правом углу.

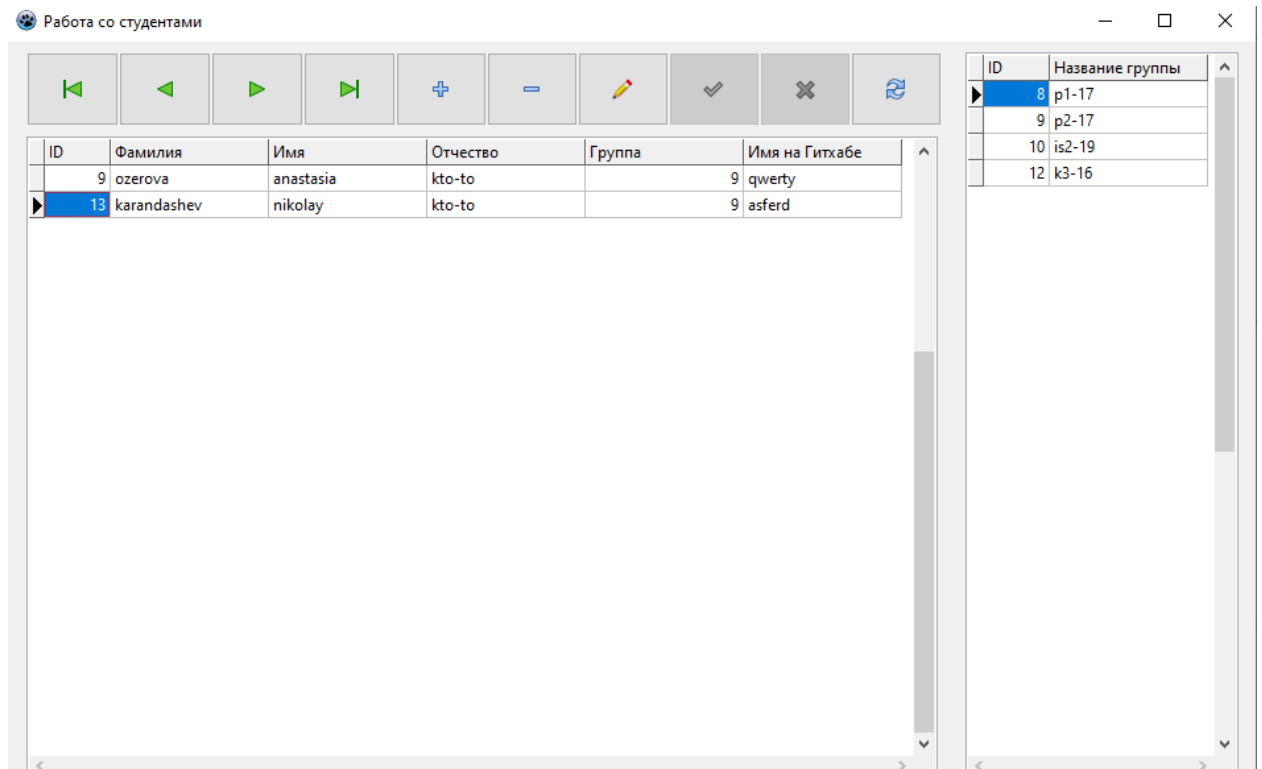


Рисунок 39. Окно "Работа со студентами"

