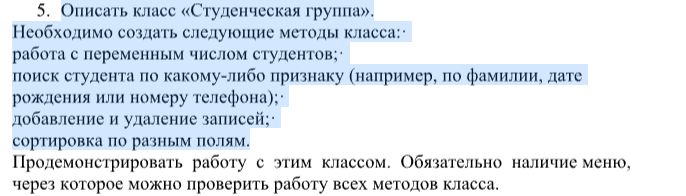
Задание:



Входные и выходные данные:

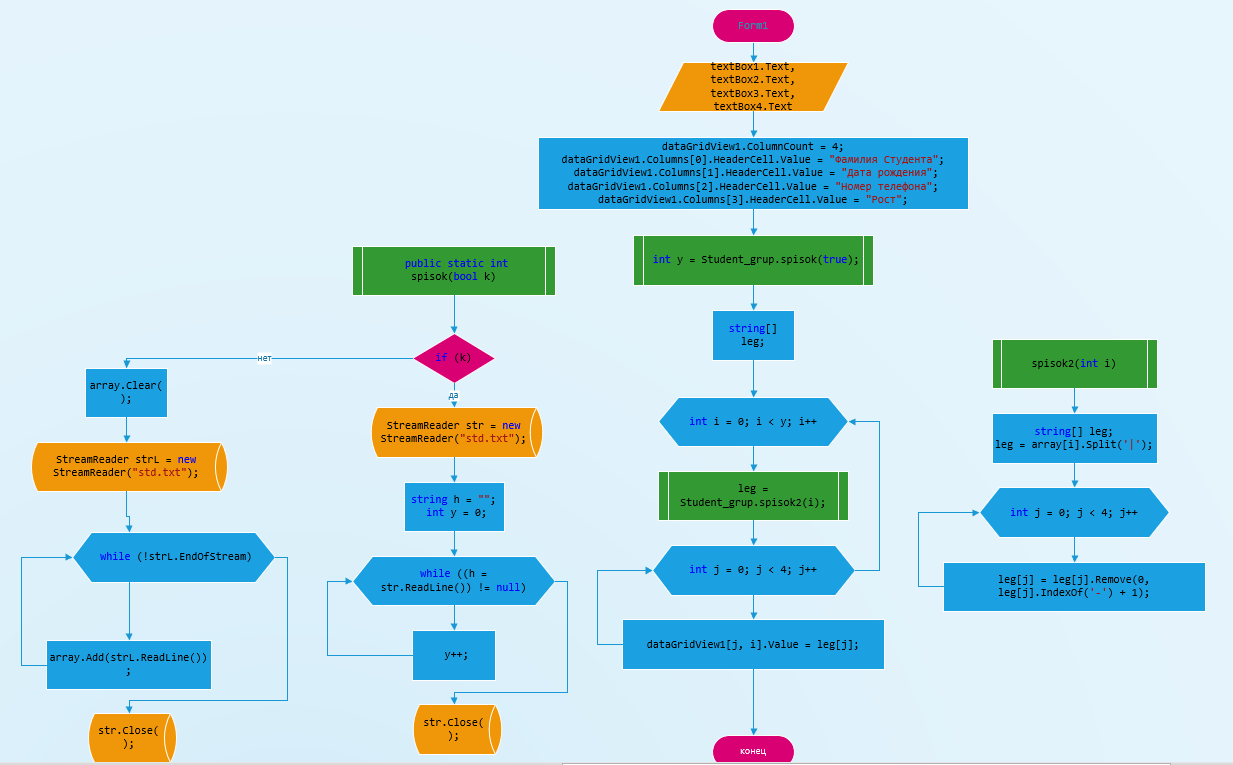
Входные данные:

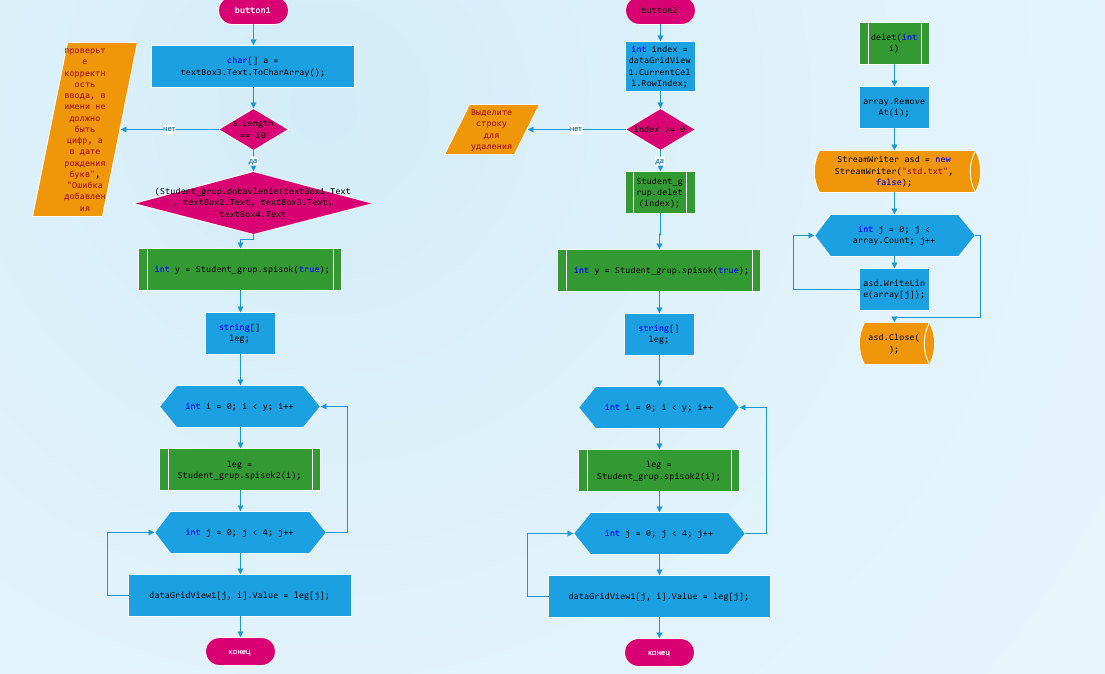


Выходные данные:



Блок-Схема:





Листинг:

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

dataGridView1.ColumnCount = 4;

dataGridView1.Columns[0].HeaderCell.Value = "Фамилия Студента";

dataGridView1.Columns[1].HeaderCell.Value = "Дата рождения";

dataGridView1.Columns[2].HeaderCell.Value = "Номер телефона";

dataGridView1.Columns[3].HeaderCell.Value = "Рост";

int y = Student\_grup.spisok(true);

dataGridView1.RowCount = y;

Student\_grup.spisok(false);

string[] leg;

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

{

leg = Student\_grup.spisok2(i);

for (int j = 0; j < 4; j++)

{

dataGridView1[j, i].Value = leg[j];

}

}

}

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

{

char[] a = textBox3.Text.ToCharArray();

if (a.Length == 10)

{

if (Student\_grup.dobavlenie(textBox1.Text, textBox2.Text, textBox3.Text, textBox4.Text))

{

MessageBox.Show("Студент добавлен в список");

int y = Student\_grup.spisok(true);

dataGridView1.RowCount = y;

Student\_grup.spisok(false);

string[] leg;

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

{

leg = Student\_grup.spisok2(i);

for (int j = 0; j < 4; j++)

{

dataGridView1[j, i].Value = leg[j];

}

}

}

else

{

MessageBox.Show("проверьте корректность ввода, в имени не должно быть цифр, а в дате рождения букв", "Ошибка добавления");

}

}

else

{

MessageBox.Show("Неверный номер телефона");

}

}

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

{

int index = dataGridView1.CurrentCell.RowIndex;

if(index >= 0)

{

Student\_grup.delet(index);

int y = Student\_grup.spisok(true);

dataGridView1.RowCount = y;

