Федеральное государственное автономное образовательное учреждение высшего образования Российской Федерации

«Российский университет транспорта (РУТ (МИИТ))»

ИНСТИТУТ ЭКОНОМИКИ И ФИНАНСОВ

КАФЕДРА «ИНФОРМАЦИОННЫЕ СИСТЕМЫ ЦИФРОВОЙ ЭКОНОМИКИ»

**Лабораторная работа №2**

**Курсовая работа**

по дисциплине: «Объектно-ориентированный анализ и программирование»

на тему: «Ознакомление с работой базы данных и c#»

Выполнил: студент группы ЭБИ-211

Богушевич Антон Максимович

Проверил: Шационок П.В.

Москва 2021г

Оглавление

[1. Обзор средств программирования 4](#_Toc85723613)

[2. Формулировка задания 5](#_Toc85723614)

[3. Требования к программе 6](#_Toc85723615)

[4. Блок схема 7](#_Toc85723616)

[5. Скриншоты работы программы 16](#_Toc85723617)

[6. Листинг программы 17](#_Toc85723618)

Вывод, список литературы, UMl диаграмма и текст моего индивидуального заданияВведение

Курсовая работа посвящена разработке приложения, по хранению студентов в базе данных. А также их редактирование, удаление и создание.

Данная работа позволит учебным заведениям хранить своих студентов в базе данных, а также вести простую статистику по ним.

Задачи по работе с данным приложением нацелены на легкую доступность к информации о студентах, а также ее редактированию.

# Обзор средств программирования

Проект был создан на языке программирования C#. Данный проект реализовывает все данные по заданию возможности и функционал.

* 1. Обзор Visual Studio

Visual Studio - это многофункциональная программа, которая поддерживает многие аспекты разработки программного обеспечения. Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки. Для работы будет выбран язык C#.

* 1. Обзор C#

C# — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework.

* 1. Обзор SQlManager

 SQlManager - утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server.

* 1. Обзор SQLServer

 SQLServer - система управления реляционными базами данных, разработанная корпорацией Microsoft.

# Формулировка задания

Создать программу, реализующую обработку данных, введенных пользователем по следующим алгоритмам:

* 1. Запись набора введенных пользователем данных в Базу данных;
  2. Отбор данных (вводятся пользователем) с последующей сортировкой по указанным признакам (по вариантам);
  3. Вывод всех элементов базы данных;
  4. Создание связей между таблицами базы данных.

# Требования к программе

* 1. Начальная форма приложения должна содержать 9 полей для ввода данных, поле для вывода текущего значения количества записей, кнопку по которой создается новый элемент таблицы базы данных student из введенных пользователем данных и отдельную кнопку для реализации каждого действия из указанных в задании.
  2. Заголовок начальной формы – ФИО студента и номер группы, заголовок формы с выводом результата действий.
  3. Результат выполнения каждого из действий выводится на новой форме, имеющей табличное поле для вывода результата и кнопку ОК, реализующей закрытие формы. Предусмотреть блокирование начальной формы до закрытия форм с выводом результатов. На форме демонстрирующей все текущие элементы массива предусмотреть возможность редактирования и удаления записей.
  4. Для ввода полей с ограниченным набором значений (пол, основа обучения) использовать элемент comboBox, для даты рождения – dateTimePicker.
  5. Предусмотреть проверку введенных пользователем данных с генерацией сообщения об ошибке в случае необходимости. Обязательна проверка на уникальность № студенческого билета
  6. Обеспечить базовое наполнение таблицы базы данных 10 записями.