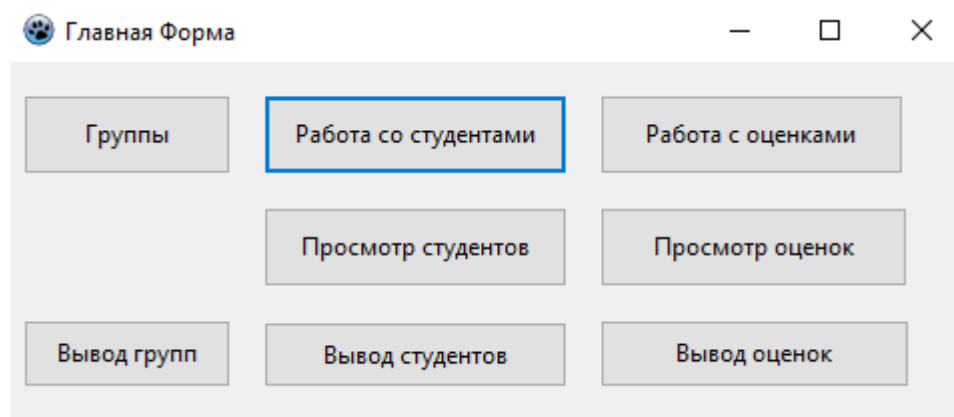


Рисунок 40. Главное окно

3.2.5. Функция открытия окна просмотра

Выберете пункт меню «Просмотр таблицы», в данном случае «Просмотр оценок», после этого откроется окно с таблицей.

