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»

Выполнила:

Студентка группы ПРО-229

Яппарова А.Р.

Проверил:

доцент кафедры ВМиК

Котельников В.А.

**Цель**: Создание и разработка простейшего приложения с GUI, реализованное в стиле MVC.

**Задачи:**

• Создать простейшее приложение с GUI, содержащее:

o три целых числа A, B и C со значением в пределах от 0 до 100

o каждое из чисел должно отображаться и редактироваться в 3 разных компонентах: в textBox, numericUpDown, trackBar (или аналогичных в других языках), при этом редактирование числа в одном поле должно приводить к изменению отображения этого числа во всех других полях

o первое и третье число могут иметь значение в пределах от 0 до 100

o второе число всегда должно быть не меньше первого и не больше третьего

o приложение должно сохранять значения чисел между запусками (запоминать при закрытии и восстанавливать значения при открытии)

• Разработанное приложение должно быть реализовано в стиле MVC:

o хранение трёх чисел должно быть организовано в виде отдельного объекта-модели

o все пересчёты и проверки должны выполняться в объекте-модели

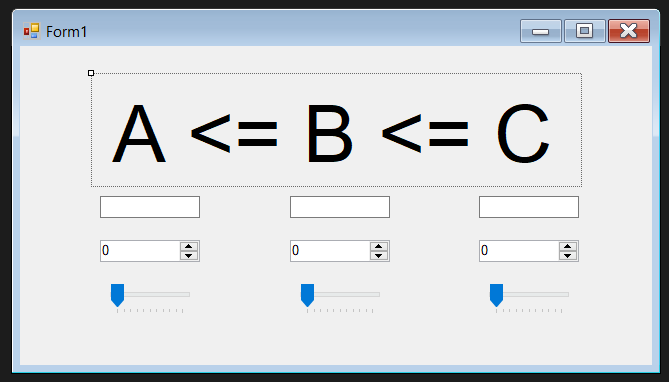
o изменение A и C должно реализовывать разрешающее поведение (при нарушении ограничений порядка модель сама перестраивается, чтобы их выполнить, введённое пользователем значение A и C сохраняется),

o изменение B должно реализовывать запрещающее поведение (при нарушении ограничений порядка модель откатывает внесённые пользователем изменения).

**Ход работы:**

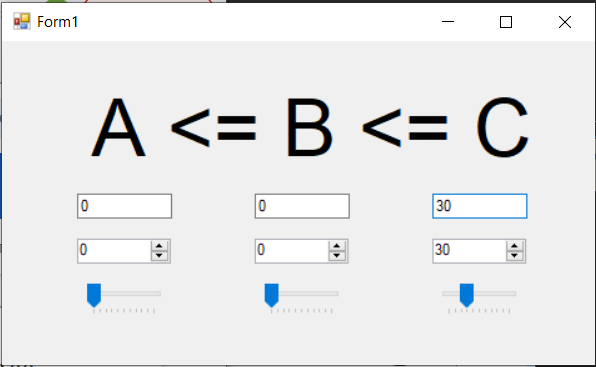
1. Создание и реализация чисел:

Неравенство: A <= B <= C представлено в форме label1, так же реализованы компоненты textBox, numericUpDown, trackBar для каждого из чисел:



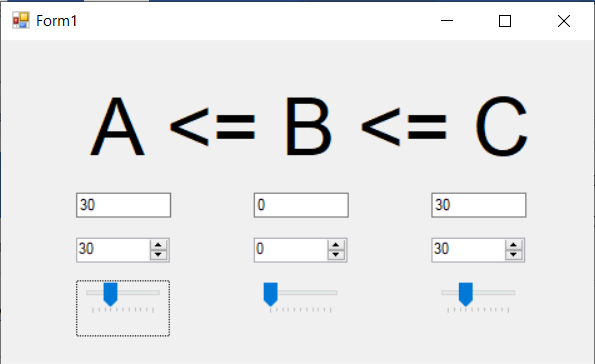
1. Функциональность программы:

Редактирование числа в одном из трех его компонентах приводит к изменению отображения в остальных:

