Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

УП 01.01

Выполнил: Деспижек Б.К.

Группа: ПР-21

Преподаватель: Мирошниченко Г.В.

2023

Оглавление

[1. Тема «Текстовые файлы» 2](#_Toc138210877)

[1.1 Постановка задачи 2](#_Toc138210878)

[1.2 Код программы 6](#_Toc138210879)

[2. Тема «Графика и рекурсия» 8](#_Toc138210880)

[2.1 Графика 8](#_Toc138210881)

[2.1.1 Задание 0.1 8](#_Toc138210882)

[2.1.2 Задание 0.2 9](#_Toc138210883)

[2.1.3 Задание 0.3 10](#_Toc138210884)

[2.1.4 Задание 1 11](#_Toc138210885)

[2.1.5 Задание 2 12](#_Toc138210886)

[2.1.6 Задание 3 13](#_Toc138210887)

[2.1.7 Задание 5 15](#_Toc138210888)

[2.1.8 Задание 6 16](#_Toc138210889)

[2.1.9 Индивидуальное задание 17](#_Toc138210890)

[3. Тема «Проект модульного тестирования» 19](#_Toc138210891)

[Задача 10 «Быки и коровы». 19](#_Toc138210893)

[3.1 План тестирования 19](#_Toc138210900)

[3.2 Программа 30](#_Toc138210901)

[4. Тема «Практика 10-11. Работа с базой данных» 34](#_Toc138210902)

[4.1 Таблицы в MS SQL Server 34](#_Toc138210903)

[4.2 Задания и их реализация 34](#_Toc138210904)

[5 Тема «Практика 12. Работа с базой данных» 40](#_Toc138210905)

[5.1 Таблицы в MS SQL Server в режиме проекта 40](#_Toc138210906)

[5.2 Связи между таблицами в MS SQL Server 42](#_Toc138210907)

[5.3 Таблицы в режиме просмотра данных в MS SQL Server 42](#_Toc138210908)

[5.4 Внешний вид формы 44](#_Toc138210909)

[5.5 Методы добавления данных в таблицу БД 46](#_Toc138210910)

[4.1 Методы, реализующие поиск по БД 52](#_Toc138210911)

[4.1 Методы обновления всех DataGrid 53](#_Toc138210912)

# Тема «Текстовые файлы»

## Постановка задачи

9) удалить все скобки и символы, заключенные между ними;

Входные данные:

файл из которого читается информация

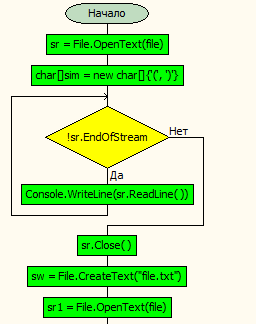
строки из файла

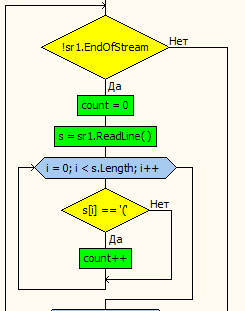
Выходные данные:

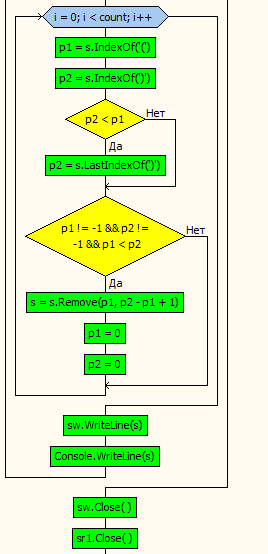
новый файл

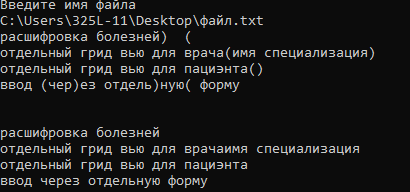
измененные строки записанные в новом файле

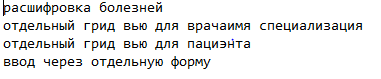
Блок-Схема:











## Код программы

using System.IO;

Console.WriteLine("Введите имя файла");

string file = Console.ReadLine();

if (File.Exists(file))

{

StreamReader sr = File.OpenText(file);

char[] sim = new char[] { '(', ')' };

while (!sr.EndOfStream)

{

Console.WriteLine(sr.ReadLine());

}

sr.Close();

StreamWriter sw = File.CreateText("file.txt");

StreamReader sr1 = File.OpenText(file);

Console.WriteLine();

while (!sr1.EndOfStream)

{

int count = 0;

string s = sr1.ReadLine();

for (int i = 0; i < s.Length; i++)

{

if (s[i] == '(')

{

count++;

}

}

for (int i = 0; i < count; i++)

{

int p1 = s.IndexOf('(');

int p2 = s.IndexOf(')');

if(p2 < p1)

{

p2=s.LastIndexOf(')');

}

if (p1 != -1 && p2 != -1 && p1 < p2)

{

s = s.Remove(p1, p2 - p1 + 1);

p1 = 0;

p2 = 0;

}

}

sw.WriteLine(s);

Console.WriteLine(s);

}

sw.Close();

sr1.Close();

Console.WriteLine();

StreamReader sr2 = File.OpenText("file.txt");

StreamWriter sw1 = File.CreateText("file1.txt");

while (!sr2.EndOfStream)

{

string s = sr2.ReadLine();

for (int i = 0; i < sim.Length; i++)

{

s = s.Replace($"{sim[i]}", "");

}

Console.WriteLine(s);

sw1.WriteLine(s);

}

sw1.Close();

sr2.Close();

}

else

{

Console.WriteLine("Такого файла не существует");

# Тема «Графика и рекурсия»

## Графика

### Задание 0.1

Задание: построить не закрашенные фигуры для каждого пункта меню: круг, прямоугольник, квадрат, треугольник, трапеция, шестиугольник, линия.

Код методов:

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

{

Graphics g = this.CreateGraphics();

Point[] tt = { new Point(30, 100), new Point(100, 100), new Point(150, 150), new Point(100, 200), new Point(30, 200), new Point(5, 150) };

g.DrawPolygon(purplePen, tt);

g.Dispose();

}

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

{

Graphics g = this.CreateGraphics();

for (int i = 20; i < 4 \* 20; i = i + 20)

{

g.DrawRectangle(redPen, i, i \* 2, i \* 3, i \* 4);

}

g.Dispose();

}

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

{

Graphics g = this.CreateGraphics();

for (int i = 20; i < 4 \* 20; i = i + 20)

{

g.DrawEllipse(bluePen, i, i \* 2, i \* 3, i \* 4);

}

g.Dispose();

}

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

{

Graphics g = this.CreateGraphics();

g.DrawLine(redPen, 200, 100, 240, 340);

g.Dispose();

}

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

{

Graphics g = this.CreateGraphics();

Point[] tt = { new Point(50, 100), new Point(50, 200), new Point(150, 150) };

g.DrawPolygon(greenPen, tt);

g.Dispose();

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

{

Graphics g = this.CreateGraphics();

Point[] tt = { new Point(30, 100), new Point(100, 100), new Point(150, 200), new Point(5, 200) };

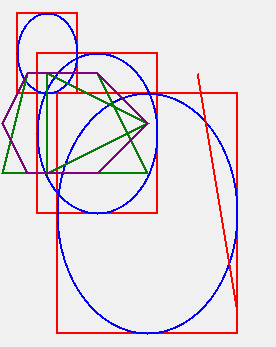
g.DrawPolygon(greenPen, tt);

g.Dispose();

}

}

Результат:



### Задание 0.2

Задание: построить закрашенные фигуры.

Код методов:

private void прямоугольникToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Graphics g = this.CreateGraphics();

g.FillRectangle(Brushes.HotPink, 100, 100, 500, 500);

g.FillRectangle(br, 200, 200, 500, 500);

g.Dispose();

}

private void кругToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Graphics g = this.CreateGraphics();

g.FillEllipse(Brushes.Aqua, 150, 150, 50, 50);

g.Dispose();

}

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

{

Graphics g = this.CreateGraphics();

Point[] tt = { new Point(150, 100), new Point(150, 200), new Point(250, 150) };

g.FillPolygon(Brushes.Crimson, tt);

g.Dispose();

}

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

{

Graphics g = this.CreateGraphics();

Point[] tt = { new Point(30, 100), new Point(100, 100), new Point(150, 150), new Point(100, 200), new Point(30, 200), new Point(5, 150) };

g.FillPolygon(Brushes.Tan, tt);

g.Dispose();

}

private void трапецияToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Graphics g = this.CreateGraphics();

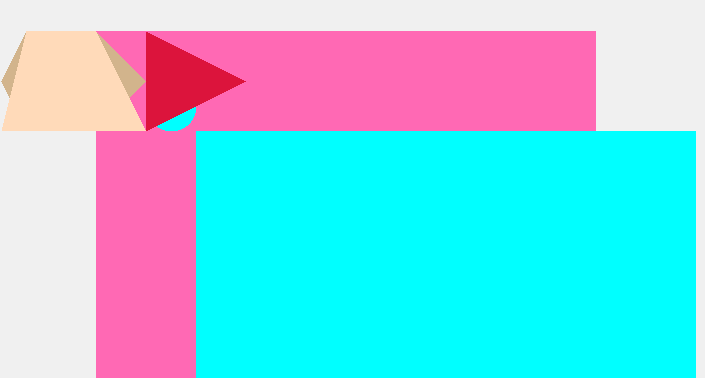
Point[] tt = { new Point(30, 100), new Point(100, 100), new Point(150, 200), new Point(5, 200) };

g.FillPolygon(Brushes.PeachPuff, tt);

g.Dispose();

}

Результат:



### Задание 0.3

Задание: нарисовать несколько окружностей и прямоугольников, вложенных друг в друга, используя цикл (общий центр или общий угол).