# Блок схема

Form1:







Form2:











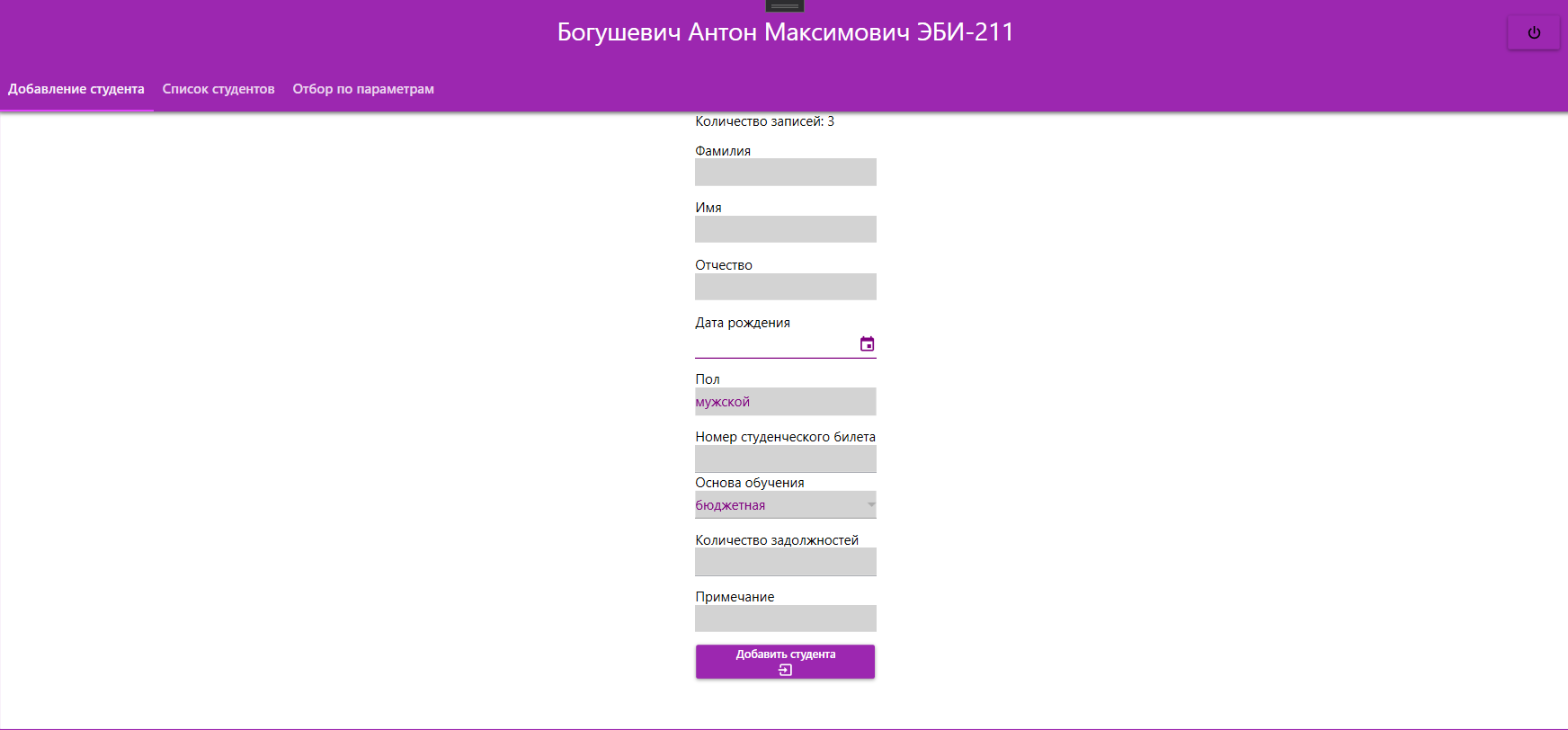
Form3:





