Федеральное агентство по образованию

Государственное Образовательное Учреждение Высшего

Профессионального Образования

Уфимский Государственный Авиационный Технический Университет

Кафедра ВМиК

Лабораторная работа №4.2

По дисциплине “Объектно-ориентированное программирование”

Тема: MVC

Выполнил:

Студент группы ПРО-230

Аверин Р.В.

Проверил:

доцент кафедры ВМиК

Котельников В.А.

Уфа-2022

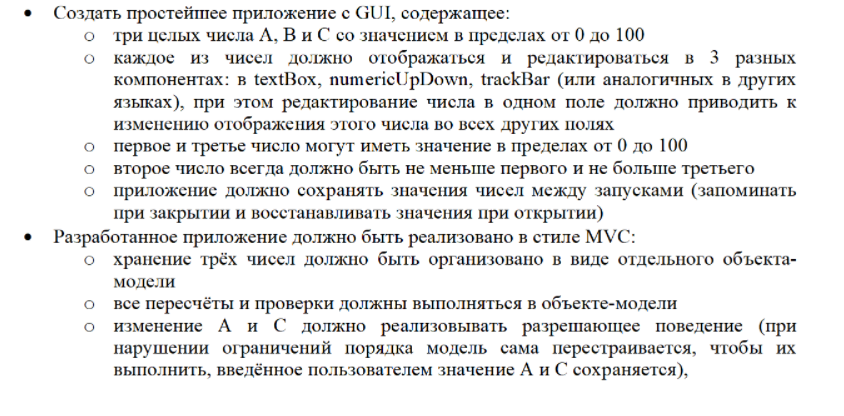
**Цель**:

Изучить схему разделения данных приложения MVC.

**Задачи:**

Написать программу, иллюстрирующую работу с MVC.

**Задание:**





**Ход работы:**

Программа представляет собой форму с девятью объектами по три типа: NumericUpDown, TextBox и TrackBar. Три объекта разных типов соответствуют одному из чисел например: A, B, C. A<=B<=C. Когда вводишь в один из объектов число, вся модель (9 объектов) изменяется в соответствии с требованиями.

1. **Функция ввода числа A.** Ввод числа А при нарушении чисел В и С меняется.Показан на рисунке 1, 2, 3.