Код метода:

private void задание1ToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Graphics g = this.CreateGraphics();

int kx = this.Width/2;

int ky = this.Height/2;

MyDraw1(g, 7, kx, ky, 50, 50);

}

private void задание2ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Graphics g = this.CreateGraphics();

int kx = this.Width / 2;

int ky = this.Height / 2;

MyDraw2(g, 7, kx, ky, 50, 50);

}

private void задание3ToolStripMenuItem\_Click(object sender, EventArgs e)

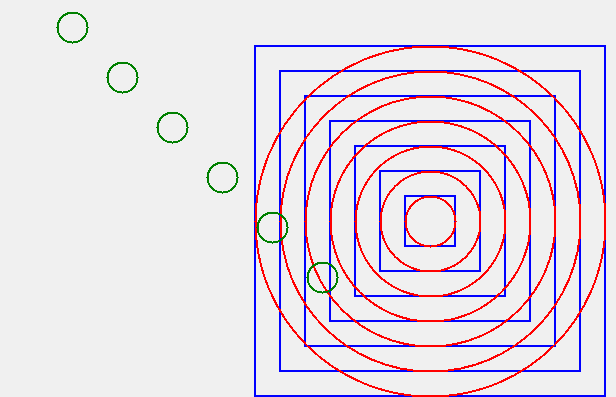
{

Graphics g = this.CreateGraphics();

MyDraw3(g, 7, 10, 10, 30, 30);

}

Результат:



### Задание 1

Задание: нарисовать снеговика и елку (три закрашенных шара, глаза, нос морковкой, на голове колпак, ёлка - три треугольника).

Код метода:

// задание 1 (снеговик)

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

{

Graphics g = this.CreateGraphics();

Point[] tt = { new Point(340, 50), new Point(380, 50), new Point(405, 100), new Point(315, 100) };

g.FillPolygon(Brushes.DarkGray, tt);

g.FillEllipse(Brushes.Snow, 320, 101, 80, 80);

g.FillEllipse(Brushes.Snow, 300, 181, 120, 120);

g.FillEllipse(Brushes.Snow, 280, 301, 160, 160);

Point[] mor = { new Point(360, 141), new Point(380, 145), new Point(360, 150) };

g.FillEllipse(Brushes.Black, 340, 130, 10, 10);

g.FillEllipse(Brushes.Black, 380, 130, 10, 10);

g.FillPolygon(Brushes.Orange, mor);

Point[] elka = { new Point(610, 70), new Point(560, 130), new Point(660, 130) };

g.FillPolygon(Brushes.DarkGreen, elka);

Point[] elka1 = { new Point(610, 130), new Point(540, 210), new Point(680, 210) };

g.FillPolygon(Brushes.DarkGreen, elka1);

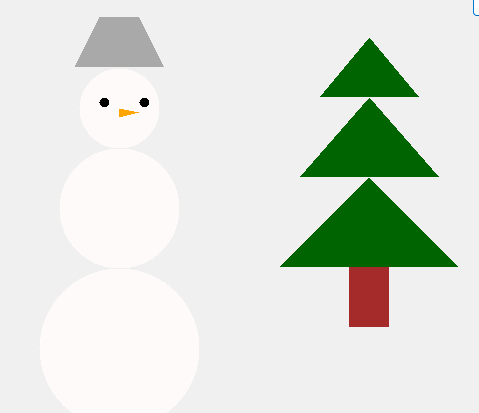
Point[] elka2 = { new Point(610, 210), new Point(520, 300), new Point(700, 300) };

g.FillPolygon(Brushes.DarkGreen, elka2);

g.FillRectangle(Brushes.Brown, 590, 300, 40, 60);

g.Dispose();

}Результат:



### Задание 2

Задание: сделать эффект летящего круга, который, долетая до любого края формы или любой другой фигуры, отлетает от нее.

Код метода:

// задание 2

private void задание2летящийШарToolStripMenuItem\_Click(object sender, EventArgs e)

{

timer1.Interval = 40;

timer1.Start();

}

private int x = 0;

private int y = 36;

private int dx = 5;

private int dy = 5;

private void timer1\_Tick(object sender, EventArgs e)

{

Draw();

x += dx;

y += dy;

if (x + 50 >= ClientSize.Width || x <= 0) { dx = -dx; }

if (y + 50 >= ClientSize.Height || y <= 36) { dy = -dy; }

Thread.Sleep(30);

Invalidate();

}

private void Draw()

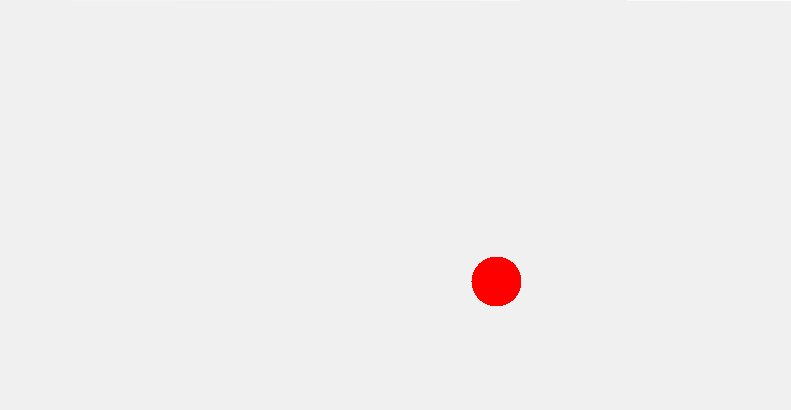
{

g = this.CreateGraphics();

g.FillEllipse(Brushes.Red, x, y, 50, 50);

}

Результат:



### Задание 3

Задание: нарисовать график функций y=x^3 или y=x^2 так, что центр координат для него в центре формы.

Задание: попробовать построить sin(x) и cos(x); учесть, что для перевода в градусы нужно константу Пи поделить на 180 и умножить на угол в радианах.

Код метода:

// 3 задание

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

{

double x0, y0, x1, y1, x2, y2;

Graphics g = this.CreateGraphics();

x0 = this.Width / 2;

y0 = this.Height / 2;

Point[] osx = { new Point(0, this.Height / 2), new Point(this.Width, this.Height / 2) };

g.DrawLines(blackPen, osx);

Point[] osy = { new Point(this.Width / 2, 0), new Point(this.Width / 2, this.Height) };

g.DrawLines(blackPen, osy);

x1 = -40;

y1 = x1 \* x1 \* x1;

for (int x = -40; x < 40; x += 1)

{

int i3 = x;

int i4 = (i3 \* i3 \* i3) / 50;

g.DrawLine(redPen, (float)(x1 + x0), (float)(y1 + y0), (float)(i3 + x0), (float)(i4 + y0));

x1 = i3;

y1 = i4;

}

x2 = -40;

y2 = x2 \* x2;

for (int x = -40; x < 40; x += 1)

{

int i3 = x;

int i4 = (i3 \* i3) / 5;

g.DrawLine(bluePen, (float)(x2 + x0), (float)(-y2 + y0), (float)(i3 + x0), (float)(-i4 + y0));

x2 = i3;

y2 = i4;

}

x2 = -40 \* 50;

y2 = Math.Sin(x2) \* 200;

for (int x = -40; x < 40; x += 1)

{

int i3 = x \* 50;

double i4 = Math.Sin(i3) \* 200;

g.DrawLine(purplePen, (float)(x2 + x0), (float)(y2 + y0), (float)(i3 + x0), (float)(i4 + y0));

x2 = i3;

y2 = i4;

}

x2 = -40 \* 50;

y2 = Math.Cos(x2) \* 200;

for (int x = -40; x < 40; x += 1)

{

int i3 = x \* 50;

double i4 = Math.Cos(i3) \* 200;

g.DrawLine(greenPen, (float)(x2 + x0), (float)(y2 + y0), (float)(i3 + x0), (float)(i4 + y0));

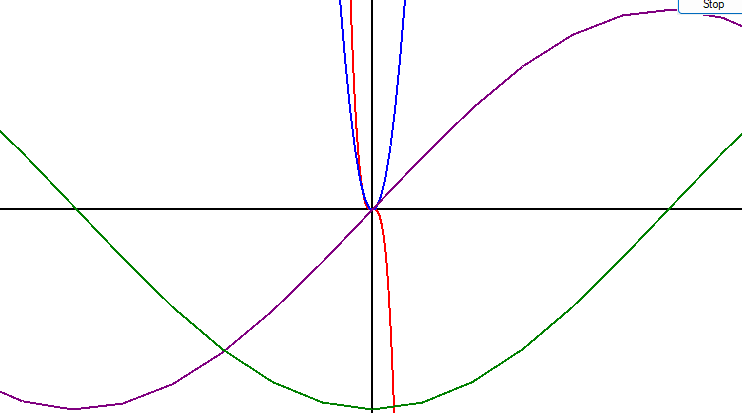
x2 = i3;

y2 = i4;

}

}

Результат:



### Задание 5

Задание: при движении и нажатии на кнопку мыши, рисуется линия. Самостоятельно разобраться с событиями перемещения мыши по форме, MouseMove, MouseDown - нажата клавиша мыши.

Код методов:

// 5 задание

private void Form1\_MouseDown(object sender, MouseEventArgs e)

{

starpoint = e.Location;

}

private void Form1\_MouseMove(object sender, MouseEventArgs e)

{

Graphics g = this.CreateGraphics();

if (e.Button == MouseButtons.Left)

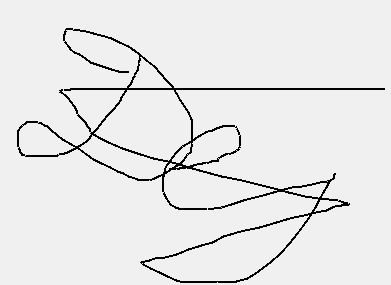
{

g.DrawLine(blackPen, starpoint, e.Location);

starpoint = e.Location;

}

Результат:



### Задание 6

Задание: делать эффект дождя или мерцающего звездного неба. (случайным образом появляются и исчезают фигуры).

Код метода:

// 6 задание (дождь)

private Random random;

private void задание6ЭффектДождяToolStripMenuItem\_Click(object sender, EventArgs e)

{

timer2.Interval = 2;

random = new Random();

timer2.Start();

}

private void timer2\_Tick(object sender, EventArgs e)

{

int xR = random.Next(0, ClientSize.Width);

int yR = random.Next(0, ClientSize.Height);

int widthR = random.Next(30, 35);

int heightR = random.Next(30, 35);

int x = random.Next(0, ClientSize.Width);

int y = random.Next(0, ClientSize.Height);

int width = random.Next(10, 15);

int height = random.Next(10, 15);

g = CreateGraphics();

g.FillEllipse(Brushes.WhiteSmoke, xR, yR, widthR, heightR);

g.FillEllipse(Brushes.Blue, x, y, width, height);

}

Результат:



### Индивидуальное задание

Задание: Разработайте программу построения треугольника Серпинского.