# Скриншоты работы программы

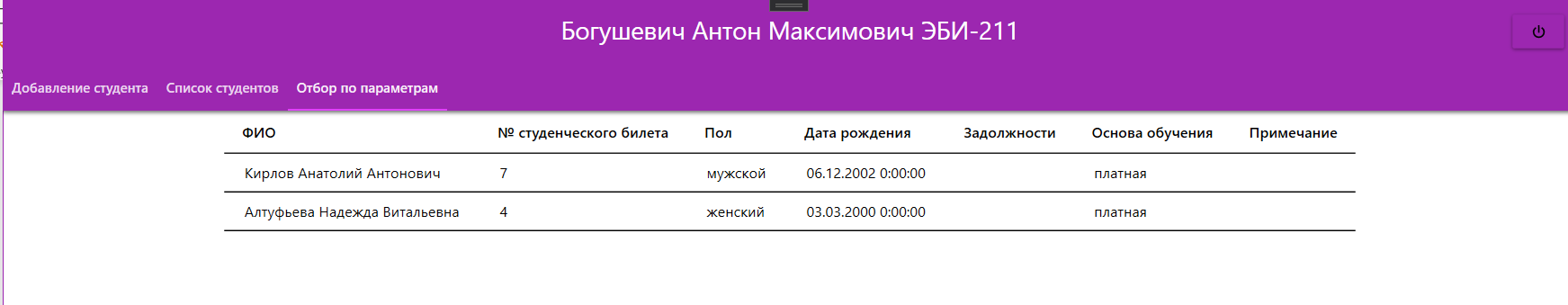
Начальная форма:



Форма обработки:



Форма отображения группы:



Скриншот базы данных:

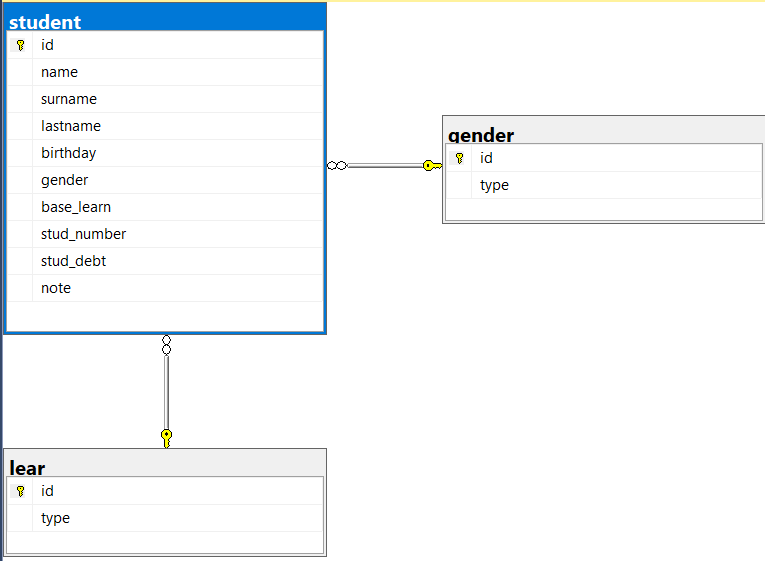


Таблица student:

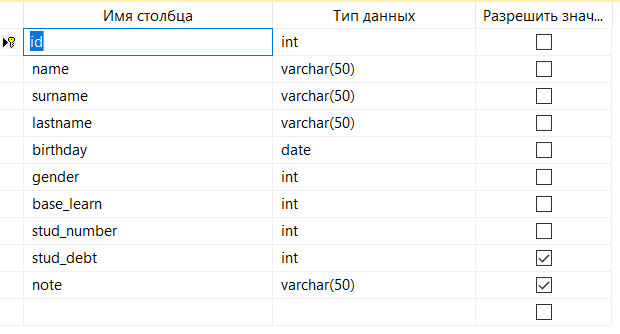


Таблица lear:

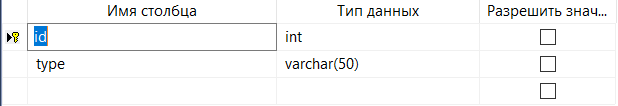
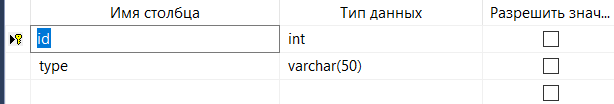


Таблица gender:



# Листинг программы

Form1.cs:

using System;

using System.Data;

using System.Drawing;

using System.Windows.Forms;

namespace Laba3

{

public partial class Form1 : Form

{

public int check = 10;

public int check\_add\_student = 10;

public Student[] student = new Student[1];

DataTable dt\_user = new DataTable();

public Form1()