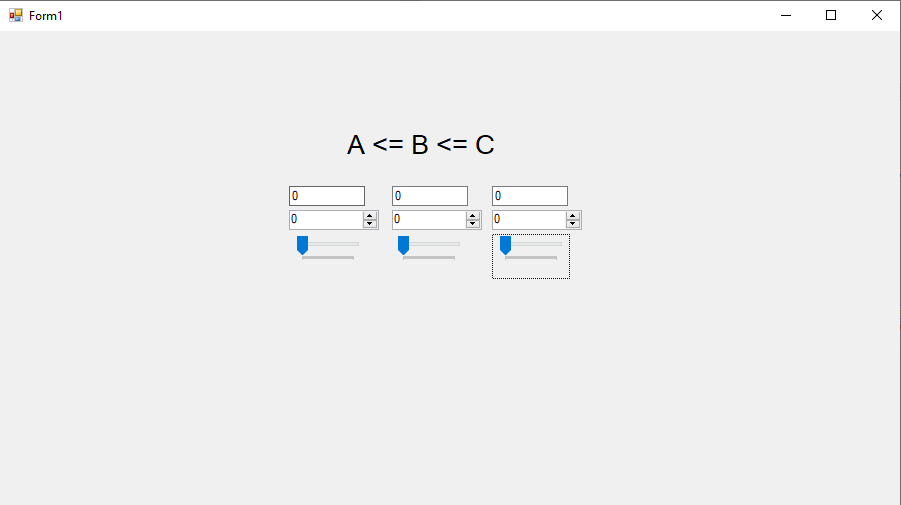


Рисунок 1

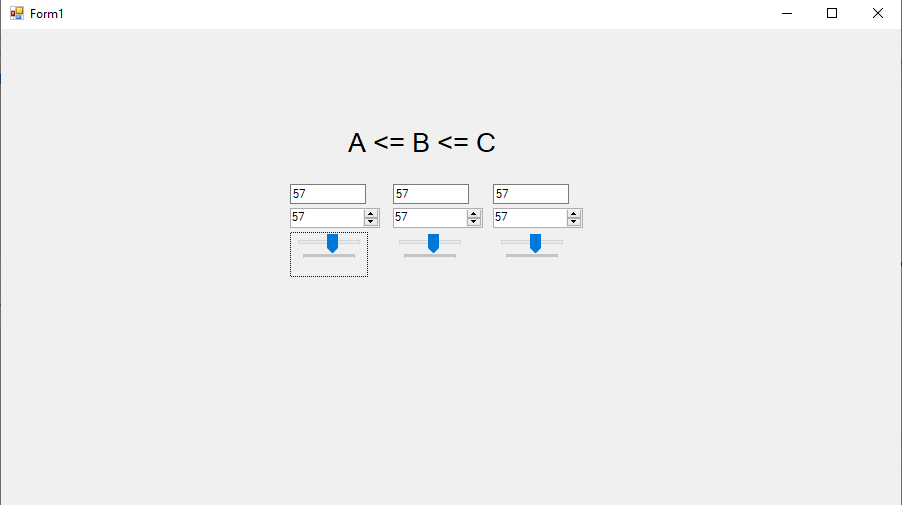