Код методов:

// треугольник Серпинского

private void задание4ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Graphics g = this.CreateGraphics();

MyDraw11(g);

g.Dispose();

}

private void MyDraw11(Graphics g)

{

g = CreateGraphics();

Point p1 = new Point(ClientSize.Width / 2, 50);

Point p2 = new Point(ClientSize.Width / 4, ClientSize.Height - 20);

Point p3 = new Point(ClientSize.Width / 2 + ClientSize.Width / 4, ClientSize.Height - 20);

DrawSierpinski(g, 7, p1, p2, p3);

}

private void DrawSierpinski(Graphics g, int level, Point p1, Point p2, Point p3)

{

if (level == 0)

{

Point[] points = { p1, p2, p3 };

using (Pen pen = new Pen(Color.Chocolate, 2))

g.DrawPolygon(pen, points);

}

else

{

Point p4 = new Point((p1.X + p2.X) / 2, (p1.Y + p2.Y) / 2);

Point p5 = new Point((p1.X + p3.X) / 2, (p1.Y + p3.Y) / 2);

Point p6 = new Point((p2.X + p3.X) / 2, (p2.Y + p3.Y) / 2);

DrawSierpinski(g, level - 1, p1, p4, p5);

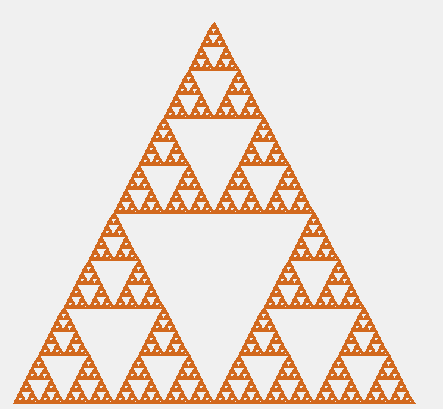
DrawSierpinski(g, level - 1, p4, p2, p6);

DrawSierpinski(g, level - 1, p5, p6, p3);

}

}

Результаты:



# Тема «Проект модульного тестирования»

## Постановка задачи:

## Задача 10 «Быки и коровы».

## Цель программы: получить автомат по ПТПМ.

## Входные данные:

## Х-целое четырехзначное число, в котором цифры не повторяются

## Выходные данные:

## Сообщения о текущем состоянии игры (Кол-во быков и коров),

## «Вы победили!», «Введите четырехзначное число», «Введите положительное число», «Введите число с неповторяющимися цифрами», «Неправильный ввод»

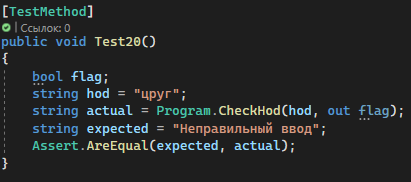
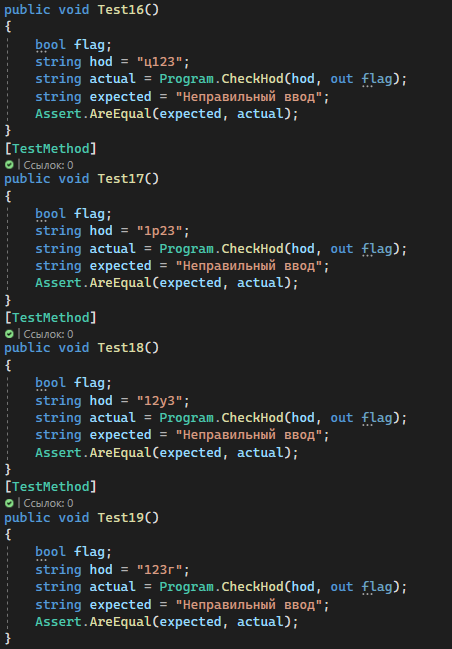
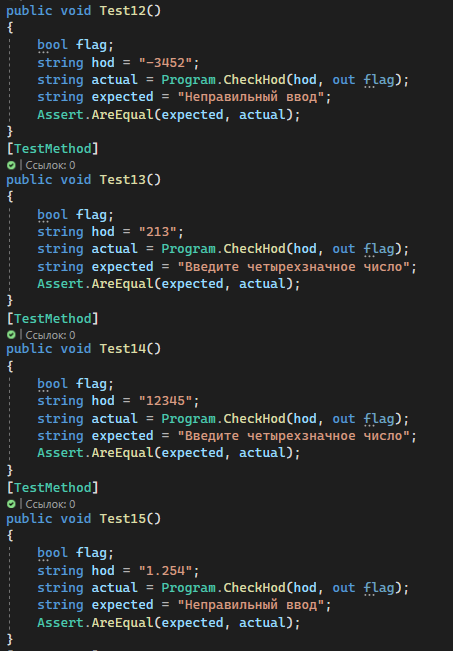
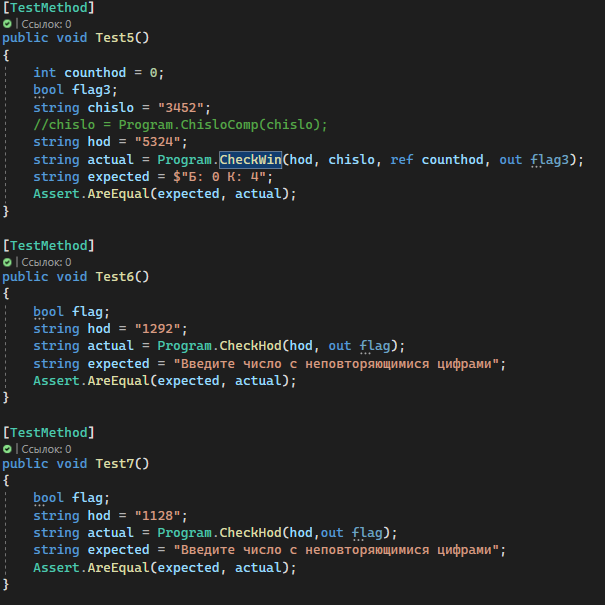
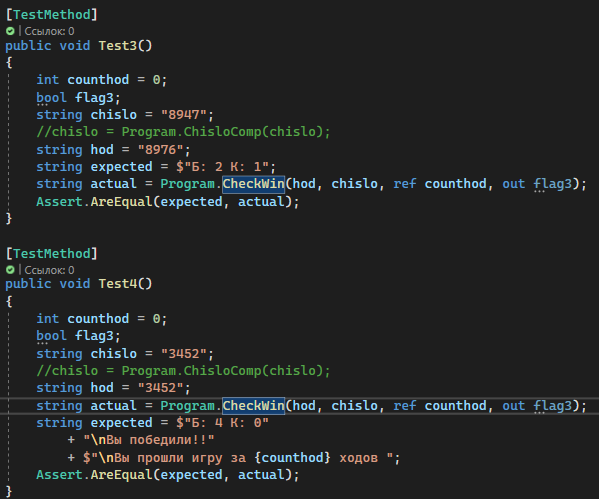
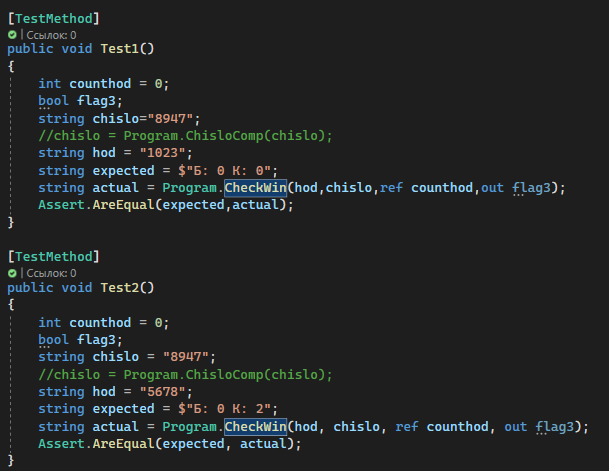
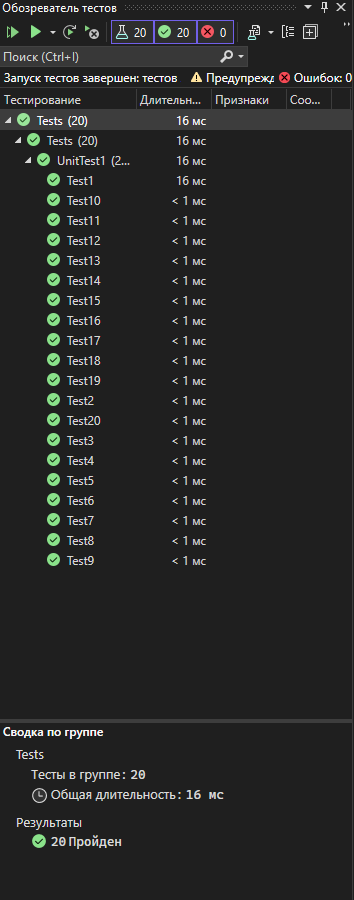
## План тестирования

Классы эквивалентности по входным данным и граничным значениям:

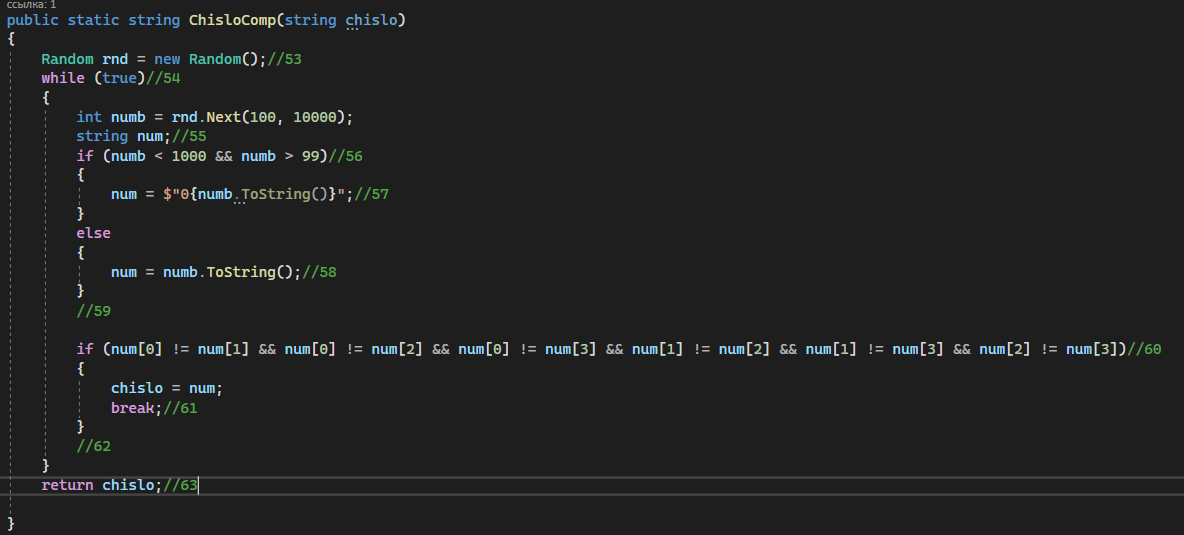
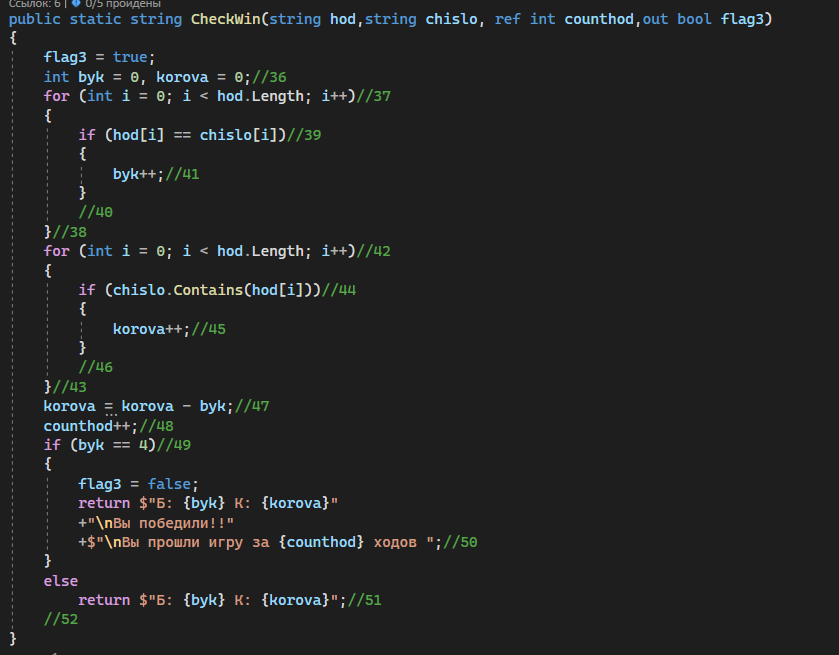
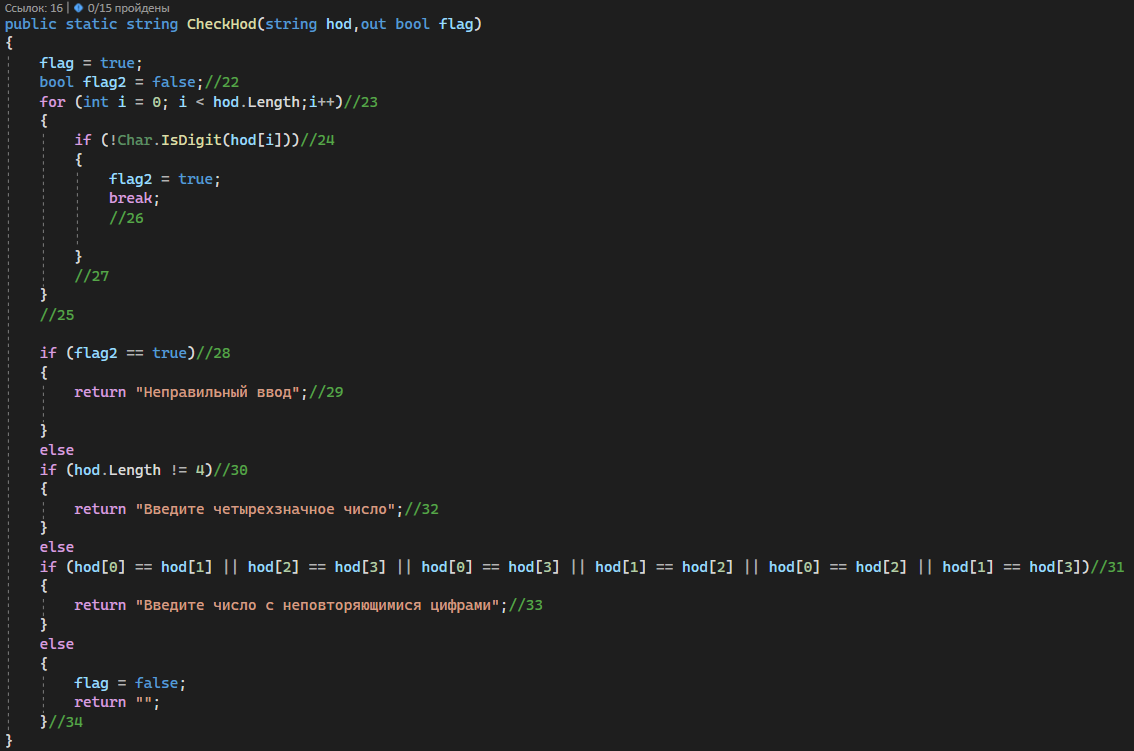
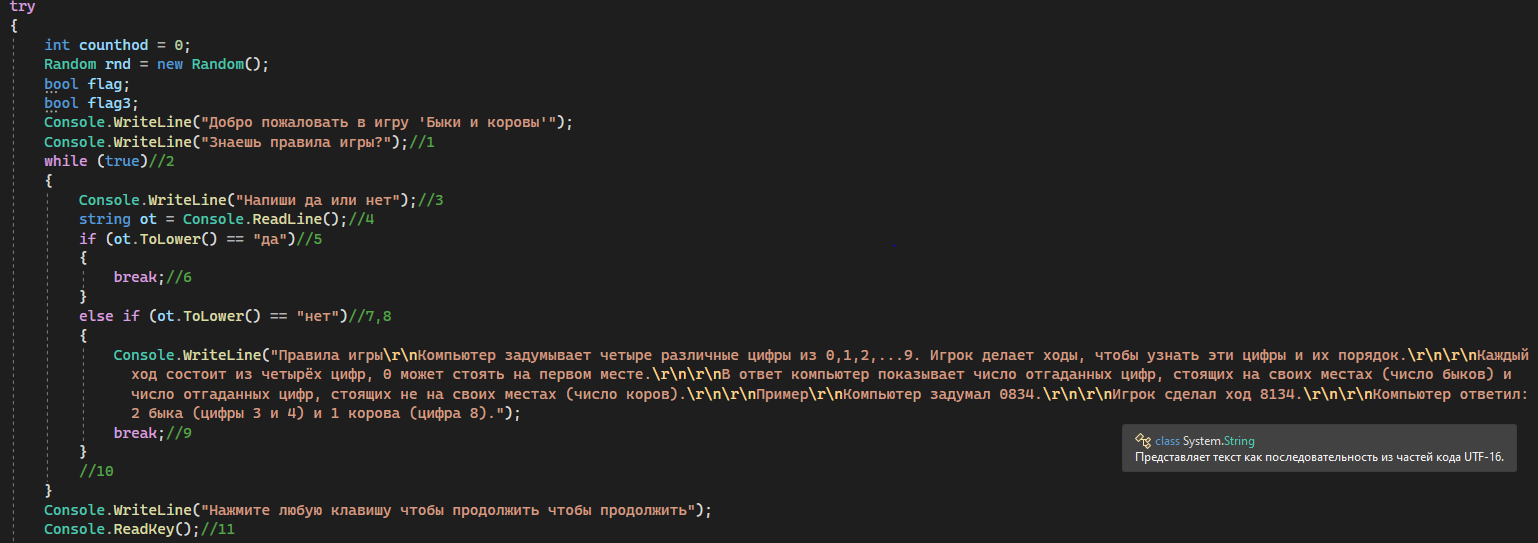
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Входные данные | Условия | Правильные классы | Неправильные классы |
| 1 | х | Х-целое четырехзначное число,  x1!=x2 && x3!=x4 &&  x2!=x3 && x1!=4 &&  x1!=x3 && x2!=x4,    9876 >= x >= 1234,  0<хi<9 | целое четырехзначное число (1),  x1 != x2 && x3 != x4 &&  x2 != x3&&x1 != 4&&  x1 != x3&&x2 != x4 (2),  х>=1023 (3),  х>=9876 (4),  0<х1<9(11)  0<х2<9(12)  0<х3<9(13)  0<х4<9(14)  Число равно заданному числу (5) | Другой тип данных (6),  x>9786 (7),  x<1023(8),  число не четырехзначное (9),  x1=x2 (19) x3=x4(20)  x2=x3(21) x1=4(22)  x1=x3(23) x2=x4 (10)  0>х1>9(15)  0>х2>9(16)  0>х3>9(17)  0>х4>9(18) |

Тесты:

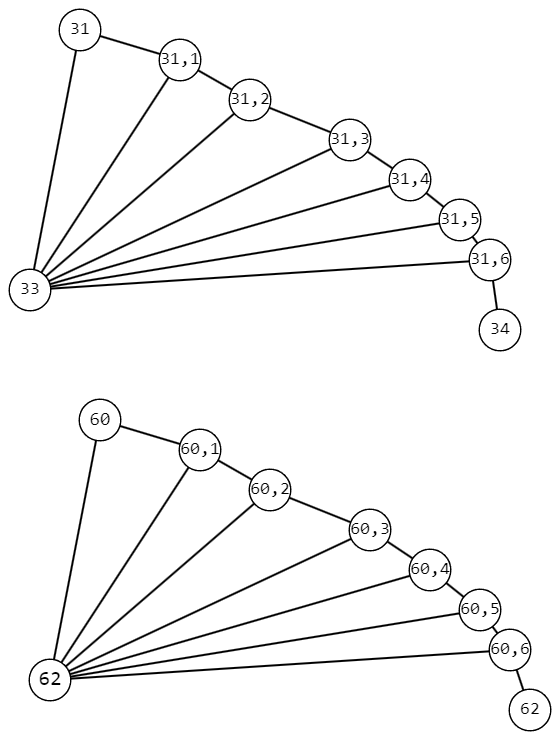
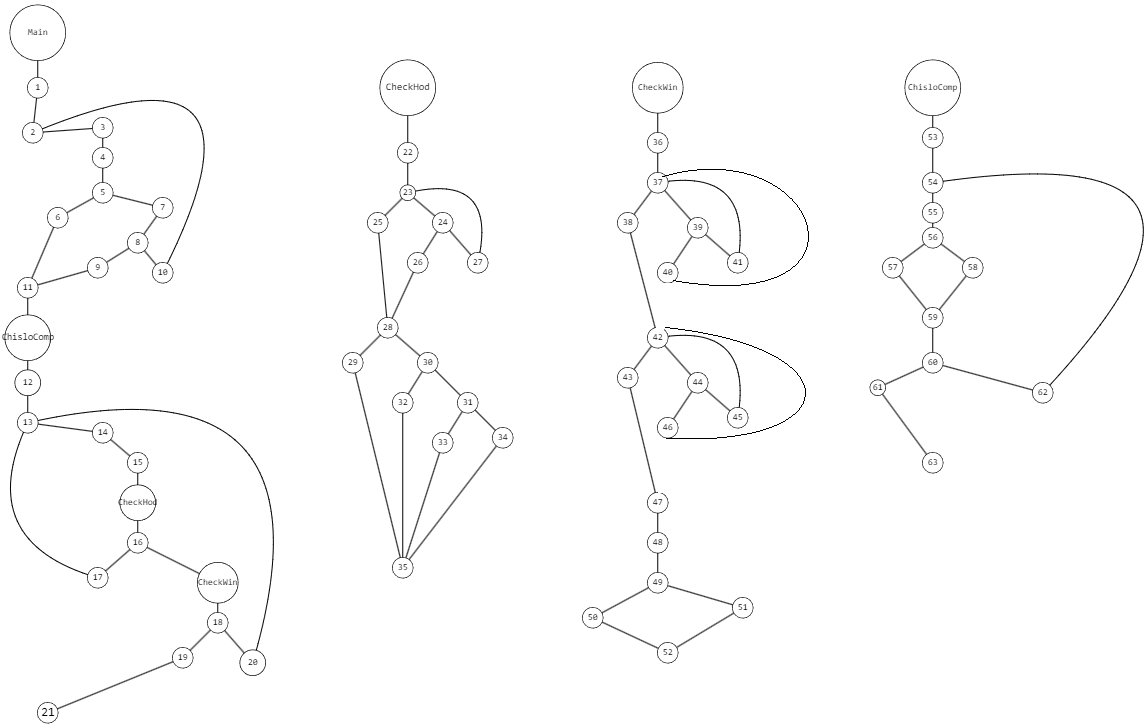
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Значения входных данных | Ожидаемый результат | Правильные классы | Неправильные классы |
| 1 | Х=1023 | Если х равно заданному числу «Вы победили»,  Иначе (Кол-во быков и коров) | 1,2,3, 11,12,13,14 |  |
| 2 | Х=5678 | Если х равно заданному числу «Вы победили»,  Иначе (Кол-во быков и коров) | 1,2, 11,12,13,14 |  |
| 3 | Х=9876 | Если х равно заданному числу «Вы победили»,  Иначе (Кол-во быков и коров) | 1,2,4, 11,12,13,14 |  |
| 4 | Х=3452 | Х равно заданному числу «Вы победили» | 1,2,5,11,12,13,14 |  |
| 5 | Х=-2335 | «Введите положительное число» |  | 8 |
| 6 | Х=1292 | «Введите число с неповторяющимися цифрами» | 11,12,13,14 | 10 |
| 7 | Х=213 | «Введите четырехзначное число» | 11,12,13 | 8,9 |
| 8 | Х=цруг | «Неправильный ввод» |  | 6 |
| 9 | Х=12345 | «Введите четырехзначное число» | 11,12,13,14 | 7,9 |
| 10 | X=1.254 | «Неправильный ввод» |  | 6 |
| 11 | Х=ц123 | «Неправильный ввод» |  | 15 |
| 12 | Х=1р23 | «Неправильный ввод» |  | 16 |
| 13 | Х=12у3 | «Неправильный ввод» |  | 17 |
| 14 | Х=123г | «Неправильный ввод» |  | 18 |
| 15 | Х=1128 | «Введите число с неповторяющимися цифрами» |  | 19 |
| 16 | Х=1218 | «Введите число с неповторяющимися цифрами» |  | 23 |
| 17 | Х=1229 | «Введите число с неповторяющимися цифрами» |  | 21 |
| 18 | Х=1922 | «Введите число с неповторяющимися цифрами» |  | 20 |
| 19 | Х=1291 | «Введите число с неповторяющимися цифрами» |  | 22 |



## Программа



Сложность программы:



Сложность алгоритма по формуле 1(цикломатическая сложность равна

количеству регионов потокового графа) = 28;

Сложность алгоритма по формуле 2(V(G)=E-N+2, где Е — количество дуг =

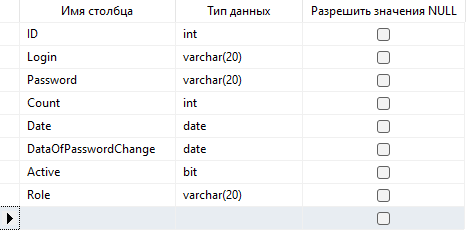
107, N — количество узлов потокового графа = 81) = 28;

Сложность алгоритма по формуле 3(количество предикатных узлов + 1) =33

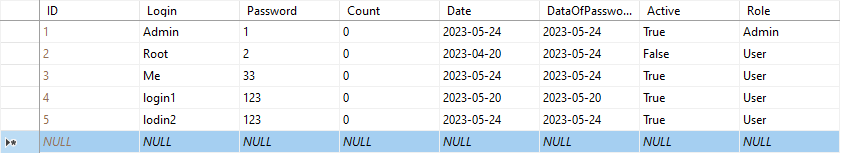
# Тема «Практика 10-11. Работа с базой данных»

## Таблицы в MS SQL Server

Таблица с полями и их типами:



Заполненная таблица:



## Задания и их реализация

Задание 1: сделать вывод всех пользователей в таблицу datagridview, которая добавлена на одну из форм:

public void Reload ()

{

connect.Open( );

adptr = new SqlDataAdapter("select \* from User\_tbl", connect);

table = new DataTable( );

adptr.Fill(table);

dataGridView1.DataSource = table;

connect.Close( );

}

Задание 2: запрограммировать кнопку «регистрация». Проверить на пустоту логина, совпадение пароля и его подтверждения, и только после добавить нового пользователя в БД. Сделать дополнительную проверку: если у нас уже есть пользователь с таким логином, то нужно сообщать об этом.

// регистрация

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

{

connect.Open();

string login, pass, passCheck;

login = loginText.Text;

pass = passText.Text;

passCheck = passCheckText.Text;

DateTime date = DateTime.Now;

if(login=="" || pass == "" || passCheck == "")

{

MessageBox.Show("Не все поля ввода данных заполнены", "Внимание");

connect.Close( );

return;

}

if (pass != passCheck)

{

MessageBox.Show("Пароли не совпадают", "Внимание");

connect.Close( );

return;

}

SqlCommand comm = new SqlCommand($"select count(\*) from dbo.User\_tbl where Login='{login}'", connect);

if ( (int) comm.ExecuteScalar( ) == 0 )

{

string query = $"INSERT INTO dbo.User\_tbl (Login, Password, Count, Date, Active, Role,DateChange) VALUES ('{login}' ,'{pass}' ,0 ,'{date}' ,'True' ,'User','{date}')";

SqlCommand command = new SqlCommand(query, connect);

command.ExecuteNonQuery( );

MessageBox.Show("Новый пользователь добавлен","Информация",MessageBoxButtons.OK, MessageBoxIcon.Information);

loginText.Text = "";

passText.Text = "";

passCheckText.Text = "";

}

else

{

MessageBox.Show("Такой логин уже есть", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close( );

}

Задание 3: авторизация. При аутентификации связка «имя пользователя/пароль» должна совпадать с одной из записей в соответствующей таблице. В случае успешной аутентификации должен быть осуществлен вход в систему. Если аутентификация неуспешна должно появляться сообщение «Вы ввели неверный логин или пароль. Пожалуйста проверьте ещё раз введенные данные». Если в течение трех раз подряд был неверно введен логин/пароль, то система должна заблокировать пользователя. Учетная запись должна блокироваться если пользователь системы не заходил в неё в течении 1 месяца.