{

InitializeComponent();

dateTimePicker1.MaxDate = DateTime.Now;

dateTimePicker1.Value = DateTime.Now;

sqlCon.SqlConnect connect = new sqlCon.SqlConnect();

connect.conOpen();

dt\_user = connect.select\_query("SELECT \* FROM[dbo].[student]"); // получаем данные из таблицы

connect.conClose();

label11.Text = dt\_user.Rows.Count.ToString();

comboBox1.DropDownStyle = ComboBoxStyle.DropDownList;

comboBox2.DropDownStyle = ComboBoxStyle.DropDownList;

}

public class Student

{

public string Name; // Имя

public string Surname; // Фамилия

public string LName; // Отчество

public int DateD; // Число ДР

public int DateM; // Месяц ДР

public int DateY; // Год ДР

public string Gender; // Пол

public int S\_number; // № студенческого билета

public int L\_base; // основа обучения

public string Score; // количество задолженностей

public string Note; // примечание

}

public int GoToForm3(string name)

{

Form3 f = new Form3(name);

f.Owner = this;

f.ShowDialog();

return 0;

}

public int GoToForm2(string name)

{

Form2 f = new Form2(name);

f.Owner = this;

f.ShowDialog();

return 0;

}

private void button3\_Click(object sender, EventArgs e)

{

check\_add\_student = Convert.ToInt32(label11.Text);

if (textBox1.Text == "")

{

MessageBox.Show("Введите Имя!", "ERROR!", MessageBoxButtons.OK, MessageBoxIcon.Error);

label1.BackColor = Color.Red;

}

else if (textBox2.Text == "")

{

MessageBox.Show("Введите Фамилию!", "ERROR!", MessageBoxButtons.OK, MessageBoxIcon.Error);

label2.BackColor = Color.Red;

}

else if (textBox3.Text == "")

{

MessageBox.Show("Введите Отчество!", "ERROR!", MessageBoxButtons.OK, MessageBoxIcon.Error);

label3.BackColor = Color.Red;

}

else

{

label1.BackColor = SystemColors.Control;

label2.BackColor = SystemColors.Control;

label3.BackColor = SystemColors.Control;

label8.BackColor = SystemColors.Control;

label6.BackColor = SystemColors.Control;

try

{

int s\_nb = Convert.ToInt32(textBox4.Text); //студ. номер

int z\_s = 0;

if (textBox5.Text != "")

{

z\_s = Convert.ToInt32(textBox5.Text); //количество задолжностей

}

sqlCon.SqlConnect connect = new sqlCon.SqlConnect();

connect.conOpen();

dt\_user = connect.select\_query("SELECT \* FROM[dbo].[student] WHERE stud\_number = '" + s\_nb.ToString() + "'"); // получаем данные из таблицы

connect.conClose();

if (dt\_user.Rows.Count > 0)

{

MessageBox.Show("Номер студ. билета уже есть в базе!", "ERROR!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

connect = new sqlCon.SqlConnect();

connect.conOpen();

int key = comboBox1.SelectedIndex + 1;

student[0] = new Student

{

Name = textBox1.Text,

Surname = textBox2.Text,

LName = textBox3.Text,

DateD = dateTimePicker1.Value.Day,

DateM = dateTimePicker1.Value.Month,

DateY = dateTimePicker1.Value.Year,

Gender = Convert.ToString(comboBox1.SelectedIndex + 1),

S\_number = s\_nb,

L\_base = comboBox2.SelectedIndex + 1,

Score = Convert.ToString(z\_s),

Note = textBox6.Text

};

int check = connect.insert\_command\_student(student); // получаем данные из таблицы

dt\_user = connect.select\_query("SELECT \* FROM[dbo].[student]"); // получаем данные из таблицы

label11.Text = dt\_user.Rows.Count.ToString();

connect.conClose();

}

}

catch (Exception y)

{

MessageBox.Show("Номер студ. билета или количество задолжностей введено не верно", "ERROR!", MessageBoxButtons.OK, MessageBoxIcon.Error);

label8.BackColor = Color.Red;

label6.BackColor = Color.Red;

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

string name = "Список платников, с сортировкой по дате рождения (убывания)";

GoToForm3(name);