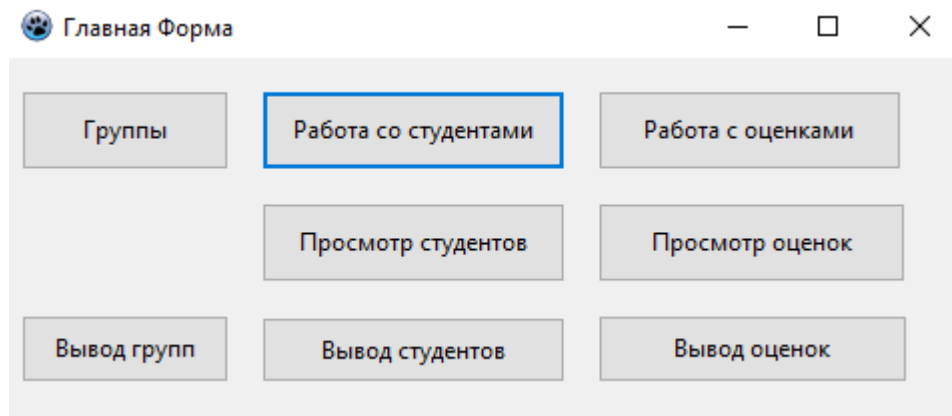


Рисунок 41. Главное окно

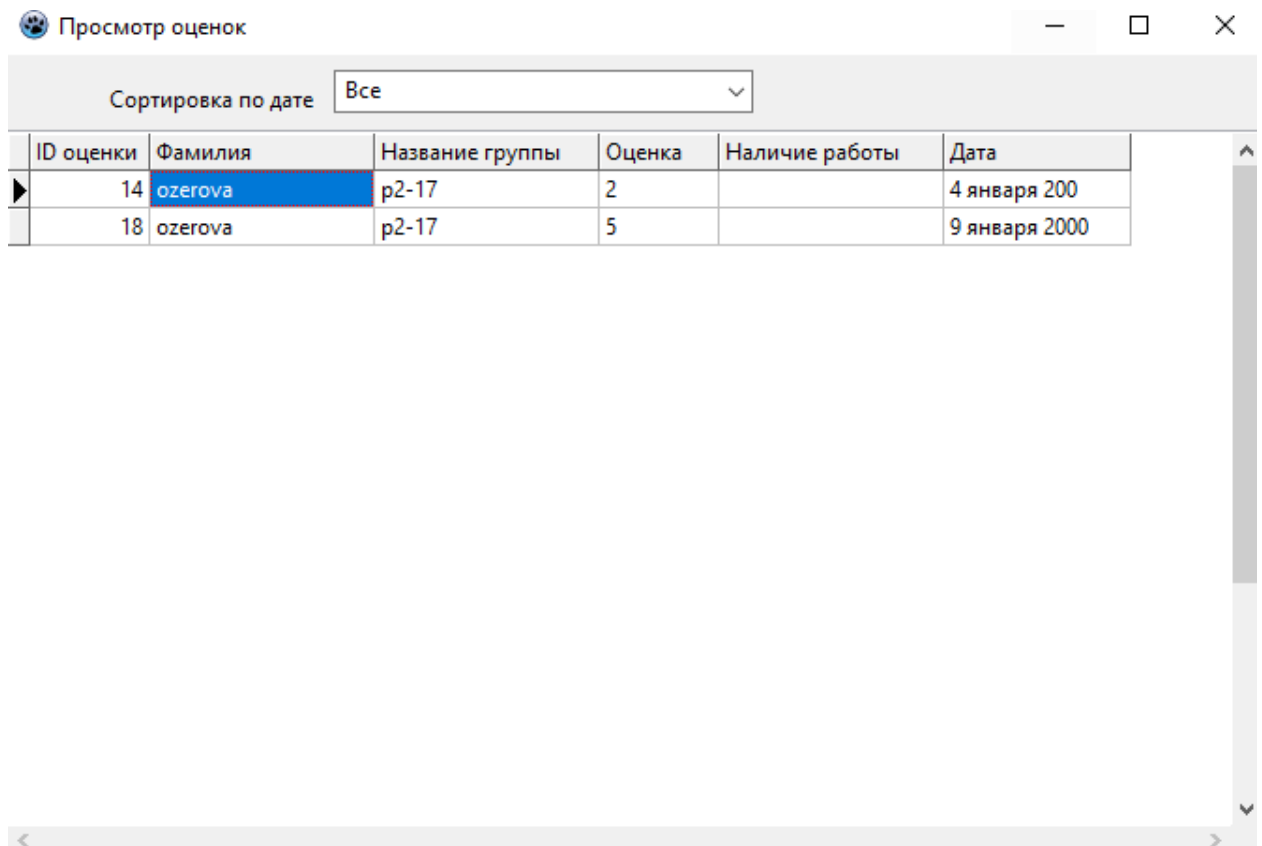


Рисунок 42. Окно "Просмотр оценок"

3.2.6. Функция сортировки

Нажмите на поле, находящееся выше таблицы. Выберите из полученного списка нужное значение, в данном случае «4 января 200».