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

{

connect.Open( );

string login, pass;

login = SignLogin.Text;

pass = SignPassword.Text;

DateTime date = DateTime.Now;

string datenow = date.ToString("yyyy-MM-dd");

DateTime dateCheck = date.AddDays(-30);

DateTime dateCheck14 = date.AddDays(-14);

DateTime datepass;

DateTime datepass14;

int id;

if ( login == "" || pass == "" )

{

MessageBox.Show("Не все поля ввода данных заполнены", "Внимание", MessageBoxButtons.OK, MessageBoxIcon.Error);

connect.Close( );

return;

}

SqlCommand checkLoginPass = new SqlCommand($"select \* from dbo.User\_tbl where Password='{pass}' and Login = '{login}'", connect);

SqlCommand CheckLogin = new SqlCommand($"select \* from dbo.User\_tbl where Login = '{login}'", connect);

SqlCommand checkRole = new SqlCommand($"select Role from dbo.User\_tbl where Password='{pass}' and Login = '{login}'", connect);

SqlCommand checkCount = new SqlCommand($"select Count from dbo.User\_tbl where Login = '{login}'", connect);

SqlCommand countPlus = new SqlCommand($"Update User\_tbl set Count = Count + 1 where Login = '{login}'", connect);

SqlCommand ActiveBlock = new SqlCommand($"Update User\_tbl set Active = 0 where Login = '{login}'", connect);

SqlCommand CheckActive = new SqlCommand($"select Active from dbo.User\_tbl where Login = '{login}'", connect);

SqlCommand CountNull = new SqlCommand($"Update User\_tbl set Count = 0 where Login = '{login}'", connect);

SqlCommand CheckDays = new SqlCommand($"select Date from dbo.User\_tbl where Login = '{login}'", connect);

SqlCommand DaysNow = new SqlCommand($"Update User\_tbl set Date = '{datenow}' where Login = '{login}'", connect);

SqlCommand CheckDays14 = new SqlCommand($"select DateChange from dbo.User\_tbl where Login = '{login}'", connect);

SqlCommand IdSave = new SqlCommand($"select ID from dbo.User\_tbl where Login = '{login}'", connect);

datepass = (DateTime)CheckDays.ExecuteScalar( );

datepass14 = (DateTime) CheckDays14.ExecuteScalar( );

if ( datepass < dateCheck )

{

ActiveBlock.ExecuteNonQuery( );

MessageBox.Show("Вы заблокированы. Так как вы давно не заходили (30 дней) ", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

connect.Close( );

return;

}

if ( CheckLogin.ExecuteScalar( ) != null )

{

if ( (bool) CheckActive.ExecuteScalar( ) == true )

{

if ( checkLoginPass.ExecuteScalar( ) != null )

{

if ( checkRole.ExecuteScalar( ).ToString( ) == "User" )

{

if ( datepass14 < dateCheck14 )

{

MessageBox.Show("Вы не меняли пароль в течение 14 дней\nМы переносим вас в форму для изменения пароля", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

CountNull.ExecuteNonQuery( );

id = (int)IdSave.ExecuteScalar( );

passForm = new UpdatePass(id);

passForm.ShowDialog( );

}

else

{

id = (int) IdSave.ExecuteScalar( );

userForm = new UserApp(id);

CountNull.ExecuteNonQuery( );

DaysNow.ExecuteNonQuery( );

userForm.ShowDialog( );

}

}

else

{

id = (int) IdSave.ExecuteScalar( );

adminForm = new AdminApp(id);

DaysNow.ExecuteNonQuery( );

CountNull.ExecuteNonQuery( );

adminForm.ShowDialog( );

}

}

else

{

if ( (int) checkCount.ExecuteScalar( ) < 3 )

{

MessageBox.Show("Неверный пароль", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

countPlus.ExecuteNonQuery( );

}

else

{

MessageBox.Show("Вы заблокированы, Превышен лимит допустимых ошибок", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

ActiveBlock.ExecuteNonQuery( );

}

}

}

else

{

MessageBox.Show("Вы заблокированы. Обратитесь к администратору", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show("Такой логин не существует", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close( );

}

Задание 4: после успешной авторизации, в случае если пользователь не менял пароль в течение 14 дней должна выводиться форма для смены пароля. Если в форме смены пароля - текущий пароль введен неверно или новый пароль не совпадает с подтверждением должно выводиться соответствующее сообщение.

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

{

connect.Open( );

string oldpass, pass, passCheck;

oldpass = oldPassText.Text;

pass = newpassText.Text;

passCheck = newpassCheckText.Text;

DateTime date = DateTime.Now;

SqlCommand CheckOldPass = new SqlCommand($"select Password from dbo.User\_tbl where ID = '{ID}'", connect);

string checkold = CheckOldPass.ExecuteScalar( ).ToString();

if ( oldpass == "" || pass == "" || passCheck == "" )

{

MessageBox.Show("Не все поля ввода данных заполнены", "Внимание");

connect.Close( );

return;

}

if ( pass != passCheck )

{

MessageBox.Show("Пароли не совпадают", "Внимание");

connect.Close( );

return;

}

if ( oldpass != checkold )

{

MessageBox.Show("Cтарый пароль не совпадает", "Внимание");

connect.Close( );

return;

}

SqlCommand UpdatePass = new SqlCommand($"Update User\_tbl set Password = '{passCheck}' where ID = '{ID}'", connect);

SqlCommand UpdateDateChange = new SqlCommand($"Update User\_tbl set DateChange = '{date}' where ID = '{ID}'", connect);

UpdatePass.ExecuteNonQuery( );

UpdateDateChange.ExecuteNonQuery( );

MessageBox.Show("Пароль изменен", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

connect.Close( );

this.Close( );

}

Задание 5: при добавлении нового пользователя в форме администратора, система должна поверять его на наличие и если пользователь в системе уже существует должно выводится соответствующее сообщение.

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

{

connect.Open();

string login, pass, passCheck;

login = loginText.Text;

pass = passText.Text;

passCheck = passCheckText.Text;

DateTime date = DateTime.Now;

if(login=="" || pass == "" || passCheck == "")

{

MessageBox.Show("Не все поля ввода данных заполнены", "Внимание");

connect.Close( );

return;

}

if (pass != passCheck)

{

MessageBox.Show("Пароли не совпадают", "Внимание");

connect.Close( );

return;

}

SqlCommand comm = new SqlCommand($"select count(\*) from dbo.User\_tbl where Login='{login}'", connect);

if ( (int) comm.ExecuteScalar( ) == 0 )

{

string query = $"INSERT INTO dbo.User\_tbl (Login, Password, Count, Date, Active, Role,DateChange) VALUES ('{login}' ,'{pass}' ,0 ,'{date}' ,'True' ,'User','{date}')";

SqlCommand command = new SqlCommand(query, connect);

command.ExecuteNonQuery( );

MessageBox.Show("Новый пользователь добавлен","Информация",MessageBoxButtons.OK, MessageBoxIcon.Information);

loginText.Text = "";

passText.Text = "";

passCheckText.Text = "";

}

else

{

MessageBox.Show("Такой логин уже есть", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close( );

}

Задание 6: предусмотреть возможность изменения текущих данных пользователей, в том числе снятие блокировки.

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

{

DateTime datenow = DateTime.Now;

string login=loginText.Text;

connect.Open( );

if ( login == "" )

{

MessageBox.Show("Поле ввода пустое", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

connect.Close( );

return;

}

SqlCommand comm = new SqlCommand($"select count(\*) from dbo.User\_tbl where Login='{login}'", connect);

if ( (int) comm.ExecuteScalar( ) != 0 )

{

SqlCommand CountNull = new SqlCommand($"Update User\_tbl set Count = 0 where Login = '{login}'", connect);

SqlCommand DaysNow = new SqlCommand($"Update User\_tbl set Date = '{datenow}' where Login = '{login}'", connect);

SqlCommand Unlock = new SqlCommand($"Update User\_tbl set Active = 'true' where Login = '{login}'", connect);

Unlock.ExecuteNonQuery( );

DaysNow.ExecuteNonQuery( );

CountNull.ExecuteNonQuery( );

MessageBox.Show("Пользователь разблокирован", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

loginText.Text = "";

}

else

{

MessageBox.Show("Такой логин не существует", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close( );

}

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

{

string login = loginText.Text;

connect.Open( );

if ( login == "" )

{

MessageBox.Show("Поле ввода пустое", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

connect.Close( );

return;

}

SqlCommand comm = new SqlCommand($"select count(\*) from dbo.User\_tbl where Login='{login}'", connect);

if ( (int) comm.ExecuteScalar( ) != 0 )

{

SqlCommand Lock = new SqlCommand($"Update User\_tbl set Active = 'false' where Login = '{login}'", connect);

Lock.ExecuteNonQuery( );

MessageBox.Show("Пользователь заблокирован", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

loginText.Text = "";

}

else

{

MessageBox.Show("Такой логин не существует", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close( );

# Тема «Практика 12. Работа с базой данных»

## Таблицы в MS SQL Server в режиме проекта

Таблица Группы

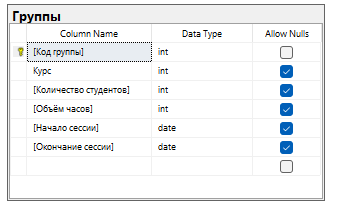


Таблица Дисциплины

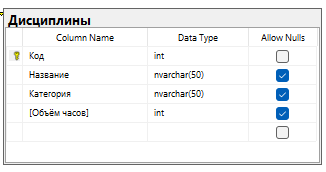


Таблица Преподаватель

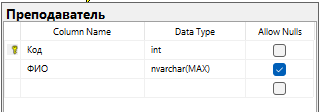
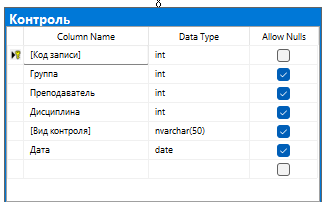
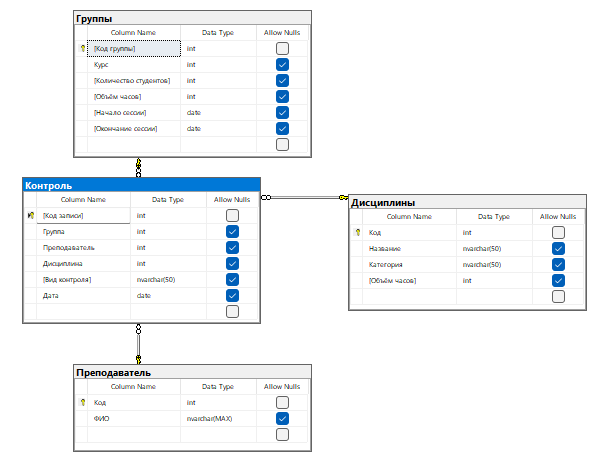


Таблица Контроль



## Связи между таблицами в MS SQL Server



## Таблицы в режиме просмотра данных в MS SQL Server

Таблица Группы.