}

private void button1\_Click(object sender, EventArgs e)

{

string name = "Список группы";

GoToForm2(name);

}

}

}

Form2.cs:

using System;

using System.Collections.Generic;

using System.Data;

using System.Windows.Forms;

namespace Laba3

{

public partial class Form2 : Form

{

public class DB\_ID

{

public int id;

}

public DataTable dt\_user;

public Form2(string name)

{

InitializeComponent();

sqlCon.SqlConnect connect = new sqlCon.SqlConnect();

connect.conOpen();

dt\_user = connect.select\_query("SELECT \* FROM[dbo].[student] AS st LEFT JOIN[dbo].[gender] AS gr ON(st.gender = gr.id) LEFT JOIN[dbo].[lear] ler ON(st.base\_learn = ler.id)"); // получаем данные из таблицы

connect.conClose();

this.Text = name;

dataGridView1.RowCount = dt\_user.Rows.Count;

dataGridView1.ColumnCount = 7;

dataGridView1.Columns[0].Name = "ФИО";

dataGridView1.Columns[1].Name = "№ студенческого билета";

dataGridView1.Columns[2].Name = "пол";

dataGridView1.Columns[3].Name = "Дата рожджения";

dataGridView1.Columns[4].Name = "Задолжности";

dataGridView1.Columns[5].Name = "Основа обучения";

dataGridView1.Columns[6].Name = "Примечание";

for (int i = 0; i < dt\_user.Rows.Count; i++)

{

dataGridView1.Rows[i].Cells[0].Value = dt\_user.Rows[i][2] + " " + dt\_user.Rows[i][1] + " " + dt\_user.Rows[i][3];

dataGridView1.Rows[i].Cells[1].Value = dt\_user.Rows[i][7];

dataGridView1.Rows[i].Cells[2].Value = dt\_user.Rows[i][11];

dataGridView1.Rows[i].Cells[3].Value = dt\_user.Rows[i][4];

dataGridView1.Rows[i].Cells[4].Value = dt\_user.Rows[i][8];

dataGridView1.Rows[i].Cells[5].Value = dt\_user.Rows[i][13];

dataGridView1.Rows[i].Cells[6].Value = dt\_user.Rows[i][9];

}

foreach (DataGridViewColumn column in dataGridView1.Columns)

{

column.SortMode = DataGridViewColumnSortMode.NotSortable;

}

}

private void button1\_Click\_1(object sender, EventArgs e)

{

Form2.ActiveForm.Close();

}

private void button2\_Click(object sender, EventArgs e)

{

Form1 form1 = this.Owner as Form1;

sqlCon.SqlConnect connect = new sqlCon.SqlConnect();

connect.conOpen();

dt\_user = connect.select\_query("SELECT \* FROM[dbo].[student]"); // получаем данные из таблицы

connect.conClose();

DB\_ID[] ids = new DB\_ID[dt\_user.Rows.Count];

for (int i = 0; i < dt\_user.Rows.Count; i++)

{

ids[i] = new DB\_ID

{

id = Convert.ToInt32(dt\_user.Rows[i][0])

};

}

DataGridViewSelectedRowCollection rows\_del = dataGridView1.SelectedRows;

connect.conOpen();

foreach (DataGridViewRow row in rows\_del)

{

int check\_1 = row.Index;

connect.delete\_command\_studentById(ids[row.Index].id);

dataGridView1.Rows.RemoveAt(row.Index);

List<DB\_ID> tmp = new List<DB\_ID>(ids);

tmp.RemoveAt(check\_1);

ids = tmp.ToArray();

form1.label11.Text = Convert.ToString(Convert.ToInt32(form1.label11.Text) - 1);

}

connect.conClose();

}

private void button3\_Click(object sender, EventArgs e)

{

Form1 form1 = this.Owner as Form1;

sqlCon.SqlConnect connect = new sqlCon.SqlConnect();

connect.conOpen();

dt\_user = connect.select\_query("SELECT \* FROM[dbo].[student]");

connect.delete\_command\_All();

for (int i = 0; i < dt\_user.Rows.Count; i++)