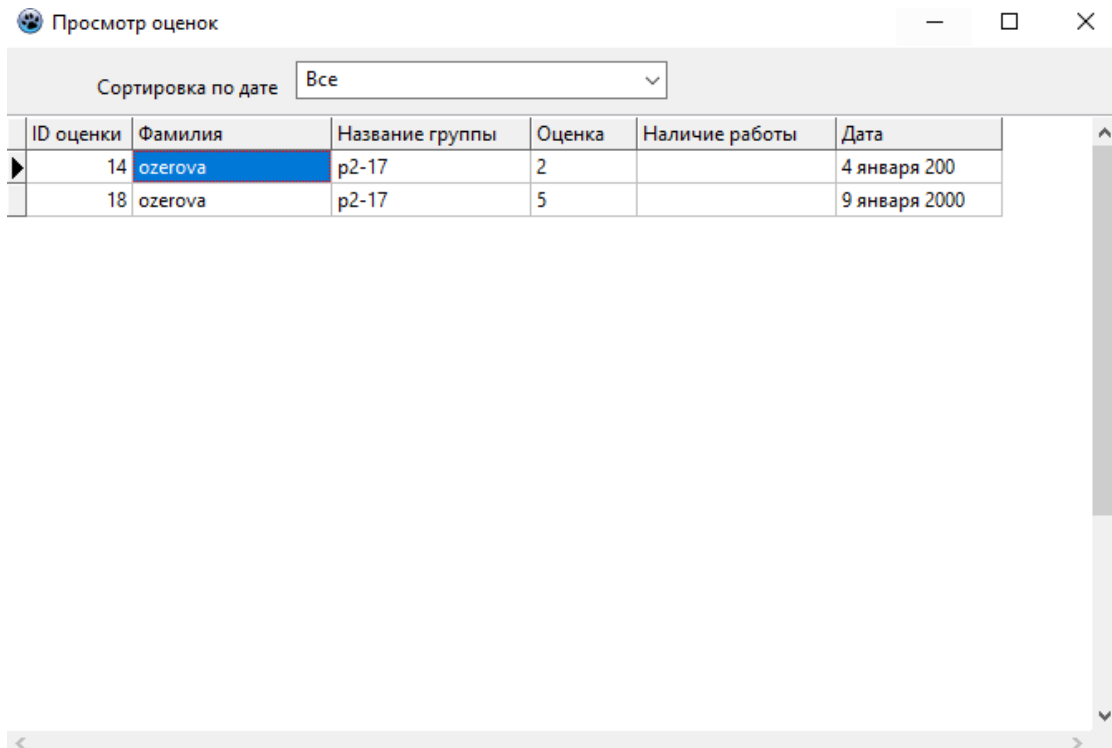


Рисунок 43. До начала сортировки

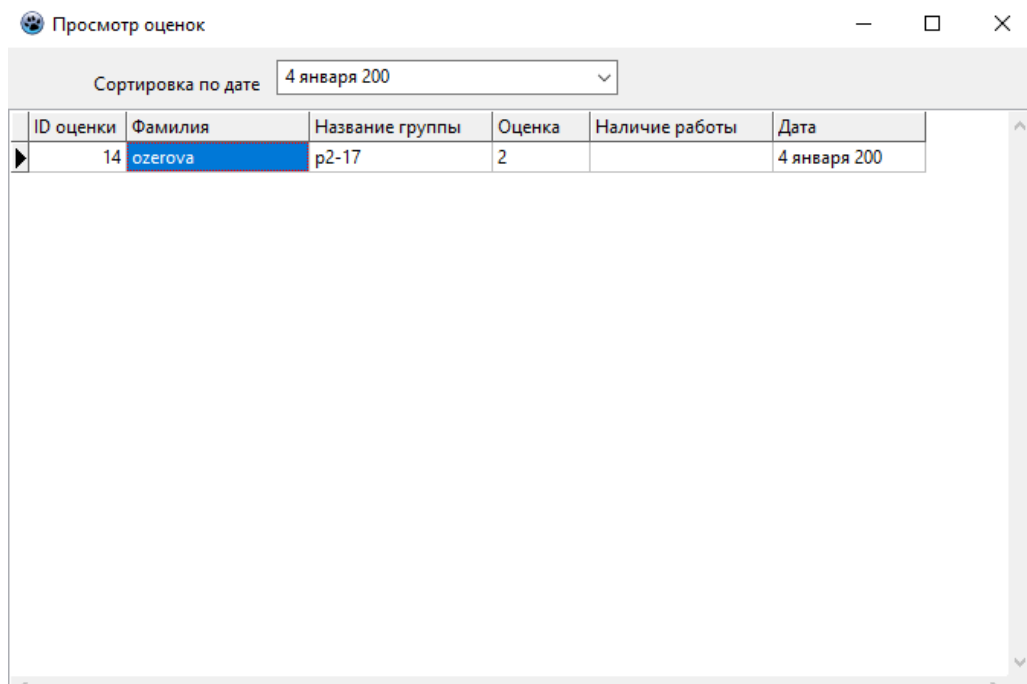


Рисунок 44. Результат сортировки

3.3 Завершение работы программы

Чтобы завершить работу программы, нажмите в главном окне на кнопку «Заккрыть» в верхнем правом углу окна.

