**Рецензия**

**на курсовую работу**

студента Шашкова Игоря Сергеевича\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

группы ИСП-О-17\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

по МДК 04.03 «Устройство функционирования информационных систем»

на темуРазработка Информационной системы “Выпускники ГГУ”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Критерии оценки | Макс. балл | Факт. балл | Комментарии |
| 1 | База данных не содержит избыточного дублирования данных | 5 |  |  |
| 2 | Наличие понятного и дружественного интерфейса с пользователем | 5 |  |  |
| 3 | Наличие необходимых видов запросов | 5 |  |  |
| 4 | Наличие унифицированной формы | 10 |  |  |
| 5 | Наличие технической документации к базе данных | 5 |  |  |
| 6 | Титульный лист оформлен в соответствии с требованиями и содержит все реквизиты | 5 |  |  |
| 7 | Материал основной части хорошо структурирован. Структура отражает логику проведенного исследования, этапы решения поставленных задач | 10 |  |  |
| 8 | Оформление работы выполнено в соответствии с требованиями | 10 |  |  |
| 9 | Использование современной литературы | 5 |  |  |
| 10 | Правильная формулировка целей и задач исследования | 5 |  |  |
| 11 | Соответствие содержания заявленной теме | 5 |  |  |
| 12 | Практическая значимость результатов работы | 5 |  |  |
| 13 | Содержание презентации удовлетворяет требованиям | 5 |  |  |
| 14 | Степень самостоятельности выполнения | 10 |  |  |
| 15 | Соблюдение сроков сдачи работы | 10 |  |  |
|  | ИТОГО: | 100 |  |  |

Шкала перевода в 5-балльную систему

86 - 100 баллов – «отлично»

70 - 85 баллов – «хорошо»

50 - 69 баллов – «удовлетворительно»

менее 50 баллов – «неудовлетворительно»

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_

Рецензент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 202\_\_г.

С оценкой ознакомлен \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Гжельский государственный университет» (ГГУ)

Колледж ГГУ

**ЗАДАНИЕ**

**ДЛЯ КУРСОВОЙ РАБОТЫ**

По МДК 04.03 «Устройство функционирования информационных систем»

Студенту 3 курса специальности 09.02.07 Информационные системы и программирования

Ф.И.О. \_\_\_\_\_

Тема:Разработка Информационной системы “Выпускники ГГУ”

**СОДЕРЖАНИЕ КУРСОВОЙРАБОТЫ**

**Введение**

**1 Проектирование базы данных**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2 Разработка базы данных**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3 Создание базы данных**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**4 Разработка руководства пользователю базой данных**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Заключение**

**Список использованных источников**

**Приложения**

Дата выдачи задания «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_\_г.

Срок выполнения задания «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_\_г.

**Руководитель курсового проектирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Председатель П(Ц)К \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Гжельский государственный университет» (ГГУ)

Колледж ГГУ

**РАЗРАБОТКА Информационной системы**

**“ВЫпускники ГГУ”**

09.02.03 – КР.02 – 18 – 0810

Курсовая работа

|  |  |
| --- | --- |
| Разработал | И.С.Шашков |
| Руководитель | Т.А.Промзелёва |
|  |  |

Электроизолятор

2020

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Содержание**   |  |  | | --- | --- | | Пояснительная записка..………………………………...……………………….. | 4 | | 1 Проектирование базы данных.……………………………..………………... | 6 | | 1.2 Создание базы данных ………………………………………………………. | 9 | | 1.3 Наполнение базы данных ….………………………........................................ | 11 | | Заключение………………………………………………………………………… | 12 | | Приложение А…………………………………………………………………….. | 13 | | Приложение Б……………………………………………………………………... | 14 | | Приложение В……………………………………………………………………... | 25 | |

**1 Проектирование базы данных**

**1.1 Предметная область**

В данном проекте выполняется разработка информационной системы

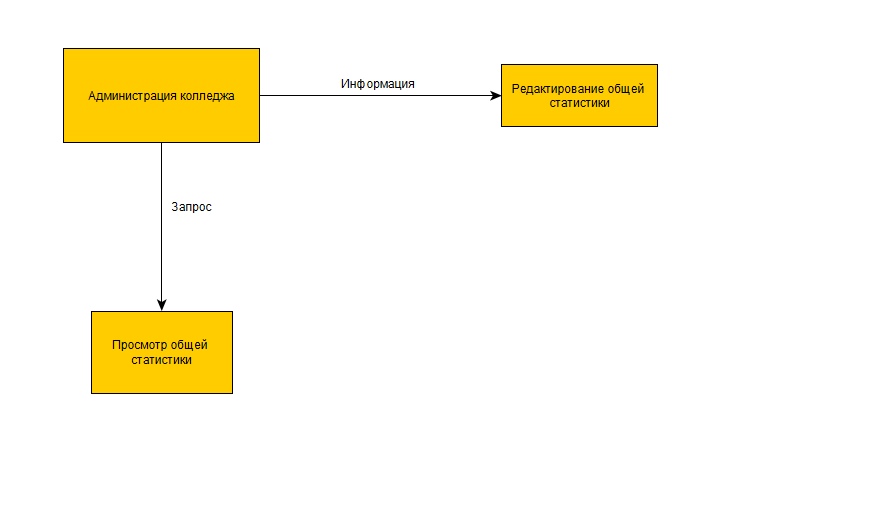
«Выпускники ГГУ».