{

string[] fullName = Convert.ToString(dataGridView1.Rows[i].Cells[0].Value).Split(' ');

DateTime date\_full = Convert.ToDateTime(dataGridView1.Rows[i].Cells[3].Value);

int check\_l\_base = 0;

if (Convert.ToString(dataGridView1.Rows[i].Cells[5].Value) == "бюджетная")

{

check\_l\_base = 1;

}

else

{

check\_l\_base = 2;

}

int check\_gender = 0;

if (Convert.ToString(dataGridView1.Rows[i].Cells[2].Value) == "мужской")

{

check\_gender = 1;

}

else

{

check\_gender = 2;

}

form1.student[0] = new Form1.Student

{

Name = Convert.ToString(fullName[1]),

Surname = Convert.ToString(fullName[0]),

LName = Convert.ToString(fullName[2]),

DateD = Convert.ToInt32(date\_full.Day),

DateM = Convert.ToInt32(date\_full.Month),

DateY = Convert.ToInt32(date\_full.Year),

Gender = Convert.ToString(check\_gender),

S\_number = Convert.ToInt32(dataGridView1.Rows[i].Cells[1].Value),

L\_base = check\_l\_base,

Score = Convert.ToString(dataGridView1.Rows[i].Cells[4].Value),

Note = Convert.ToString(dataGridView1.Rows[i].Cells[6].Value)

};

int check = connect.insert\_command\_student(form1.student);

}

connect.conClose();

}

}

}

Form3.cs

using System;

using System.Data;

using System.Linq;

using System.Windows.Forms;

namespace Laba3

{

public partial class Form3 : Form

{

public class id\_date

{

public string FullName;

public DateTime date1;

public int studnumber;

public string L\_base;

public string Score;

public string Note;

public string Gender;

}

public Form3(string name)

{

InitializeComponent();

this.Text = name;

sqlCon.SqlConnect connect = new sqlCon.SqlConnect();

connect.conOpen();

DataTable dt\_user = connect.select\_query("SELECT \* FROM[dbo].[student] AS st INNER JOIN[dbo].[gender] AS gr ON(st.gender = gr.id) INNER JOIN[dbo].[lear] ler ON(st.base\_learn = ler.id) where ler.type = 'платная'"); // получаем данные из таблицы

connect.conClose();

dataGridView1.RowCount = dt\_user.Rows.Count;

id\_date[] idanddate = new id\_date[dt\_user.Rows.Count];

dataGridView1.ColumnCount = 7;

dataGridView1.Columns[0].Name = "ФИО";

dataGridView1.Columns[0].Name = "ФИО";

dataGridView1.Columns[1].Name = "№ студенческого билета";

dataGridView1.Columns[2].Name = "пол";

dataGridView1.Columns[3].Name = "Дата рожджения";

dataGridView1.Columns[4].Name = "Задолжности";

dataGridView1.Columns[5].Name = "Основа обучения";

dataGridView1.Columns[6].Name = "Примечание";

int add\_in\_datagried = 0;

for (int i = 0; i < dt\_user.Rows.Count; i++)

{

idanddate[i] = new id\_date

{

FullName = dt\_user.Rows[i][2] + " " + dt\_user.Rows[i][1] + " " + dt\_user.Rows[i][3],

date1 = Convert.ToDateTime(dt\_user.Rows[i][4]),

studnumber = Convert.ToInt32(dt\_user.Rows[i][7]),

Gender = Convert.ToString(dt\_user.Rows[i][11]),

Score = Convert.ToString(dt\_user.Rows[i][8]),

L\_base = Convert.ToString(dt\_user.Rows[i][13]),

Note = Convert.ToString(dt\_user.Rows[i][9])

};

}

var list = idanddate.OrderByDescending(x => x.date1).ToList();

for (int i = 0; i < dt\_user.Rows.Count; i++)

{

dataGridView1.Rows[i].Cells[0].Value = list[i].FullName;

dataGridView1.Rows[i].Cells[1].Value = list[i].studnumber;

dataGridView1.Rows[i].Cells[2].Value = list[i].Gender;

dataGridView1.Rows[i].Cells[3].Value = list[i].date1;

dataGridView1.Rows[i].Cells[4].Value = list[i].Score;

dataGridView1.Rows[i].Cells[5].Value = list[i].L\_base;

dataGridView1.Rows[i].Cells[6].Value = list[i].Note;