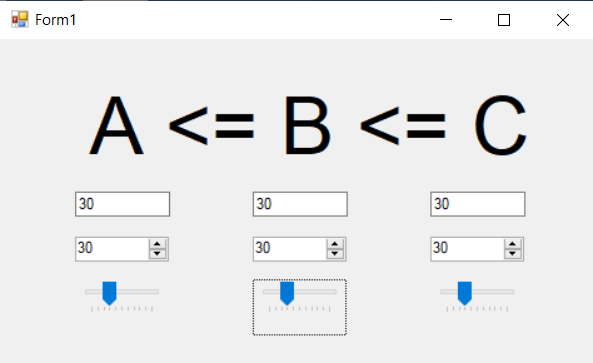


На рисунке можно увидеть, как при попытке изменить значение в textBox3 значения в numericUpDown3 и trackBar3 так же меняются.

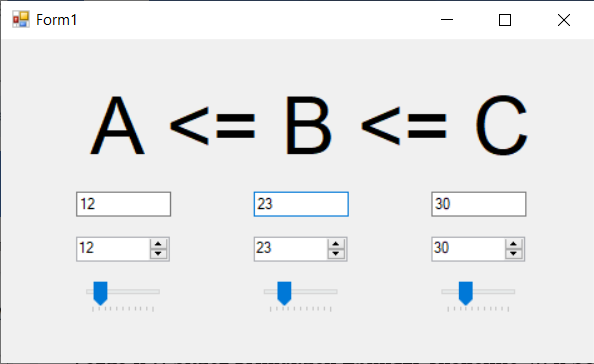
По условию задачи значения первого и третьего числа могут меняться в пределах от 0 до 100.   
Если установить значения в С 30, то значение А не сможет превышать 30, это и реализовано в программе:



Тогда и В будет вынужден принять значение 30 и больше никакое:

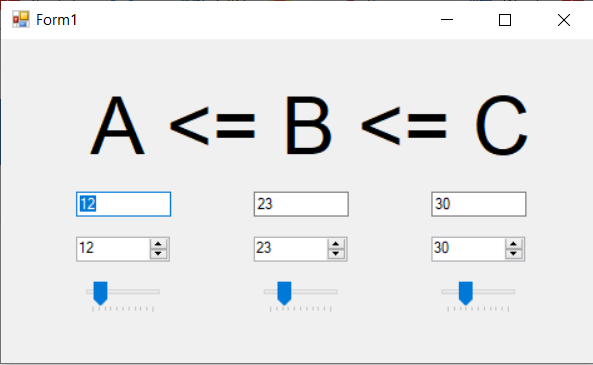


Если А будет равен, к примеру 12, то В сможет принимать любое значение с промежутка [12;30]:

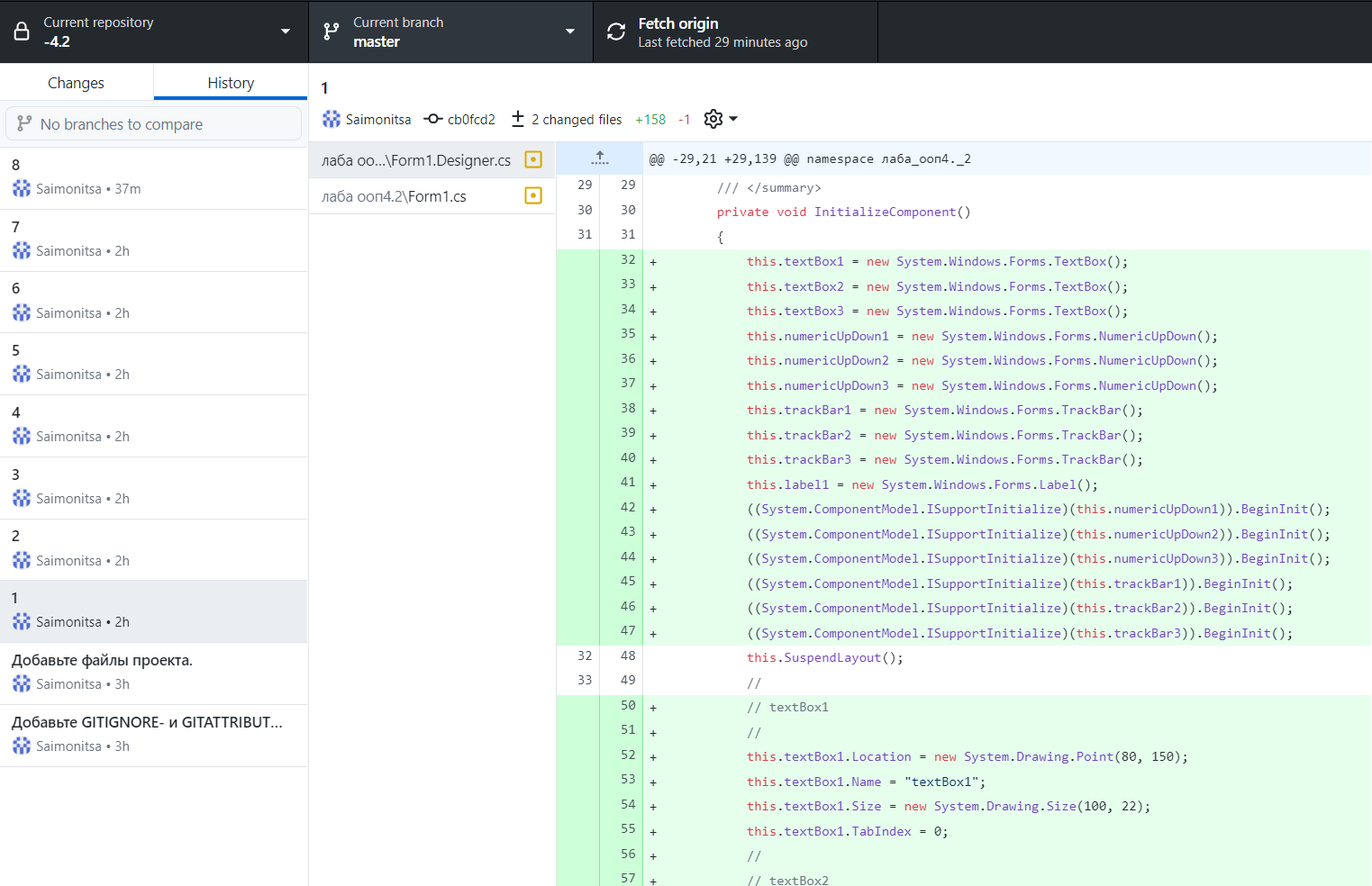


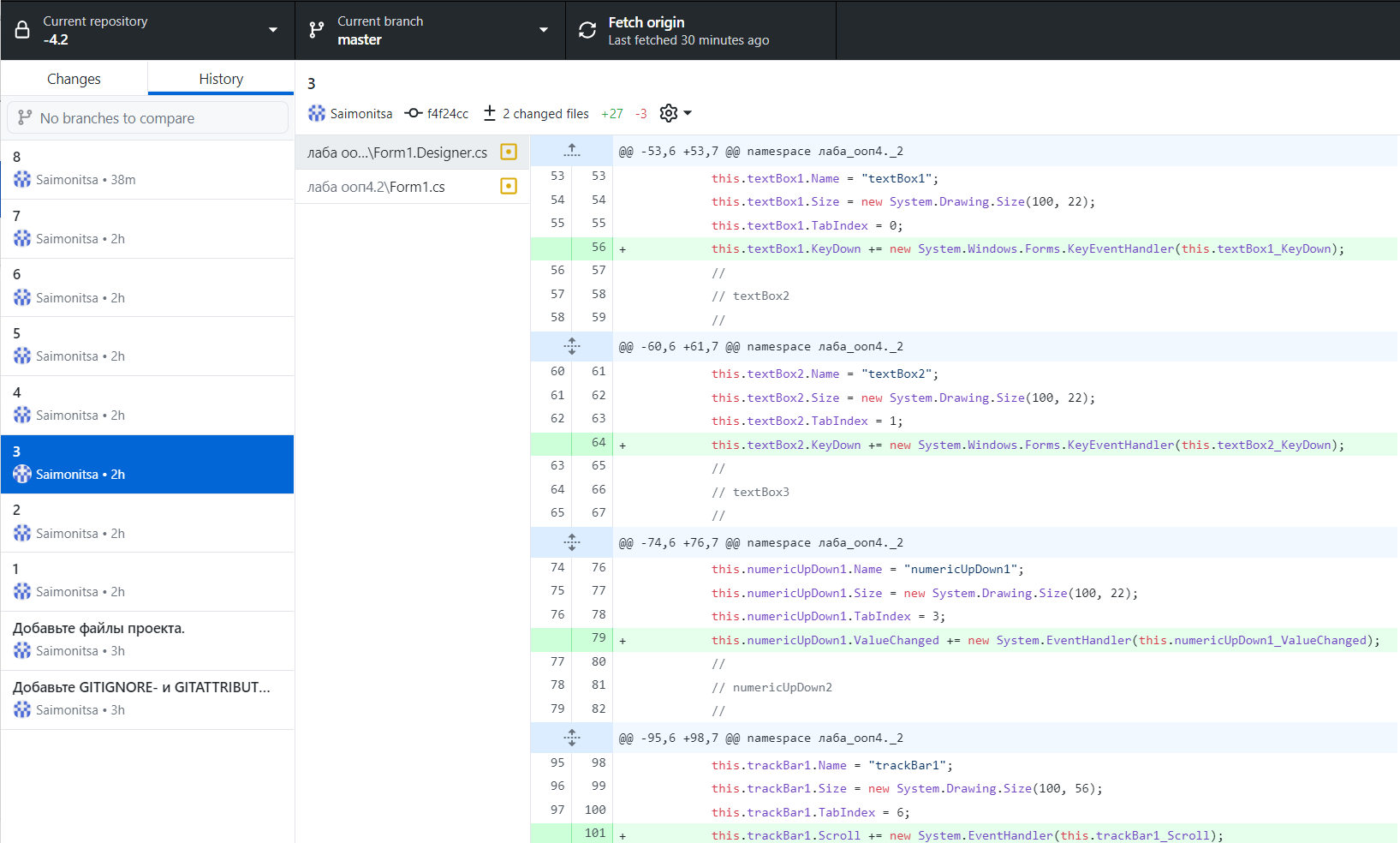
1. Сохранение значений:

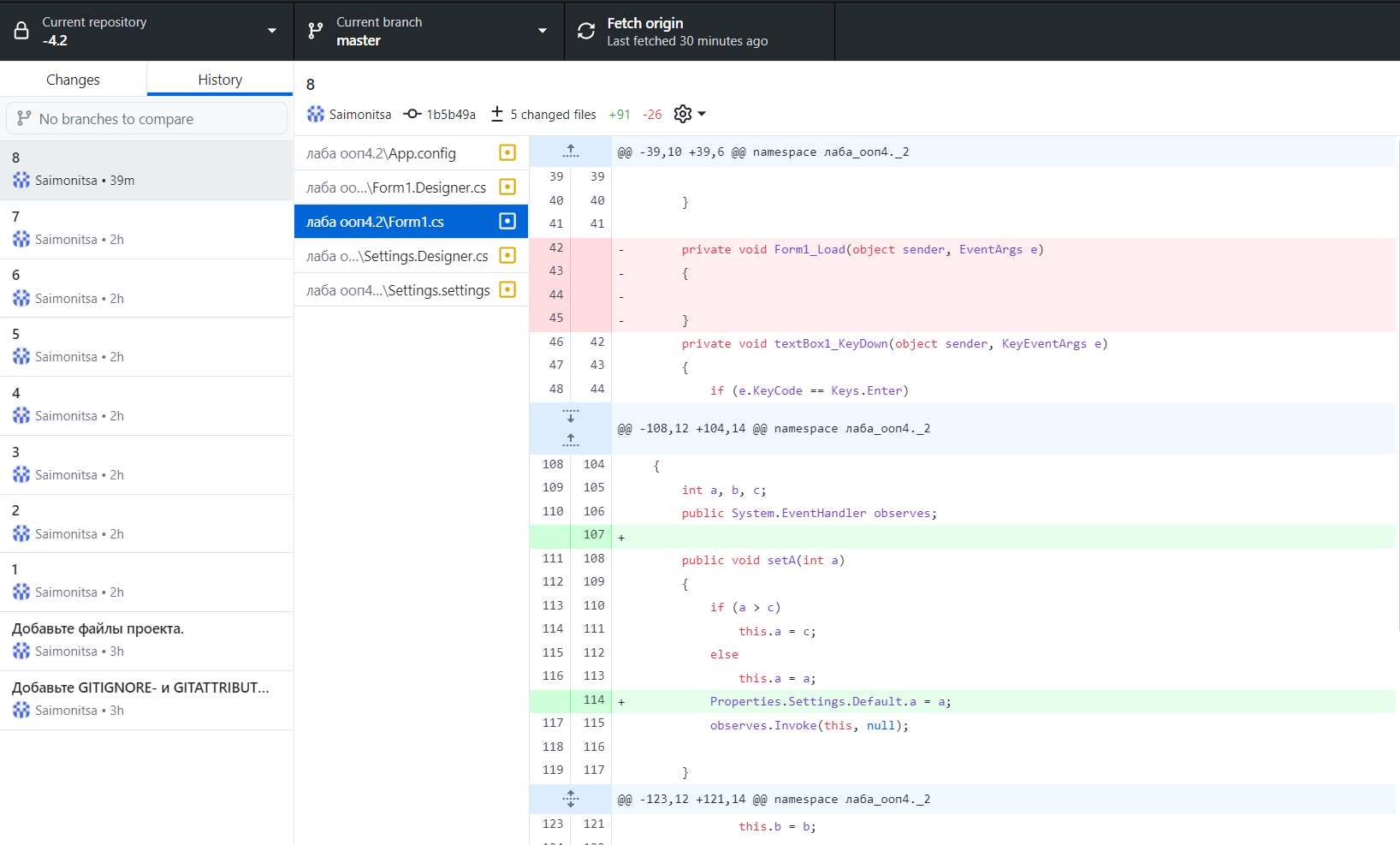
При закрытии и запуске программы значения сохраняются такими же, как и при последних изменениях:



**История коммитов разработки:**

****

****

****

**Заключение:** в ходе лабораторной работы мы научились создавать простейшее приложение с GUI, реализованное в стиле MVC и работать с ним.

**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;

namespace лаба\_ооп4.\_2

{

public partial class Form1 : Form

{

Model model;

public Form1()

{

InitializeComponent();

model = new Model();

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

trackBar1.Scroll += trackBar1\_Scroll;

trackBar2.Scroll += trackBar2\_Scroll;

trackBar3.Scroll += trackBar3\_Scroll;

textBox1.DataBindings.Add("Text", Properties.Settings.Default, "a");

textBox2.DataBindings.Add("Text", Properties.Settings.Default, "b");

textBox3.DataBindings.Add("Text", Properties.Settings.Default, "c");

numericUpDown1.DataBindings.Add("Text", Properties.Settings.Default, "a");

numericUpDown2.DataBindings.Add("Text", Properties.Settings.Default, "b");

numericUpDown3.DataBindings.Add("Text", Properties.Settings.Default, "c");

trackBar1.DataBindings.Add("Text", Properties.Settings.Default, "a");

trackBar2.DataBindings.Add("Text", Properties.Settings.Default, "b");

trackBar3.DataBindings.Add("Text", Properties.Settings.Default, "c");

this.FormClosing += (o, e) => Properties.Settings.Default.Save();

}

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

{

if (e.KeyCode == Keys.Enter)

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

}

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

{

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

}

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

{

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

}

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

{

if (e.KeyCode == Keys.Enter)

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

}

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

{

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

}

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

{

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

}

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

{

if (e.KeyCode == Keys.Enter)

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

}

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

{

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

}

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

{

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

}

private void UpdateFromModel(object sender, EventArgs e)

{

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

numericUpDown1.Value = model.getA();

trackBar1.Value = model.getA();

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

numericUpDown2.Value = model.getB();

trackBar2.Value = model.getB();

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

numericUpDown3.Value = model.getC();

trackBar3.Value = model.getC();

}

}

public class Model

{

int a, b, c;

public System.EventHandler observes;

public void setA(int a)

{

if (a > c)

this.a = c;

else

this.a = a;

Properties.Settings.Default.a = a;

observes.Invoke(this, null);

}

public void setB(int b)

{

if (b <= c && b >= a)

this.b = b;

else

this.b = a;

Properties.Settings.Default.b = b;

observes.Invoke(this, null);

}

public void setC(int c)

{

this.c = c;

Properties.Settings.Default.c = c;

observes.Invoke(this, null);

}

public int getA()

{

return a;

}

public int getB()

{

return b;

}

public int getC()

{

return c;

}

}

}