Таблица Дисциплины

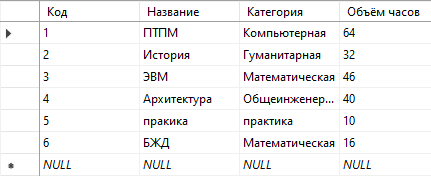


Таблица Преподаватель

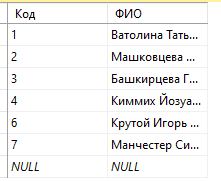
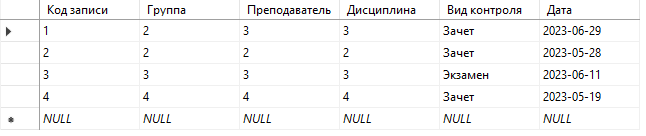
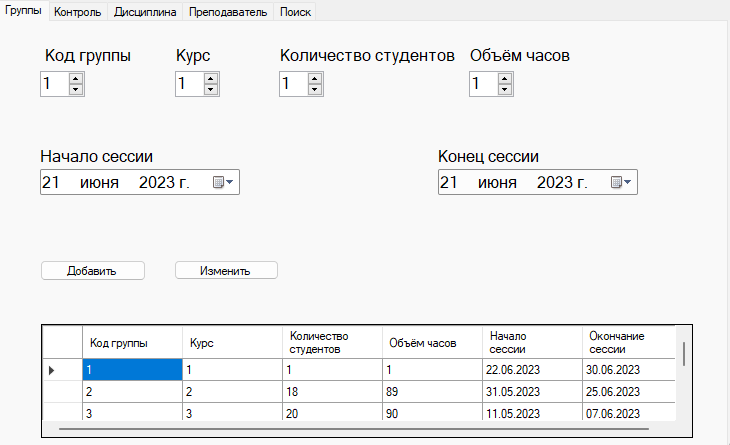
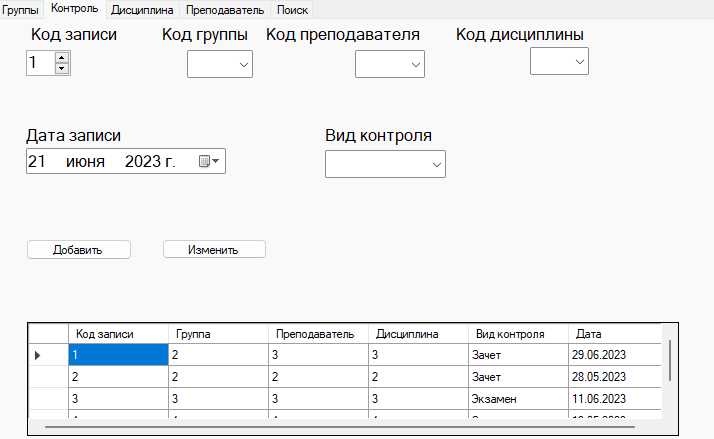
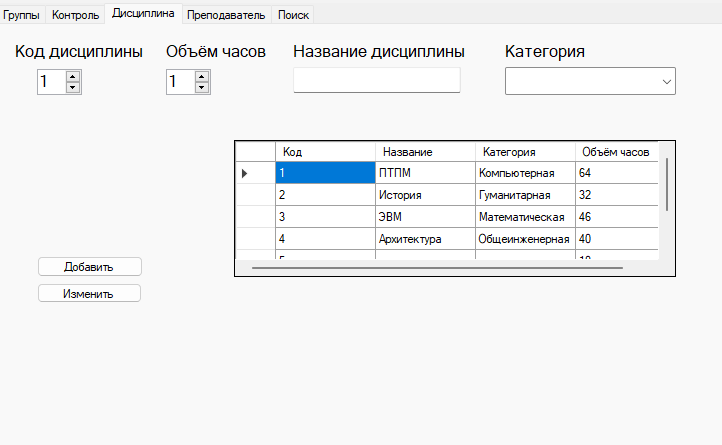
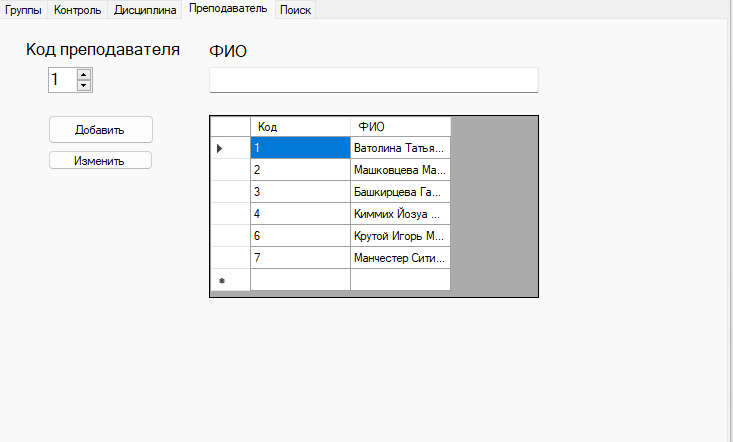
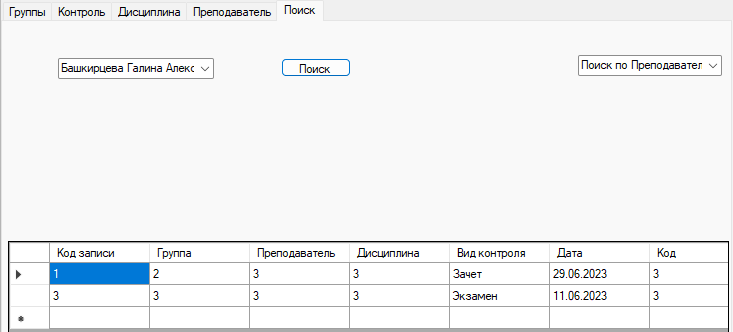


Таблица Контроль



## Внешний вид формы



## Методы добавления данных в таблицу БД

Методы добавления:

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

{

int kod, kurs, countst, hours;

DateTime startses, endses;

DateTime datenow = DateTime.Now;

kod = (int)numericUpDown1.Value;

kurs = (int) numericUpDown2.Value;

countst = (int) numericUpDown3.Value;

hours = (int) numericUpDown4.Value;

startses = StartSession.Value;

endses = EndSession.Value;

connect.Open( );

SqlCommand comm = new SqlCommand($"select count(\*) from dbo.Группы where [Код группы] = {kod}", connect);

if ( (int) comm.ExecuteScalar( ) == 0 )

{

if(startses<datenow && endses < startses )

{

MessageBox.Show("Значение дат неправильно.\nИсправьте пожалуйста", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

connect.Close( );

return;

}

string query = $"INSERT INTO dbo.Группы ([Код группы], [Курс], [Количество студентов], [Объём часов], [Начало сессии], [Окончание сессии]) VALUES ({kod} ,{kurs} ,{countst},{hours},'{startses}' ,'{endses}')";

SqlCommand command = new SqlCommand(query, connect);

command.ExecuteNonQuery( );

MessageBox.Show("Группа добавлена", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Такая группа уже существует", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close( );

ReloadGroup();

ComboboxLoad();

}

private void button4\_Click(object sender, EventArgs e)

{

int kod, group, discip, prepod;

string vid;

DateTime date;

DateTime datenow = DateTime.Now;

if(comboBox2.Text=="" || comboBox3.Text==""|| comboBox4.Text=="")

{

MessageBox.Show("Выберите данные", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

kod = (int)numericUpDown11.Value;

group = int.Parse(comboBox2.Text);

discip = int.Parse(comboBox3.Text);

prepod = int.Parse(comboBox4.Text);

date = dateZapis.Value;

vid = comboBox5.Text;

connect.Open();

SqlCommand comm = new SqlCommand($"select count(\*) from dbo.Контроль where [Код записи] = {kod}", connect);

if ((int)comm.ExecuteScalar() == 0)

{

if (date < datenow)

{

MessageBox.Show("Значение дат неправильно.\nИсправьте пожалуйста", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

connect.Close();

return;

}

string query = $"INSERT INTO dbo.Контроль ([Код записи], [Группа], [Преподаватель], [Дисциплина], [Вид контроля], [Дата]) VALUES ({kod} ,{group} ,{discip},{prepod},'{date}' ,'{vid}')";

SqlCommand command = new SqlCommand(query, connect);

command.ExecuteNonQuery();

MessageBox.Show("Запись добавлена", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Такая запись уже существует\nМожет вы хотели отредактировать", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close();

ReloadControl();

ComboboxLoad();

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

int kod, hours;

string kategoria, name;

kod = (int) numericUpDown8.Value;

kategoria = comboBox1.Text;

name = textBox1.Text;

hours = (int) numericUpDown5.Value;

if(kategoria == "" || name == "") {

MessageBox.Show("Не все поля ввода данных заполнены", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

connect.Open();

SqlCommand comm = new SqlCommand($"select count(\*) from dbo.Дисциплины where [Код] = {kod}", connect);

if ((int) comm.ExecuteScalar() == 0) {

string query = $"INSERT INTO dbo.Дисциплины (Код, Название, Категория, [Объём часов]) VALUES ({kod} ,'{name}' ,'{kategoria}',{hours})";

SqlCommand command = new SqlCommand(query, connect);

command.ExecuteNonQuery();

MessageBox.Show("Дисциплина добавлена", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

} else {

MessageBox.Show("Такой код дисциплины уже существует", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close();

ReloadDiscip();

ComboboxLoad();

}

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

int kod;

string name;

kod = (int) numericUpDown6.Value;

name = textBox2.Text;

if (name == "") {

MessageBox.Show("Не все поля ввода данных заполнены", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string[] checkname = name.Split(new char[] { ' ' },StringSplitOptions.RemoveEmptyEntries);

if(checkname.Length != 3) {

MessageBox.Show("Пожалуйста введите ФИО правильно", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

connect.Open();

SqlCommand comm = new SqlCommand($"select count(\*) from Преподаватель where [Код] = {kod}", connect);

if ((int) comm.ExecuteScalar() == 0) {

string query = $"INSERT INTO Преподаватель (Код, ФИО) VALUES ({kod} ,'{name}')";

SqlCommand command = new SqlCommand(query, connect);

command.ExecuteNonQuery();

MessageBox.Show("Преподаватель добавлен", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

} else {

MessageBox.Show("Такой код преподавателя уже существует", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close();

ReloadPrepod();

ComboboxLoad();

}

Методы редактирования:

private void button5\_Click(object sender, EventArgs e)

{

int kod, kurs, countst, hours;

DateTime startses, endses;

DateTime datenow = DateTime.Now;

kod = (int)numericUpDown1.Value;

kurs = (int)numericUpDown2.Value;

countst = (int)numericUpDown3.Value;

hours = (int)numericUpDown4.Value;

startses = StartSession.Value;

endses = EndSession.Value;

connect.Open();

SqlCommand comm = new SqlCommand($"select count(\*) from dbo.Группы where [Код группы] = {kod}", connect);

if ((int)comm.ExecuteScalar() == 0)