}

foreach (DataGridViewColumn column in dataGridView1.Columns)

{

column.SortMode = DataGridViewColumnSortMode.NotSortable;

}

}

private void button1\_Click(object sender, EventArgs e)

{

Form3.ActiveForm.Close();

}

}

}

SqlConnect.cs:

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace Laba3.sqlCon

{

public class SqlConnect

{

public SqlConnection Con { get; set; }//the object

private string conString { get; set; }//the string to store your connection parameters

public void conOpen()

{

conString = "Data Source=.\\NEMOOO;Initial Catalog=students;Integrated Security=True"; //the same as you post in your post

Con = new SqlConnection(conString);//

try

{

Con.Open();//try to open the connection

}

catch (Exception ex)

{

MessageBox.Show("Ошибка подключения к бд!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

public void conClose()

{

Con.Close();//close the connection

}

public DataTable select\_query(string query)

{

DataTable dataTable = new DataTable("dataBase");

SqlCommand sqlCommand = Con.CreateCommand(); // создаём команду

sqlCommand.CommandText = query;

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand); // создаём обработчик

sqlDataAdapter.Fill(dataTable); // возращаем таблицу с результатом

return dataTable;

}

public int insert\_command\_student(Form1.Student[] student)

{

string key = student[0].Name;

string stmt = "INSERT INTO [dbo].[student] ([name],[surname],[lastname],[birthday],[gender],[base\_learn],[stud\_number],[stud\_debt],[note]) VALUES (@Name , @Surname, @Lastname, @Birthday, @Gender, @Base\_Learn, @Stud\_Number, @Stud\_Debt, @Note)";

SqlCommand cmd = new SqlCommand(stmt, Con);

cmd.Parameters.Add("@Name", SqlDbType.VarChar, 50);

cmd.Parameters.Add("@Surname", SqlDbType.VarChar, 50);

cmd.Parameters.Add("@Lastname", SqlDbType.VarChar, 50);

cmd.Parameters.Add("@Birthday", SqlDbType.Date);

cmd.Parameters.Add("@Gender", SqlDbType.Int);

cmd.Parameters.Add("@Base\_Learn", SqlDbType.Int);

cmd.Parameters.Add("@Stud\_Number", SqlDbType.Int);

cmd.Parameters.Add("@Stud\_Debt", SqlDbType.Int);

cmd.Parameters.Add("@Note", SqlDbType.VarChar, 50);

cmd.Parameters["@Name"].Value = student[0].Name;

cmd.Parameters["@Surname"].Value = student[0].Surname;

cmd.Parameters["@Lastname"].Value = student[0].LName;

string check\_date = Convert.ToString(student[0].DateD.ToString() + '.' + student[0].DateM.ToString() + '.' + student[0].DateY.ToString());

cmd.Parameters["@Birthday"].Value = check\_date;

cmd.Parameters["@Gender"].Value = student[0].Gender;

cmd.Parameters["@Base\_Learn"].Value = student[0].L\_base;

cmd.Parameters["@Stud\_Number"].Value = student[0].S\_number;

if (student[0].Score == "")

{

cmd.Parameters["@Stud\_Debt"].Value = DBNull.Value;

}

else { cmd.Parameters["@Stud\_Debt"].Value = student[0].Score; }

cmd.Parameters["@Note"].Value = student[0].Note;

cmd.ExecuteNonQuery();

return 0;

}

public void delete\_command\_studentById(int id)

{

string stmt = "DELETE FROM [dbo].[student] WHERE id = " + id;

SqlCommand cmd = new SqlCommand(stmt, Con);

cmd.ExecuteNonQuery();

}

public void delete\_command\_All()

{

string stmt = "DELETE FROM [dbo].[student]";

SqlCommand cmd = new SqlCommand(stmt, Con);

cmd.ExecuteNonQuery();

}

}

}