**1.2 Описание предметной области**

У БД «Выпускники ГГУ» есть 4 сущности: Студент, Специальность, Группа, Администрация Колледжа.

**1.3 Потоки**

Показано на диаграмме потоков данных.



**1.4 Состав пользовательской БД и их информационные потребности**

**Состав:** Бывшие студенты ГГУ, общая информация по каждому студенту (Ф.И.О., Место проживания, дата начала-конца обучения, бывшая группа и т.д.)

**Потребности:** Вывод общей информации по каждому студенту, сортировка и

поиск по студента.

**1.5 Автоматизированные функции**

Автоматизированных функций не предусмотрено.

**1.6 Список данных, требуемых для реализации функций**

Список выпускников и информации по ним.

**1.7 Список данных, требуемых для реализации автоматизированных функций**

Данных для автоматизированных функций не предусмотрено.

**1.8 Список не повторяющихся данных с кратким описанием**

Студент – Ф.И.О. студента(ключ сущности Студенты).

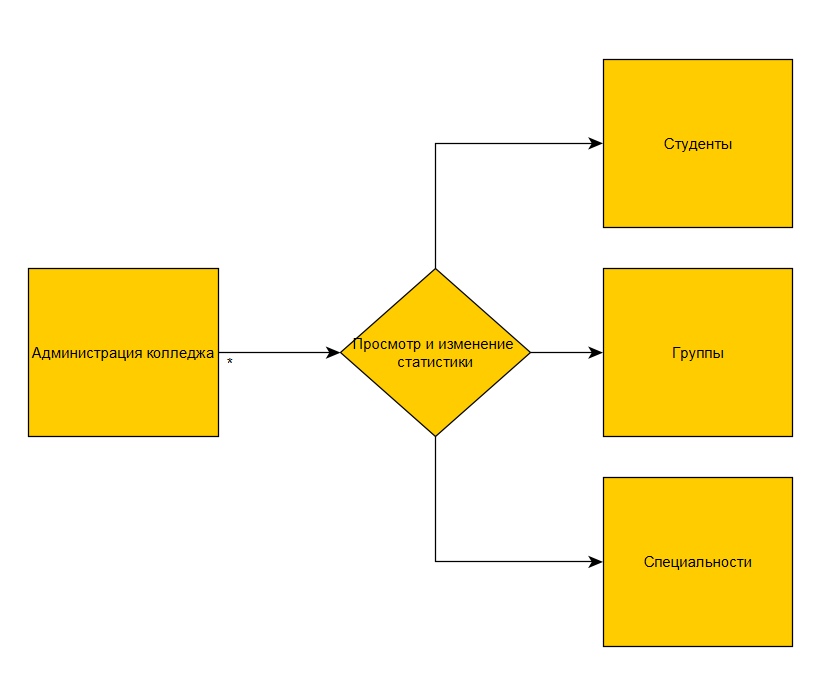
Группа\_ID – аббревиатура группы(ключ сущности Группы).

ID (специальности) – сокращенное наименование специальности (ключ сущности Специальности).

ПИН-КОД – пин код для входа в БД (ключ сущности Администрация колледжа).

**1.9 Структурные связи**

Структурные связи показаны на диаграмме сущность связь:

[](https://vk.com/photo322578463_457244612)

**Описание связей**

**1.9.1 Просмотр и изменение статистики (Администрация колледжа - Студенты)**

Администратор колледжа может изменять и просматривать информацию о студентах (выпускники).

**1.9.2 Просмотр и изменение статистики (Администрация колледжа - Группы)**

Администратор колледжа может изменять и просматривать информацию о группах.

**1.9.3 Просмотр и изменение статистики (Администрация колледжа - Специальности)**

Администратор колледжа может изменять и просматривать информацию о специальностях.

|  |
| --- |
| **1.10 ER модель БД**  Данная ER модель сделана по стандарту IDEF1.  [https://sun9-4.userapi.com/c854220/v854220505/115e1a/fJpRn1UsnY0.jpg](https://vk.com/photo322578463_457244611) |