Рисунок 2

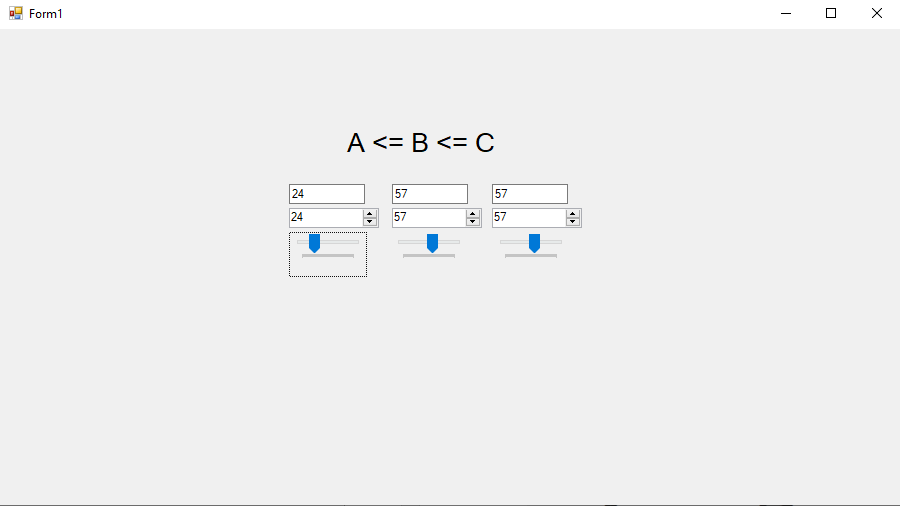
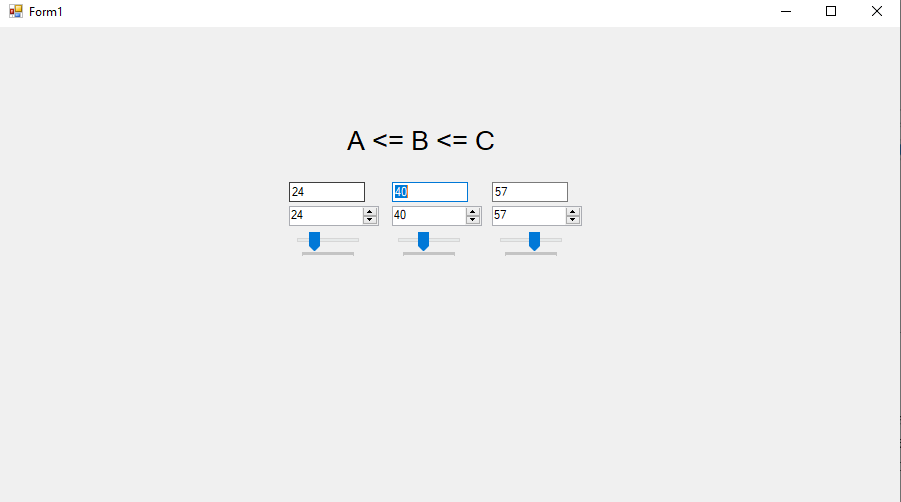