Student\_grup.spisok(false);

string[] leg;

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

{

leg = Student\_grup.spisok2(i);

for (int j = 0; j < 4; j++)

{

dataGridView1[j, i].Value = leg[j];

}

}

}

else

{

MessageBox.Show("Выделите строку для удаления");

}

}

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

{

}

}

class Student\_grup

{

public static bool dobavlenie(string fam, string data\_rozd, string nomer\_tel, string rost)

{

try

{

if (proverk(fam) && proverk(data\_rozd) && !proverk(nomer\_tel) && !proverk(rost))

{

return false;

}

else

{

if (Convert.ToInt16(rost) < 140 || Convert.ToInt16(rost) > 220)

{

return false;

}

else

{

StreamWriter asd = new StreamWriter("std.txt", true);

string a = $"Фамилия-{fam}|Дата рождения-{data\_rozd}|Номер телефона-{nomer\_tel}|Рост-{rost}";

asd.WriteLine(a);

asd.Close();

return true;

}

}

}

catch

{

return false;

}

}

static bool proverk(string a)

{

bool proverka\_cifr = (a.Contains('0') || a.Contains('1') || a.Contains('2') || a.Contains('3') || a.Contains('4') || a.Contains('5') || a.Contains('6') || a.Contains('7') || a.Contains('8') || a.Contains('9'));

return proverka\_cifr;

}

static List<String> array = new List<string>();

public static int spisok(bool k)

{

if (k)

{

StreamReader str = new StreamReader("std.txt");

string h = "";

int y = 0;

while ((h = str.ReadLine()) != null)

{

y++;

}

str.Close();

return y;

}

else

{

array.Clear();

StreamReader strL = new StreamReader("std.txt");

while (!strL.EndOfStream)

{

array.Add(strL.ReadLine());

}

strL.Close();

return 0;

}

}

public static string[] spisok2(int i)

{

string[] leg;

leg = array[i].Split('|');

for (int j = 0; j < 4; j++)

{

leg[j] = leg[j].Remove(0, leg[j].IndexOf('-') + 1);

}

return leg;

}

public static void delet(int i)

{

array.RemoveAt(i);

StreamWriter asd = new StreamWriter("std.txt", false);

for (int j = 0; j < array.Count; j++)

{

asd.WriteLine(array[j]);

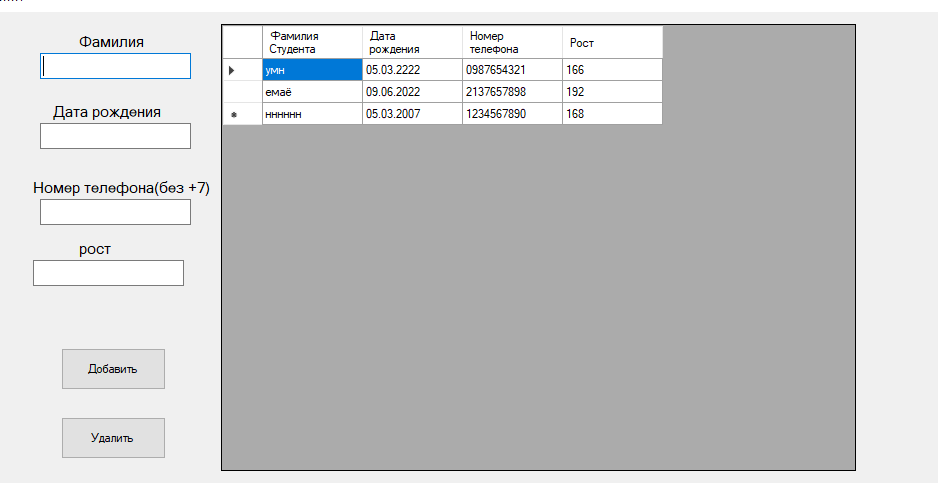
}

asd.Close();

}

}

Тестовые ситуации:



Вывод:

Программа работает корректно