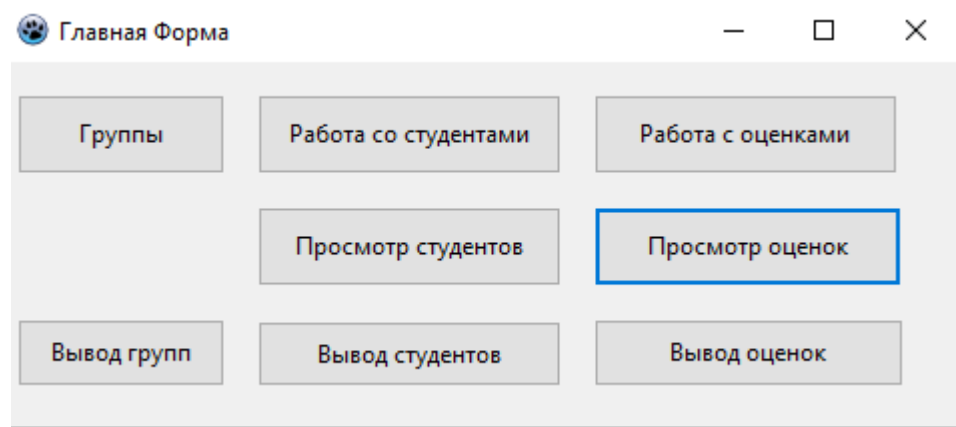


Рисунок 45. Главное окно

Заключение

В результате выполнения курсового проекта было разработано приложение «Учет выполнения работ студентами» для учета различной информации, а именно для отслеживания работ студентов и их оценок.

В ходе работы были проанализированы существующие разработки, посвященные данному направлению, проанализированы законы, направленные на сферу образования, и изучены следующие темы: «Работа с базами данных FireBird», «Разработка приложения с базами данных» и «Программирование в среде Lazarus».

В дальнейшем планирую разработать автоматическую проверку, автоматическую простановку оценок, вывод таблиц в html формате, а также сделать обработчик ошибок и поменять интерфейс.

Приложения:**Приложение 1 код главного модуля:**

```

unit maindata;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, IBDatabase, IBTable, IBCustomDataSet, IBQuery,
  db, StdCtrls, Dialogs;
type
  //список элементов формы
  TMainDatas = class(TDataModule)
    OpenDialog1: TOpenDialog;
    QDatDATAS: TIBStringField;
    QDstudIDGROUPS: TIBIntegerField;
    QDstudIDSTUDENTS: TIBIntegerField;
    QDstudNAMEGROUPS: TIBStringField;
    QDstudSECONDNAME: TIBStringField;
    QueryOcDATAS: TIBStringField;
    QueryOcHAVEWORK: TIBStringField;
    QueryOcIDOC: TIBIntegerField;
    QueryOcNAMEGROUPS: TIBStringField;
    QueryOcOCENKA: TIBStringField;
    QueryOcSECONDNAME: TIBStringField;
    SqDat: TDataSource;
    SQstud: TDataSource;
    QDat: TIBQuery;
    QDstud: TIBQuery;
    SourceOc: TDataSource;
    QueryOc: TIBQuery;
    Workstd: TIBTable;
    SourceWorkStd: TDataSource;
    LookOcenkiDATAS: TIBStringField;
    LookOcenkiHAVEWORK: TIBStringField;
    LookOcenkiIDOC: TIBIntegerField;
    LookOcenkiNAMEGROUPS: TIBStringField;
    LookOcenkiSECONDNAME: TIBStringField;
    LookOcenkiVALUE: TIBIntegerField;
    SourceNameGR: TDataSource;
    QuarryNameGR: TIBQuery;
    MainStudentFATHERNAME: TIBStringField;
    MainStudentIDGROUPS: TIBIntegerField;
    MainStudentIDSTUDENTS: TIBIntegerField;
    MainStudentNAME: TIBStringField;
    MainStudentNAMEINGIT: TIBStringField;
    MainStudentSECONDNAME: TIBStringField;
    QueryStudentsFATHERNAME: TIBStringField;
    QueryStudentsIDSTUDENTS: TIBIntegerField;
    QueryStudentsNAME: TIBStringField;
    QueryStudentsNAMEGROUPS: TIBStringField;
    QueryStudentsNAMEINGIT: TIBStringField;
    QueryStudentsSECONDNAME: TIBStringField;
    SourceStudent: TDataSource;
    MainStudent: TIBTable;
    SourceQSt: TDataSource;
    QueryStudents: TIBQuery;
    MainGroupsIDGROUPS: TIBIntegerField;
    MainGroupsNAMEGROUPS: TIBStringField;
    SourceGroupe: TDataSource;
  end;