|  |
| --- |
| **2 Создание базы данных.**  **2.1 Таблицы отражающие сущность БД**  Таблица сущности Студенты:    Таблица сущности Группы:    Таблица сущности специальности:    **2.2 Характеристики атрибутов**  **2.2.1 Тип атрибута**  Дата\_Начала\_Обучения – Цифровой тип данных.  Дата\_Конца \_Обучения – Цифровой тип данных.  Оценка (Дипломная работа) – Цифровой тип данных.  Время обучения – Цифровой тип данных.  Все другие данные – Текстовой тип.  **2.2.2 Вычисляемые поля**  В данной БД они отсутствуют.  **2.3 Ключи таблиц**  Студент – ключ сущности Студенты.  Группа\_ID – ключ сущности Группы.  ID (специальности) - ключ сущности Специальности.  ПИН-КОД- ключ сущности Администрация колледжа.  **2.4 Три вида обязательных форм**  **2.4.1 Управления транзакции**  Есть возможность при необходимости вводить, удалять и изменять данные с помощью элементов формы.  **2.4.2 Для доступа к данным**  Обработка данных не требуется.  **2.4.3 Управляющая(кнопочная)**  Отсутствуют из-за не надобности.  **2.4.4 Создания отчётов**  Отсутствуют по заданию.  **2.5 Запросы и Макросы**  Отсутствуют.  **3 Наполнение БД**  Каждая таблица заполнена более чем на 10 пунктов.  **Заключение**  **В результате выполнения курсовой работы была разработана информационная система “Выпускники ГГУ”, которая позволяет изменять и просматривать данные выпускниках ГГУ.** |
| **Приложение А**  Рисунок 1 – Начальная форма приложения      Рисунок 2 – Основная форма приложения    **Приложение Б**   |  | | --- | | **РД 50-34.698-90 (п.п. 3.4 Руководство пользователя)** | | **Введение** | | Область применения | | Описание возможностей | | Уровень подготовки пользователя | | Перечень эксплуатационной документации, с которыми необходимо ознакомиться пользователю | | **Назначение и условия применения** | | Виды деятельности и функции, для автоматизации которых предназначена программа | | Условия, при которых обеспечивается применение программы | | **Подготовка к работе** | | Состав и содержание дистрибутивного носителя данных | | Порядок загрузки данных и программ | | Проверка работоспособности | | **Описание операций** | | Описание всех выполняемых функций, задач, комплексов задач, процедур | | Описание операций технологического процесса обработки данных, необходимых для выполнения функций, комплексов задач (задач), процедур | | **Аварийные ситуации** (действия в случае длительных отказов технических средств, обнаружении несанкционированного вмешательства в данные, действия по восстановлению программ или данных, и т.п. ) | | **Рекомендации по освоению** |   **1 Введение**  Данное руководство предназначено для использования системы.  **1.1 Область применения**  Предназначено для пользования администрацией колледжа.  **1.2 Описание возможностей**  Даёт возможность для просмотра и изменения статистики выпускников ГГУ.  **1.3 Уровень подготовки пользователя**  Использование И.С. не требует подготовки.  **1.4 Перечень эксплуатационной документации, с которой необходимо**  **ознакомится пользователю**  Данная документация отсутствует.  **2 Назначение и условия применения**  Просмотр и/или изменение статистики администрацией колледжа.  **2.1 Виды деятельности и функции, для автоматизации которых предназначена программа**  Поиск информации о выпускниках ГГУ.  **2.2 Условия, при которых обеспечивается применение программы**  **2.2.1 Требования к аппаратному обеспечению**  Требуется компьютер работающий на операционных системах Windows 7/10 и разрешение экрана должно быть больше 800x600.  **2.2.2 Квалификация пользователя**  Пользователь должен обладать навыками использования операционных систем Windows 7/10(Смотря на какой О.С. компьютер).  **3 Подготовка к работе**  Требуется только узнать ПИН-КОД для входа в приложение  (при первом запуске требуется ввести его).  **3.1 Состав и содержание дистрибутивного носителя данных**  И.С. Будет находится на ПК пользователя.  **3.2 Порядок загрузки данных и программ**  Запуск – происходит с помощью двойного клика на ярлык программы.  Рис 1. Приложения А.  Работа в программе происходит в главном окне.  Рис 2. Приложения А.  Завершение – клик на крестик в правом верхнем углу.  **3.3 Проверка работоспособности**  ф  **4 Описание операций**  1 поиск по не полному совпадению(атрибуты: группе(ID), Месту проживания, Оценка(Дипломная работа), Куратор, Времени обучения)    2 сортировка (атрибуты: группа(ID), Кол-во студентов, Имя , ID, Оценка дипломной работы)    3 Отбор по 3 критериям(атрибутам: группа(ID), Оценка дипломной работы(3,4 или 5) Место проживания)  1 отбор:      Пример отсутствует из-за малого кол-ва групп.  