Рисунок 3

1. **Функции ввода числа B.** Ввод числа B при нарушении ограничений, ставит либо наименьшее возможное, либо наибольшее в зависимости от того, ограничения какого из чисел A и C были нарушены. Показано на рисунке4,5,6 и 7 Рисунок 4

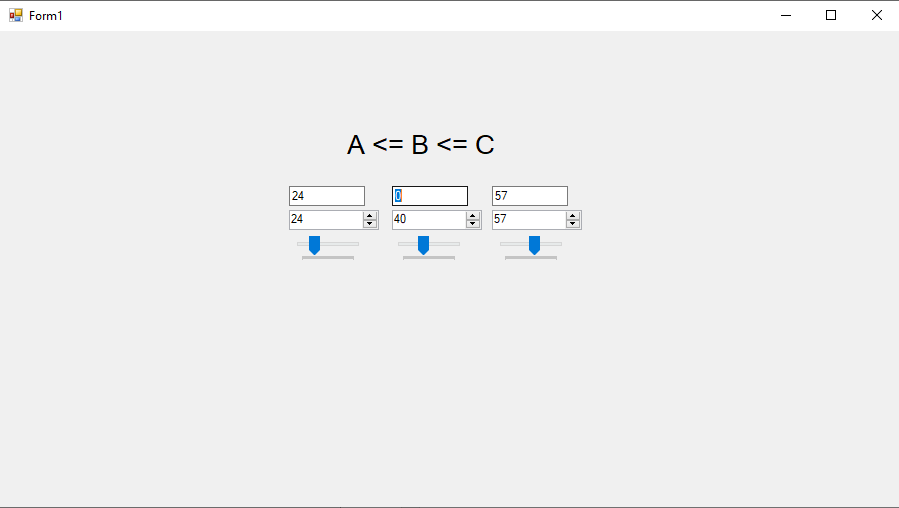


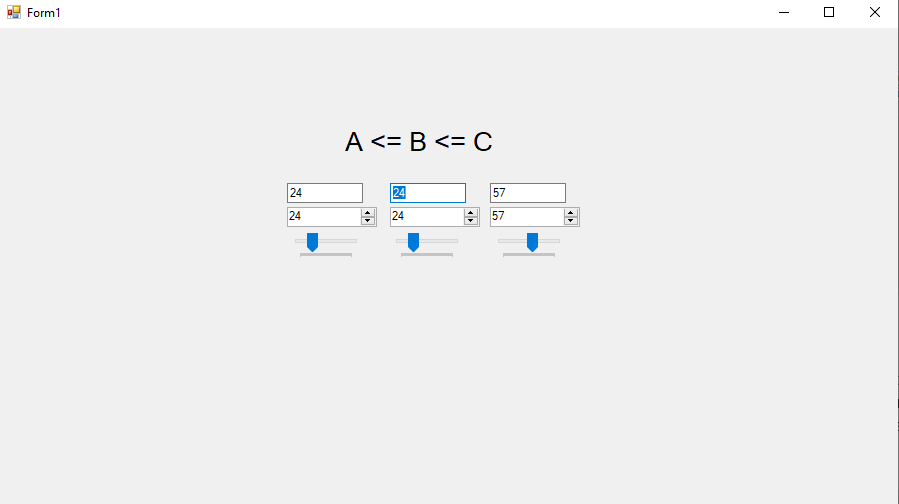
Рисунок 5

Рисунок 6

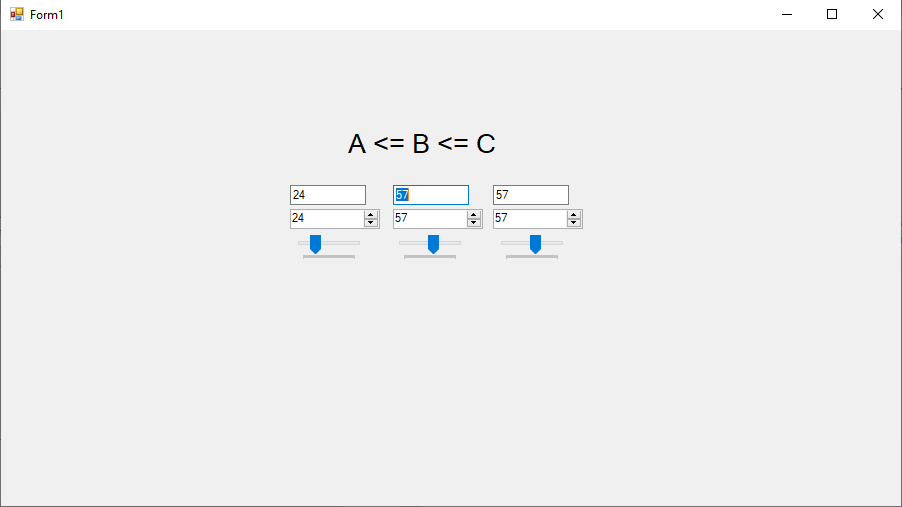


Рисунок 7

1. **Функция ввода числа C.** Ввод числа С при нарушении ограничений порядка модель сама перестраивается, чтобы их выполнить. Показана на рисунке 8,9,10