```

```

MainGroups: TIBTable;
IdTrans: TIBTransaction;
IDData: TIBDatabase;
WorkstdDATAS: TIBStringField;
WorkstdHAVEWORK: TIBStringField;
WorkstdIDOC: TIBIntegerField;
WorkstdIDSTUD: TIBIntegerField;
WorkstdOCENKA: TIBStringField;
WorkWithOcDATAS: TIBStringField;
WorkWithOcHAVEWORK: TIBStringField;
WorkWithOcIDGROUPS: TIBIntegerField;
WorkWithOcIDOC: TIBIntegerField;
WorkWithOcIDSTUDENTS: TIBIntegerField;
WorkWithOcVALUE: TIBStringField;
procedure DataModuleCreate(Sender: TObject);
private
public
end;
var
  MainDatas: TMainDatas;
Implementation uses WorkStudentsForms;
{$R *.lfm}
{ TMainDatas }
procedure TMainDatas.DataModuleCreate(Sender: TObject);
begin //поочередно включаем все необходимые нам компоненты
if(opendialog1.Execute) then begin
  IDData.DatabaseName:='127.0.0.1:' +
ExtractFileDir(OpenDialog1.FileName)+'\MYSTUDENTS.FDB';
//записываем путь в дату
  IDData.Connected:=true; //подключаем
  IdTrans.StartTransaction; //начинаем транзакцию
  MainGroups.open; //далее открываем все запросы и таблицы
  QueryStudents.Open;
  MainStudent.Open;
  quarryNameGR.open;
  workstd.open;
  QueryOc.Open;
  Qdstud.open;
  qdat.open;
end;
end;
end.

```

Листинг №2 модуля просмотра таблицы:

```

unit LookOcenki;
{$mode objfpc}{$H+}
interface
uses
  Classes, SysUtils, Forms, Controls, Graphics, Dialogs, DBGrids,
  StdCtrls;
type
  { TLookinOc }
  TLookinOc = class(TForm)
    ComboBox1: TComboBox;
    Label1: TLabel;
    lookoc: TDBGrid;
    procedure ComboBox1Change(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
  public
  end;
var
  LookinOc: TLookinOc;
implementation
  uses maindata ;
  {$R *.lfm}
  { TLookinOc }
  procedure TLookinOc.FormCreate(Sender: TObject);
  begin
    begin
      maindata.MainDatas.Qdat.close;
      maindata.MainDatas.Qdat.open;
      maindata.MainDatas.QueryOc.close;
      maindata.MainDatas.QueryOc.open;
      while not(maindata.MainDatas.Qdat.eof) do //проходим по таблице с
//датами
        begin
          ComboBox1.Items.Add(maindata.MainDatas.Qdat.FieldName('datas').AsString); //добавляем в комбобокс соответствующее значение из запроса
            maindata.MainDatas.Qdat.next; //сдвиг указателя
        end;
      end;
    end;
  end;
  procedure TLookinOc.ComboBox1Change(Sender: TObject);
  begin
    maindata.MainDatas.QueryOc.close; //закрываем таблицу
    if not (ComboBox1.Text = 'Все') then //проверка на включение всех
      maindata.MainDatas.QueryOc.SQL[3]:=' and ocenki.datas='+ chr(39)
+ComboBox1.Text+ chr(39) //добавляем соответствующий кусок кода
    else
      maindata.MainDatas.QueryOc.SQL[3]:=''; //иначе удаляем
      maindata.MainDatas.QueryOc.open;
    end;
  end;
end.

```


Список литературы и интернет-источников

ⁱ https://ies.unitech-mo.ru/remote_provision - Образовательный портал

ⁱⁱ <https://www.skype.com/ru/> - Skype

ⁱⁱⁱ <https://discord.com/> - Discord

^{iv} <https://zoom-com.ru/?yclid=3412574462916452926> – Zoom

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учеб. пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. — Москва: ИД «ФОРУМ»: ИНФРА-М, 2019. — 400 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-104071-3. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1011120>
<https://cyberleninka.ru/article/n/sovremennoe-obrazovanie-kakoe-ono>