2 отбор:      Пример выбора отбора по оценке 5    3 отбор:    Пример отбора по месту жительства – Раменское    4 Функция сортировки наоборот    5 Функция ввода ПИН-КОД’а    6 Функция смены ПИН-КОД’а    **4.1 Описание всех выполняемых функций, задач, комплексов задач, процедур**  **4.1.1 поиск по не полному совпадению(атрибуты: группе(ID), Месту проживания, Оценка(Дипломная работа), Куратор, Времени обучения)**    Данная функция ищет записи с атрибутом, который вы вписываете в окне справа  Пример работы:      **4.1.2 сортировка (атрибуты: группа(ID), Кол-во студентов, Имя , ID, Оценка дипломной работы)**    Данная функция сортирует данные по возрастанию  Пример работы:      **4.1.3 Отбор по 3 критериям(атрибутам: группа(ID), Оценка дипломной работы(3,4 или 5) Место проживания)**  Функция выполняет быстрый поиск с заранее заданными критериями.  1 отбор:      Пример отсутствует из-за малого кол-ва групп.  2 отбор:      Пример выбора отбора по оценке 5    3 отбор:    Пример отбора по месту жительства – Раменское    **4.1.4 Функция сортировки наоборот**  Меняет выбранную сортировку на сортировку по убыванию и обратно.    Пример работы:  До нажатия(сортировка по оценке диплома)    После нажатия кнопки “Сортировка наоборот”, критерий тот-же(оценка диплома):    **4.1.5 Ввод ПИН-КОД’а**    Проверка правильности ввода пин-кода.  **4.1.6 смена пин-кода**    Смена пин-кода после проверки правильности его ввода.  **4.2 Описание операций технологического процесса обработки данных, необходимых для выполнения функций, комплексов задач (задач), процедур**  Обработка данных происходит при подтверждении личности (Вводе ПИН-КОДА).  **Ф**  При нажатии кнопки ввести ПИН-КОД, программа проверяет файл Pas.txt в корне программы, если код совпадает, программа запускается  Также обработка данных происходит при смене пин-кода.    При нажатии кнопки ввести пин-код, программа проверяет файл Pas.txt в корне программы, если кода совпадает, программа запускается и сохраняет новый пароль.  **5. Аварийные ситуации(действия в случае длительных отказов технических средств, обнаружении несанкционированного вмешательства в данные, действия по восстановлению программ или данных, и т.п. )**  1 при сбое программы перезапустить её, если не помогло, переустановите программу.    **6. Рекомендации по освоению**  Отсутвует.  **Приложение В**  **Листинг**  **MainBDW.cpp**  //---------------------------------------------------------------------------  #include <vcl.h>  #pragma hdrstop  #include <windows.h>  #include "MainBDW.h"  #include "MessageBoxOcenki.h"  #include "MessageBoxViborGruppi.h"  #include "fstream"  //---------------------------------------------------------------------------  #pragma package(smart\_init)  #pragma resource "\*.dfm"  TMBDWindow \*MBDWindow;  int zetta = 0, Password, Nachalnii = 0;  UnicodeString TableStudents = "Студенты", TableGroups = "Группы", TableSpec = "Специальности", CurrentUsedTable, Path = (ExtractFilePath(Application->ExeName) + "\\Pin.txt"); //Path - путь для файла сохранения пароля  std::ifstream iOP;//принятый пароль(записанный пароль)  std::ofstream oZP;//отправленный пароль(запись)  //---------------------------------------------------------------------------  \_\_fastcall TMBDWindow::TMBDWindow(TComponent\* Owner)  : TForm(Owner)  {  ADOConnection1 -> Close();  ADOConnection1 -> ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;User ID=Admin;Data Source=" + ExtractFilePath(Application->ExeName) + "\\Vipuskniki.mdb;Mode=Share Deny None;Jet OLEDB:System database="";Jet OLEDB:Registry Path="";Jet OLEDB:Database Password="";Jet OLEDB:Engine Type=5;Jet OLEDB:Database Locking Mode=1;Jet OLEDB:Global Partial Bulk Ops=2;Jet OLEDB:Global Bulk Transactions=1;Jet OLEDB:New Database Password="";Jet OLEDB:Create System Database=False;Jet OLEDB:Encrypt Database=False;Jet OLEDB:Don't Copy Locale on Compact=False;Jet OLEDB:Compact Without Replica Repair=False;Jet OLEDB:SFP=False;";  CurrentUsedTable = TableStudents;  ADOQuery1->SQL->Clear();  ADOConnection1 -> Open();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable;  ADOQuery1 -> Active = true;  DBNavigator1 -> Visible = false;  Button1 -> Visible = false;  Button2 -> Visible = false;  Button3 -> Visible = false;  Button4 -> Visible = false;  Button5 -> Visible = false;  Button6 -> Visible = false;  Button7 -> Visible = false;  DBGrid1 -> Visible = false;  Edit1 -> Visible = false;  Edit2 -> Visible = false;  Label1 -> Visible = false;  ComboBox1 -> Visible = false;  ComboBox2 -> Visible = false;  Label2 -> Visible = false;  Label4 -> Visible = false;  ochistka -> Visible = false;  Button8 -> Visible = true;  Edit3 -> Visible = true;  iOP.open(Path.t\_str());  iOP >> Password;  iOP.close();  if(Password == Nachalnii)  {  ShowMessage("Это Ваше первое открытие программы, пожалуйста введите Ваш пин-код");  }  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::Button1Click(TObject \*Sender)  {  CurrentUsedTable = TableGroups;  