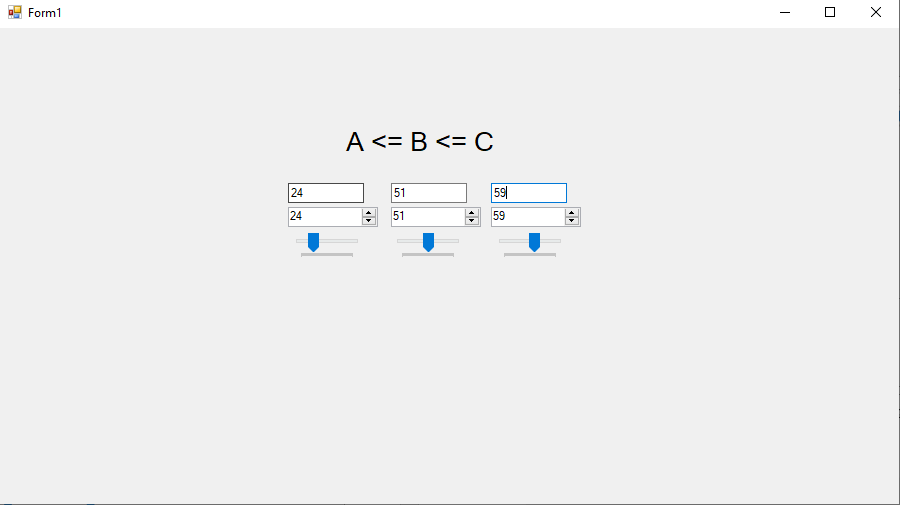


Рисунок 8

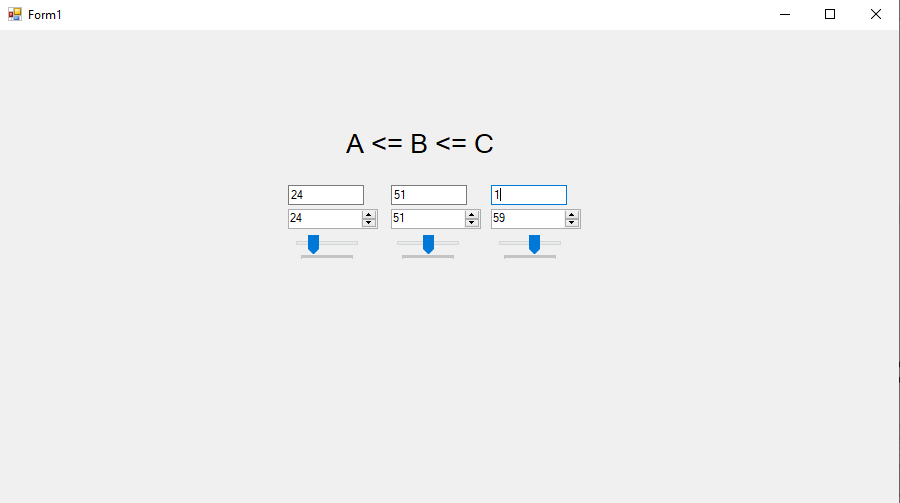


Рисунок 9

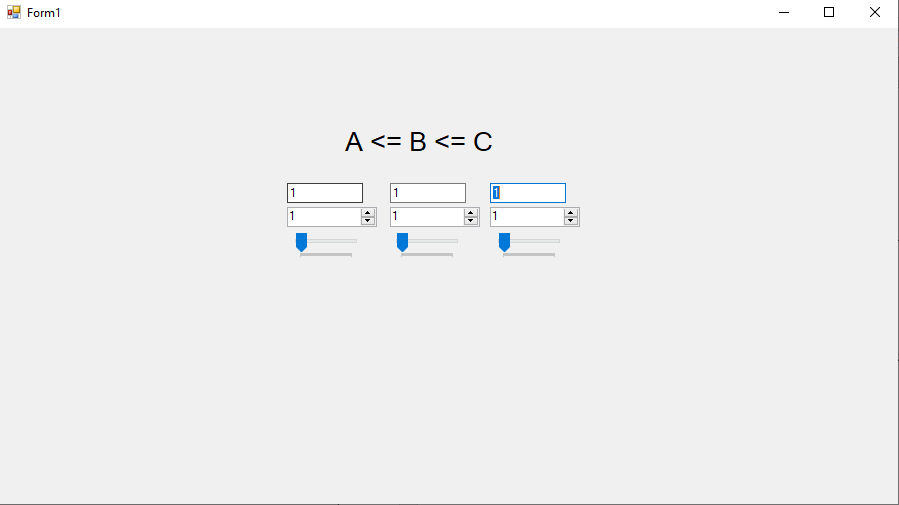


Рисунок 10

**Заключение**

Таким образом была изучена схема разделения данных приложения MVC.

**Листинг программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace лаба4\_2\_ООП

{

public partial class Form1 : Form

{

Model model;

public Form1()

{

InitializeComponent();

model = new Model();

model.observers += new System.EventHandler (this.UpdateFromModel);

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

model.SetValue1(Decimal.ToInt32(numericUpDown1.Value));

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

model.SetValue1(Decimal.ToInt32(trackBar1.Value));

}

private void textBox1\_KeyDown(object sender, KeyEventArgs e)

{

if(e.KeyCode == Keys.Enter)

model.SetValue1(Convert.ToInt32(textBox1.Text));

}

private void numericUpDown2\_ValueChanged(object sender, EventArgs e)

{

model.SetValue2(Decimal.ToInt32(numericUpDown2.Value));

}

private void trackBar2\_Scroll(object sender, EventArgs e)

{

model.SetValue2(Decimal.ToInt32(trackBar2.Value));

}

private void textBox2\_KeyDown(object sender, KeyEventArgs e)

{

if(e.KeyCode == Keys.Enter)

model.SetValue2(Convert.ToInt32(textBox2.Text));

}

private void numericUpDown3\_ValueChanged(object sender, EventArgs e)