{

MessageBox.Show("Может вы хотели Добавить такую группу", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

if (startses < datenow && endses < startses)

{

MessageBox.Show("Значение дат неправильно.\nИсправьте пожалуйста", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

connect.Close();

return;

}

string query = $"Update Группы set [Количество студентов]= {countst} where [Код группы]= {kod}";

string query1 = $"Update Группы set [Объём часов]= {hours} where [Код группы]={kod}";

string query2 = $"Update Группы set Курс= {kurs} where [Код группы]= {kod}";

string query3 = $"Update Группы set [Начало сессии]= '{startses}' where [Код группы]= {kod}";

string query4 = $"Update Группы set [Окончание сессии]= '{endses}' where [Код группы]= {kod}";

SqlCommand command = new SqlCommand(query, connect);

SqlCommand command1 = new SqlCommand(query1, connect);

SqlCommand command2 = new SqlCommand(query2, connect);

SqlCommand command3 = new SqlCommand(query3, connect);

SqlCommand command4 = new SqlCommand(query4, connect);

command.ExecuteNonQuery();

command1.ExecuteNonQuery();

command2.ExecuteNonQuery();

command3.ExecuteNonQuery();

command4.ExecuteNonQuery();

MessageBox.Show("Группа изменена", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

connect.Close();

ReloadGroup();

ComboboxLoad();

}

private void button6\_Click(object sender, EventArgs e)

{

int kod, group, discip, prepod;

string vid;

DateTime date;

DateTime datenow = DateTime.Now;

if (comboBox2.Text == "" || comboBox3.Text == "" || comboBox4.Text == "")

{

MessageBox.Show("Выберите данные", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

kod = (int)numericUpDown11.Value;

group = int.Parse(comboBox2.Text);

discip = int.Parse(comboBox3.Text);

prepod = int.Parse(comboBox4.Text);

date = dateZapis.Value;

vid = comboBox5.Text;

connect.Open();

SqlCommand comm = new SqlCommand($"select count(\*) from dbo.Контроль where [Код записи] = {kod}", connect);

if ((int)comm.ExecuteScalar() != 0)

{

if (date < datenow)

{

MessageBox.Show("Значение дат неправильно.\nИсправьте пожалуйста", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

connect.Close();

return;

}

string query = $"update Контроль set Группа = {group} where [Код записи]= {kod}";

string query1 = $"update Контроль set Преподаватель = '{prepod}' where [Код записи]= {kod}";

string query2 = $"update Контроль set Дисциплина = {discip} where [Код записи]= {kod}";

string query3 = $"update Контроль set [Вид контроля] = '{vid}' where [Код записи]= {kod}";

string query4 = $"update Контроль set Дата = '{date}' where [Код записи]= {kod}";

SqlCommand command = new SqlCommand(query, connect);

SqlCommand command1= new SqlCommand(query1, connect);

SqlCommand command2= new SqlCommand(query2, connect);

SqlCommand command3 = new SqlCommand(query3, connect);

SqlCommand command4 = new SqlCommand(query4, connect);

command.ExecuteNonQuery();

command1.ExecuteNonQuery();

command2.ExecuteNonQuery();

command3.ExecuteNonQuery();

command4.ExecuteNonQuery();

MessageBox.Show("Запись изменена", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Такая не существует\nМожет вы хотели отредактировать", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close();

ReloadControl();

ComboboxLoad();

}

private void button7\_Click(object sender, EventArgs e)

{

int kod, hours;

string kategoria, name;

kod = (int)numericUpDown8.Value;

kategoria = comboBox1.Text;

name = textBox1.Text;

hours = (int)numericUpDown5.Value;

if (kategoria == "" || name == "")

{

MessageBox.Show("Не все поля ввода данных заполнены", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

connect.Open();

SqlCommand comm = new SqlCommand($"select count(\*) from dbo.Дисциплины where [Код] = {kod}", connect);

if ((int)comm.ExecuteScalar() != 0)

{

string query = $"update Дисциплины set Название= '{name}' where Код = {kod}";

string query1 = $"update Дисциплины set Категория= '{kategoria}' where Код = {kod}";

string query2 = $"update Дисциплины set [Объём часов]= {hours} where Код = {kod}";

SqlCommand command = new SqlCommand(query, connect);

SqlCommand command1 = new SqlCommand(query1, connect);

SqlCommand command2 = new SqlCommand(query2, connect);

command.ExecuteNonQuery();

command1.ExecuteNonQuery();

command2.ExecuteNonQuery();

MessageBox.Show("Дисциплина изменена", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Такой код дисциплины не существует", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close();

ReloadDiscip();

ComboboxLoad();

}

private void button8\_Click(object sender, EventArgs e)

{

int kod;

string name;

kod = (int)numericUpDown6.Value;

name = textBox2.Text;

if (name == "")

{

MessageBox.Show("Не все поля ввода данных заполнены", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

string[] checkname = name.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries);

if (checkname.Length != 3)

{

MessageBox.Show("Пожалуйста введите ФИО правильно", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

connect.Open();

SqlCommand comm = new SqlCommand($"select count(\*) from Преподаватель where [Код] = {kod}", connect);

if ((int)comm.ExecuteScalar() != 0)

{

string query = $"update Преподаватель set ФИО = '{name}' where Код = {kod}";

SqlCommand command = new SqlCommand(query, connect);

command.ExecuteNonQuery();

MessageBox.Show("Преподаватель изменен", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("Такой код преподавателя не существует", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

connect.Close();

ReloadPrepod();

ComboboxLoad();

## 

## Методы, реализующие поиск по БД

private void button9\_Click(object sender, EventArgs e)

{

if (comboBox6.Text == "")

{

MessageBox.Show("Выберите тип поиска", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if(comboBox6.Text == "Поиск по Преподавателю")

{

string query = $"select \* from Контроль \r\ninner join Преподаватель on Преподаватель.Код = Контроль.Преподаватель\r\nwhere Преподаватель.ФИО = '{comboBox7.Text}'";

ReloadSearch(query);

}

else if(comboBox6.Text == "Поиск по Дисциплине")

{

string query = $"select \* from Контроль inner join Дисциплины on Дисциплины.Код = Контроль.Дисциплина where Дисциплины.Название = '{comboBox7.Text}'";

ReloadSearch(query);

}

}

private void comboBox7\_KeyUp(object sender, KeyEventArgs e)

{

comboBox6.Text = "";

comboBox7.Text = "";

comboBox7.Items.Clear();

}

private void ComboBoxPrepodLoad()

{

comboBox7.Items.Clear();

connect.Open();

adptr = new SqlDataAdapter("select distinct Преподаватель.ФИО from Преподаватель", connect);

table = new DataTable();

adptr.Fill(table);

foreach (DataRow r in table.Rows)

{

foreach (var cell in r.ItemArray)

comboBox7.Items.Add(cell);

}

connect.Close();

}

private void ComboBoxDiscipLoad()

{

comboBox7.Items.Clear();

connect.Open();

adptr = new SqlDataAdapter("select distinct Дисциплины.Название from Дисциплины", connect);

table = new DataTable();

adptr.Fill(table);

foreach (DataRow r in table.Rows)

{

foreach (var cell in r.ItemArray)

comboBox7.Items.Add(cell);

}

connect.Close();

}

private void comboBox6\_SelectedIndexChanged(object sender, EventArgs e)

{

comboBox7.Text = "";

if (comboBox6.Text == "Поиск по Преподавателю")

{

ComboBoxPrepodLoad();

}

else if (comboBox6.Text == "Поиск по Дисциплине")

{

ComboBoxDiscipLoad();

}

}

public void ReloadSearch( string query)

{

connect.Open();

adptr = new SqlDataAdapter(query, connect);

table = new DataTable();

adptr.Fill(table);

dataGridView5.DataSource = table;

connect.Close();

## Методы обновления всех DataGrid

Метод для удаления теста по его TestID:

public void ReloadGroup ()

{

connect.Open( );

adptr = new SqlDataAdapter("select \* from Группы", connect);

table = new DataTable( );

adptr.Fill(table);

dataGridView1.DataSource = table;

connect.Close( );

}

public void ReloadDiscip () {

connect.Open();

adptr = new SqlDataAdapter("select \* from Дисциплины", connect);

table = new DataTable();

adptr.Fill(table);

dataGridView2.DataSource = table;

connect.Close();

}

public void ReloadPrepod () {

connect.Open();

adptr = new SqlDataAdapter("select \* from Преподаватель", connect);

table = new DataTable();

adptr.Fill(table);

dataGridView3.DataSource = table;

connect.Close();

}

public void ReloadControl () {

connect.Open();

adptr = new SqlDataAdapter("select \* from Контроль", connect);

table = new DataTable();

adptr.Fill(table);

dataGridView4.DataSource = table;

connect.Close();

}

public void ReloadSearch( string query)

{

connect.Open();

adptr = new SqlDataAdapter(query, connect);

table = new DataTable();

adptr.Fill(table);

dataGridView5.DataSource = table;

connect.Close();

}

Метод обновления всех ComboBox:

public void ComboboxLoad () {

comboBox1.Items.Clear();

comboBox2.Items.Clear();

comboBox3.Items.Clear();

comboBox4.Items.Clear();

connect.Open();

adptr = new SqlDataAdapter("select distinct Дисциплины.Категория from Дисциплины", connect);

table = new DataTable();

adptr.Fill(table);

foreach (DataRow r in table.Rows) {

foreach (var cell in r.ItemArray)

comboBox1.Items.Add(cell);

}

adptr = new SqlDataAdapter("select distinct Дисциплины.Код from Дисциплины", connect);

table = new DataTable();

adptr.Fill(table);

foreach (DataRow r in table.Rows)

{

foreach (var cell in r.ItemArray)

comboBox4.Items.Add(cell);

}

adptr = new SqlDataAdapter("select distinct Группы.[Код группы] from Группы", connect);

table = new DataTable();

adptr.Fill(table);

foreach (DataRow r in table.Rows)

{

foreach (var cell in r.ItemArray)

comboBox2.Items.Add(cell);

}

adptr = new SqlDataAdapter("select distinct Преподаватель.Код from Преподаватель", connect);

table = new DataTable();

adptr.Fill(table);

foreach (DataRow r in table.Rows)

{

foreach (var cell in r.ItemArray)

comboBox3.Items.Add(cell);

}

connect.Close();

}