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable;  ADOQuery1->Active = true;  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::Button2Click(TObject \*Sender)  {  CurrentUsedTable = TableSpec;  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable;  ADOQuery1->Active = true;  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::Button3Click(TObject \*Sender)  {  CurrentUsedTable = TableStudents;  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable;  ADOQuery1->Active = true;  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::Button5Click(TObject \*Sender)  {  MesageBox -> Visible = true;  MesageBox -> Edit1 -> Text = CurrentUsedTable;  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::Button4Click(TObject \*Sender)  {  MesageBoxGruppi -> Edit1 -> Text = CurrentUsedTable;  MesageBoxGruppi -> Visible = true;  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::Button6Click(TObject \*Sender)  {  if(CurrentUsedTable == TableStudents)  {  CurrentUsedTable = TableStudents;  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Место\_проживания LIKE '%Раменское%'";  ADOQuery1->Active = true;  }  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::ochistkaClick(TObject \*Sender)  {  CurrentUsedTable = TableStudents;  ComboBox1 -> Text = "Выберите";  ComboBox2 -> Text = "Выберите";  Edit1 -> Text = "";  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable;  ADOQuery1->Active = true;  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::PreduprejdenieTimer(TObject \*Sender) //Авто определение расположения элементов UI на экране  {  /\*if (Edit2 -> Text == " SELECT \* FROM Студенты WHERE Место\_проживания LIKE '%Раменское%'") {  Button8 -> Caption = "JOPA";  }  else PAMAGITI NE CHITAET PACHEMUTO!!!!  {  Button8 -> Caption = "NeJOPA";  }\*/  if(ComboBox1 -> Text == "Выберите")  {  Edit1 -> Enabled = false;  }  else  {  Edit1 -> Enabled = true;  }  Edit2 -> Text = ADOQuery1->SQL->Text;  DBNavigator1 -> Height = 110;  DBNavigator1 -> Width = 300;  DBNavigator1 -> Left = ClientWidth - DBNavigator1 -> Width; //Чем больше LEFT тем больше двигается в право  DBNavigator1 -> Top = ClientHeight - DBNavigator1 -> Height; //Чем больше TOP тем ниже двигается в низ  Button1 -> Height = 34;  Button1 -> Width = 180;  Button1 -> Left = DBNavigator1 -> Left - Button1 -> Width ;  Button1 -> Top = DBNavigator1 -> Top ;  Button2 -> Height = 34;  Button2 -> Width = 180;  Button2 -> Left = Button1 -> Left;  Button2 -> Top = Button1 -> Top + Button2 -> Height + 4;  Button3 -> Height = 34;  Button3 -> Width = 180;  Button3 -> Left = Button1 -> Left;  Button3 -> Top = Button2 -> Top + Button3 -> Height + 4;  Button4 -> Height = 35;  Button4 -> Width = 150;  Button4 -> Left = ClientWidth - ClientWidth;  Button4 -> Top = ClientHeight - Button4 -> Height;  Button5 -> Height = 35;  Button5 -> Width = 150;  Button5 -> Left = Button4 -> Left + Button5 -> Width + 1;  Button5 -> Top = Button4 -> Top;  Button6 -> Height = 35;  Button6 -> Width = 150;  Button6 -> Left = Button5 -> Left + Button6 -> Width + 1;  Button6 -> Top = Button5 -> Top ;  Button7 -> Height = 35;  Button7 -> Width = 125;  Button7 -> Left = (ClientWidth - ClientWidth) + (Button4 -> Width + Button5 -> Width + Button6 -> Width) + 2 - (Button6 -> Width/2);  Button7 -> Top = Button6 -> Top - Button7 -> Height - 4;  DBGrid1 -> Height = ClientHeight - DBNavigator1 -> Height;  DBGrid1 -> Width = ClientWidth;  Edit1 -> Left = ComboBox1 -> Left + 156;  Edit1 -> Top = ComboBox1 -> Top;  Edit2 -> Left = ochistka -> Left;  Edit2 -> Top = ochistka -> Top - 50;  Label1 -> Left = Label2 -> Left ;  Label1 -> Top = Label2 -> Top - 25 ;  ComboBox1 -> Left = Label1 -> Left + 50 ;  ComboBox1 -> Top = Label1 -> Top - 3 ;  ComboBox2 -> Left = Label2 -> Left + 80;  ComboBox2 -> Top = Button7 -> Top - 4;  Label2 -> Left = Label4 -> Left;  Label2 -> Top = Button7 -> Top - 1;  Label4 -> Left = ClientWidth - ClientWidth + 8;  Label4 -> Top = Button6 -> Top - 15;  ochistka -> Height = 35;  ochistka -> Width = 100;  ochistka -> Left =(ClientWidth/2) - ochistka -> Width;  ochistka -> Top = ClientHeight - ochistka -> Height;  Button8 -> Left = ClientWidth / 2;  Button8 -> Top = ClientHeight / 2;  Edit3 -> Top = Button8 -> Top - 50;  Edit3 -> Left = Button8 -> Left - 15;  Button9 -> Left = 10;  Button9 -> Top = 0;  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::ComboBox2Change(TObject \*Sender)  {  if(ComboBox2->Text == "ID" && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY ID ASC";  ADOQuery1->Active = true;  zetta = 0;  }  else if(ComboBox2->Text == "ID" && !(CurrentUsedTable == TableStudents))  {  ShowMessage("Для данной соритировки требуется открыть табл. Студенты");  }  else if(ComboBox2->Text == "По имени" && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Студент ASC";  ADOQuery1->Active = true;  zetta = 0;  }  else if(ComboBox2->Text == "По имени" && !