{

model.SetValue3(Decimal.ToInt32(numericUpDown3.Value));

}

private void trackBar3\_Scroll(object sender, EventArgs e)

{

model.SetValue3(Decimal.ToInt32(trackBar3.Value));

}

private void textBox3\_KeyDown(object sender, KeyEventArgs e)

{

if(e.KeyCode == Keys.Enter)

model.SetValue3(Convert.ToInt32(textBox3.Text));

}

private void UpdateFromModel(object sender, EventArgs e)

{

textBox1.Text = model.GetValue1().ToString();

numericUpDown1.Value = model.GetValue1();

trackBar1.Value = model.GetValue1();

textBox2.Text = model.GetValue2().ToString();

numericUpDown2.Value = model.GetValue2();

trackBar2.Value = model.GetValue2();

textBox3.Text = model.GetValue3().ToString();

numericUpDown3.Value = model.GetValue3();

trackBar3.Value = model.GetValue3();

}

private void Form1\_FormClosed(object sender, FormClosedEventArgs e)

{

if (File.Exists("text.txt"))

File.Delete("Text.txt");

StreamWriter sw = new StreamWriter("text.txt");

sw.WriteLine(model.GetValue1().ToString());

sw.WriteLine(model.GetValue2().ToString());

sw.WriteLine(model.GetValue3().ToString());

sw.Close();

}

private void Form1\_Load(object sender, EventArgs e)

{

if (File.Exists("text.txt"))

{

StreamReader sr = new StreamReader("text.txt");

model.SetValue1(Convert.ToInt32 (sr.ReadLine()));

model.SetValue2(Convert.ToInt32(sr.ReadLine()));

model.SetValue3(Convert.ToInt32(sr.ReadLine()));

sr.Close();

}

}

private void label1\_Click\_1(object sender, EventArgs e)

{

}

}

}

public class Model

{

int Value1, Value2, Value3;

public System.EventHandler observers;

public Model()

{

Value1 = 0;

Value2 = 50;

Value3 = 100;

}

public void SetValue1(int Value)

{

this.Value1 = Value;

if (this.Value1 > 100) {

while (this.Value1 > 100)

{

this.Value1--;

}

}

if (this.Value1 < 0)

{

while (this.Value1 < 0)

{

this.Value1++;

}

}

if (this.Value1 > this.Value3)

{

while (this.Value1 > this.Value3 && this.Value3 < 100)

{

this.Value3++;

}

}

if (this.Value1 > this.Value2)

{

while (this.Value1 > this.Value2 && this.Value2 < this.Value3)

{

this.Value2++;

}

}

observers.Invoke(this, null);

}

public void SetValue2(int Value)

{

this.Value2 = Value;

if (this.Value2 > 100)

{

while (this.Value2 > 100)

{

this.Value2--;

}

}

if (this.Value2 < this.Value1)

{

while (this.Value2 < this.Value1 && this.Value2 < this.Value3)

{

this.Value2++;

}

}

if (this.Value2 > this.Value3)

{

while (this.Value2 > this.Value3 && this.Value2 > this.Value1)

{

this.Value2--;

}

}

observers.Invoke(this, null);

}

public void SetValue3(int Value)

{

this.Value3 = Value;

if (this.Value3 > 100)

{

while (this.Value3 > 100)

{

this.Value3--;

}

}

if (this.Value3 < 0)

{

while (this.Value3 < 0)

{

this.Value3++;

}

}

if (this.Value3 < this.Value1)

{

while (this.Value3 < this.Value1 && this.Value1 > 0)

{

this.Value1--;

}

}

if (this.Value3 < this.Value2)

{

while (this.Value3 < this.Value2 && this.Value2 > this.Value1)

{

this.Value2--;

}

}

observers.Invoke(this, null);

}

public int GetValue1()

{

return this.Value1;

}

public int GetValue2()

{

return this.Value2;

}

public int GetValue3()

{

return this.Value3;

}

};