(CurrentUsedTable == TableStudents))  {  ShowMessage("Для данной соритировки требуется открыть табл. Студенты");  }  else if(ComboBox2->Text == "Группе(ID)" && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Группа\_ID ASC";  ADOQuery1->Active = true;  zetta = 0;  }  else if(ComboBox2->Text == "Группе(ID)" && !(CurrentUsedTable == TableStudents))  {  ShowMessage("Для данной соритировки требуется открыть табл. Студенты или группы");  }  else if(ComboBox2->Text == "Кол-ву студентов" && CurrentUsedTable == TableGroups)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Количество\_студентов ASC";  ADOQuery1->Active = true;  zetta = 0;  }  else if(ComboBox2->Text == "Кол-ву студентов" && !(CurrentUsedTable == TableGroups))  {  ShowMessage("Для данной соритировки требуется открыть табл. Группы");  }  else if(ComboBox2->Text == "Оценке дипломной работы" && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Оценка\_Дипломная\_Работа ASC";  ADOQuery1->Active = true;  zetta = 0;  }  else if(ComboBox2->Text == "Оценке дипломной работы" && !(CurrentUsedTable == TableStudents))  {  ShowMessage("Для данной соритировки требуется открыть табл. Студенты");  }  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::Button7Click(TObject \*Sender)  {  if(ComboBox2->Text == "ID" && zetta == 0 && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY ID DESC";  ADOQuery1->Active = true;  zetta++;  }  else if(ComboBox2->Text == "ID" && zetta == 1 && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY ID ASC";  ADOQuery1->Active = true;  zetta = 0;  }  else if(ComboBox2->Text == "По имени" && zetta == 0 && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Студент DESC";  ADOQuery1->Active = true;  zetta++;  }  else if(ComboBox2->Text == "По имени" && zetta == 1 && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Студент ASC";  ADOQuery1->Active = true;  zetta = 0;  }  else if(ComboBox2->Text == "Группе(ID)" && zetta == 0 && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Группа\_ID DESC";  ADOQuery1->Active = true;  zetta++;  }  else if(ComboBox2->Text == "Группе(ID)" && zetta == 1 && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Группа\_ID ASC";  ADOQuery1->Active = true;  zetta = 0;  }  else if(ComboBox2->Text == "Кол-ву студентов" && zetta == 0 && CurrentUsedTable == TableGroups)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Количество\_студентов DESC";  ADOQuery1->Active = true;  zetta++;  }  else if(ComboBox2->Text == "Кол-ву студентов" && zetta == 1 && CurrentUsedTable == TableGroups)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Количество\_студентов ASC";  ADOQuery1->Active = true;  zetta = 0;  }  else if(ComboBox2->Text == "Оценке дипломной работы" && zetta == 0 && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Оценка\_Дипломная\_Работа DESC";  ADOQuery1->Active = true;  zetta++;  }  else if(ComboBox2->Text == "Оценке дипломной работы" && zetta == 1 && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " ORDER BY Оценка\_Дипломная\_Работа ASC";  ADOQuery1->Active = true;  zetta = 0;  }  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::Edit1Change(TObject \*Sender)  {  if(ComboBox1->Text == "группе(ID)." && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Группа\_ID LIKE '%"+ Edit1 -> Text +"%'";  ADOQuery1->Active = true;  }  else if(ComboBox1->Text == "группе(ID)." && !(CurrentUsedTable == TableStudents))  {  ShowMessage("Для данной соритировки требуется открыть табл. Студенты");  }  else if(ComboBox1->Text == "Месту проживания." && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Место\_проживания LIKE '%"+ Edit1 -> Text +"%'";  ADOQuery1->Active = true;  }  else if(ComboBox1->Text == "Месту проживания." && !(CurrentUsedTable == TableStudents))  {  ShowMessage("Для данной соритировки требуется открыть табл. Студенты");  }  else if(ComboBox1->Text == "Оценке(Дипломная работа)." && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Оценка\_Дипломная\_Работа LIKE '%"+ Edit1 -> Text +"%'";  ADOQuery1->Active = true;  }  else if(ComboBox1->Text == "Оценке(Дипломная работа)." && !(CurrentUsedTable == TableStudents))  {  ShowMessage("Для данной соритировки требуется открыть табл. Студенты");  }  else if(ComboBox1->Text == "Куратору." && (CurrentUsedTable == TableGroups || CurrentUsedTable == TableSpec))  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Куратор LIKE '%"+ Edit1 -> Text +"%'";  ADOQuery1->Active = true;  }  else if(ComboBox1->Text == "Куратору." && !(CurrentUsedTable == TableGroups))  {  ShowMessage("Для данной соритировки требуется открыть табл. Группы или Специальности");  }  else if(ComboBox1->Text == "Времени обучения." && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Время обучения LIKE '%"+ Edit1 -> Text +"%'";  ADOQuery1->Active = true;  }  else if(ComboBox1->Text == "Времени обучения." && !(CurrentUsedTable == TableSpec))  {  ShowMessage("Для данной соритировки требуется открыть табл. Специальности");  }  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::Button8Click(TObject \*Sender)  {  iOP.open(Path.t\_str());  iOP >> Password;  iOP.close();  if(Password == Nachalnii)  {  Password = StrToInt(Edit3 -> Text);  if(Password == Nachalnii)  {  Nachalnii++;  }  oZP.open(Path.t\_str());  oZP << Password;  oZP.close();  }  if(Edit3 -> Text == Password)  {  DBNavigator1 -> Visible = true;  Button1 -> Visible = true;  Button2 -> Visible = true;  Button3 -> Visible = true;  Button4 -> Visible = true;  Button5 -> Visible = true;  Button6 -> Visible = true;  Button7 -> Visible = true;  DBGrid1 -> Visible = true;  Edit1 -> Visible = true;  Edit2 -> Visible = false;  Label1 -> Visible = true;  ComboBox1 -> Visible = true;  ComboBox2 -> Visible = true;  Label2 -> Visible = true;  Label4 -> Visible = true;  ochistka -> Visible = true;  Button8 -> Visible = false;  Edit3 -> Visible = false;  Button9 -> Visible = false;  Edit4 -> Visible = false;  Edit5 -> Visible = false;  }  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::ComboBox1Change(TObject \*Sender)  {  if(ComboBox1->Text == "группе(ID)." && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Группа\_ID LIKE '%"+ Edit1 -> Text +"%'";  ADOQuery1->Active = true;  }  else if(ComboBox1->Text == "группе(ID)." && !(CurrentUsedTable == TableStudents))  {  ShowMessage("Для данной соритировки требуется открыть табл. Студенты");  }  else if(ComboBox1->Text == "Месту проживания." && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Место\_проживания LIKE '%"+ Edit1 -> Text +"%'";  ADOQuery1->Active = true;  }  else if(ComboBox1->Text == "Месту проживания." && !(CurrentUsedTable == TableStudents))  {  ShowMessage("Для данной соритировки требуется открыть табл. Студенты");  }  else if(ComboBox1->Text == "Оценке(Дипломная работа)." && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Оценка\_Дипломная\_Работа LIKE '%"+ Edit1 -> Text +"%'";  ADOQuery1->Active = true;  }  else if(ComboBox1->Text == "Оценке(Дипломная работа)." && !(CurrentUsedTable == TableStudents))  {  ShowMessage("Для данной соритировки требуется открыть табл. Студенты");  }  else if(ComboBox1->Text == "Куратору." && (CurrentUsedTable == TableGroups || CurrentUsedTable == TableSpec))  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Куратор LIKE '%"+ Edit1 -> Text +"%'";  ADOQuery1->Active = true;  }  else if(ComboBox1->Text == "Куратору." && !(CurrentUsedTable == TableGroups))  {  ShowMessage("Для данной соритировки требуется открыть табл. Группы или Специальности");  }  else if(ComboBox1->Text == "Времени обучения." && CurrentUsedTable == TableStudents)  {  ADOQuery1->Active = false;  ADOQuery1->SQL->Clear();  ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Время обучения LIKE '%"+ Edit1 -> Text +"%'";  ADOQuery1->Active = true;  }  else if(ComboBox1->Text == "Времени обучения." && !(CurrentUsedTable == TableSpec))  {  ShowMessage("Для данной соритировки требуется открыть табл. Специальности");  }  }  //---------------------------------------------------------------------------  void \_\_fastcall TMBDWindow::Button9Click(TObject \*Sender)  {  iOP.open(Path.t\_str());  iOP >> Password;  iOP.close();  if(Edit4 -> Text == Password)  {  Password = StrToInt(Edit5 -> Text);  oZP.open(Path.t\_str());  oZP << Password;  oZP.close();  DBNavigator1 -> Visible = true;  Button1 -> Visible = true;  Button2 -> Visible = true;  Button3 -> Visible = true;  Button4 -> Visible = true;  Button5 -> Visible = true;  Button6 -> Visible = true;  Button7 -> Visible = true;  DBGrid1 -> Visible = true;  Edit1 -> Visible = true;  Edit2 -> Visible = false;  Label1 -> Visible = true;  ComboBox1 -> Visible = true;  ComboBox2 -> Visible = true;  Label2 -> Visible = true;  Label4 -> Visible = true;  ochistka -> Visible = true;  Button8 -> Visible = false;  Edit3 -> Visible = false;  Button9 -> Visible = false;  Edit4 -> Visible = false;  Edit5 -> Visible = false;  }  }  //---------------------------------------------------------------------------  **MessageBoxOcenki.cpp**  //---------------------------------------------------------------------------  #include <vcl.h>  #pragma hdrstop  #include "MessageBoxOcenki.h"  #include "MainBDW.h"  //---------------------------------------------------------------------------  #pragma package(smart\_init)  #pragma resource "\*.dfm"  TMesageBox \*MesageBox;  UnicodeString TableStudents = "Студенты", TableGroups = "Группы", TableSpec = "Специальности", CurrentUsedTable;  //---------------------------------------------------------------------------  \_\_fastcall TMesageBox::TMesageBox(TComponent\* Owner)  : TForm(Owner)  {  } //ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Место\_проживания LIKE '%Раменское%'";  //---------------------------------------------------------------------------  void \_\_fastcall TMesageBox::Button1Click(TObject \*Sender)  {  MesageBox -> Visible = false;  if(RadioS5 -> Checked == true && Edit1 -> Text == TableStudents)  {  MBDWindow->ADOQuery1->Active = false;  MBDWindow->ADOQuery1->SQL->Clear();  MBDWindow->ADOQuery1->SQL->Text ="SELECT \* FROM " + TableStudents + " WHERE Оценка\_Дипломная\_Работа = 5";  MBDWindow->ADOQuery1->Active = true;  RadioS5 -> Checked = false; //Изменить в БД на Оценка(Дипломная\_работа)  }  else if (!((RadioS5 -> Checked == true || RadioS4 -> Checked == true || RadioS3 -> Checked == true)))  {  ShowMessage("Для выполнения действия требуется выбрать один из вариантов, для выхода из окна нажмите CANCEL");  }  else if(!(Edit1 -> Text == TableStudents))  {  ShowMessage("Для данного отбора требуется открыть табл." + TableStudents);  RadioS5 -> Checked = false; RadioS4 -> Checked = false; RadioS3 -> Checked = false;  }  else if(RadioS4 -> Checked == true && Edit1 -> Text == TableStudents)  {  MBDWindow->ADOQuery1->Active = false;  MBDWindow->ADOQuery1->SQL->Clear();  MBDWindow->ADOQuery1->SQL->Text ="SELECT \* FROM " + TableStudents + " WHERE Оценка\_Дипломная\_Работа = 4";  MBDWindow->ADOQuery1->Active = true;  RadioS4 -> Checked = false;  }  else if(RadioS3 -> Checked == true && Edit1 -> Text == TableStudents)  {  MBDWindow->ADOQuery1->Active = false;  MBDWindow->ADOQuery1->SQL->Clear();  MBDWindow->ADOQuery1->SQL->Text ="SELECT \* FROM " + TableStudents + " WHERE Оценка\_Дипломная\_Работа = 3";  MBDWindow->ADOQuery1->Active = true;  RadioS3 -> Checked = false;  }  }  //---------------------------------------------------------------------------  void \_\_fastcall TMesageBox::Button2Click(TObject \*Sender)  {  MesageBox -> Visible = false;  }  //---------------------------------------------------------------------------  **MessageBoxGruppi.cpp**  //---------------------------------------------------------------------------  #include <vcl.h>  #pragma hdrstop  #include "MessageBoxOcenki.h"  #include "MainBDW.h"  //---------------------------------------------------------------------------  #pragma package(smart\_init)  #pragma resource "\*.dfm"  TMesageBox \*MesageBox;  UnicodeString TableStudents = "Студенты", TableGroups = "Группы", TableSpec = "Специальности", CurrentUsedTable;  //---------------------------------------------------------------------------  \_\_fastcall TMesageBox::TMesageBox(TComponent\* Owner)  : TForm(Owner)  {  } //ADOQuery1->SQL->Text ="SELECT \* FROM " + CurrentUsedTable + " WHERE Место\_проживания LIKE '%Раменское%'";  //---------------------------------------------------------------------------  void \_\_fastcall TMesageBox::Button1Click(TObject \*Sender)  {  MesageBox -> Visible = false;  if(RadioS5 -> Checked == true && Edit1 -> Text == TableStudents)  {  MBDWindow->ADOQuery1->Active = false;  MBDWindow->ADOQuery1->SQL->Clear();  MBDWindow->ADOQuery1->SQL->Text ="SELECT \* FROM " + TableStudents + " WHERE Оценка\_Дипломная\_Работа = 5";  MBDWindow->ADOQuery1->Active = true;  RadioS5 -> Checked = false; //Изменить в БД на Оценка(Дипломная\_работа)  }  else if (!((RadioS5 -> Checked == true || RadioS4 -> Checked == true || RadioS3 -> Checked == true)))  {  ShowMessage("Для выполнения действия требуется выбрать один из вариантов, для выхода из окна нажмите CANCEL");  }  else if(!(Edit1 -> Text == TableStudents))  {  ShowMessage("Для данного отбора требуется открыть табл." + TableStudents);  RadioS5 -> Checked = false; RadioS4 -> Checked = false; RadioS3 -> Checked = false;  }  else if(RadioS4 -> Checked == true && Edit1 -> Text == TableStudents)  {  MBDWindow->ADOQuery1->Active = false;  MBDWindow->ADOQuery1->SQL->Clear();  MBDWindow->ADOQuery1->SQL->Text ="SELECT \* FROM " + TableStudents + " WHERE Оценка\_Дипломная\_Работа = 4";  MBDWindow->ADOQuery1->Active = true;  RadioS4 -> Checked = false;  }  else if(RadioS3 -> Checked == true && Edit1 -> Text == TableStudents)  {  MBDWindow->ADOQuery1->Active = false;  MBDWindow->ADOQuery1->SQL->Clear();  MBDWindow->ADOQuery1->SQL->Text ="SELECT \* FROM " + TableStudents + " WHERE Оценка\_Дипломная\_Работа = 3";  MBDWindow->ADOQuery1->Active = true;  RadioS3 -> Checked = false;  }  }  //---------------------------------------------------------------------------  void \_\_fastcall TMesageBox::Button2Click(TObject \*Sender)  {  MesageBox -> Visible = false;  }  //--------------------------------------------------------------------------- |