ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»

в г. Смоленске

Кафедра вычислительной техники:

Отчет

По лабораторной работе № 6

Тема: «Простейшие классы»

по курсу: «Технология программирования»

Студент: Иванов В.В.

Группа: ИВТ1-18

Преподаватель: Малашенкова И.В.

Вариант: 7

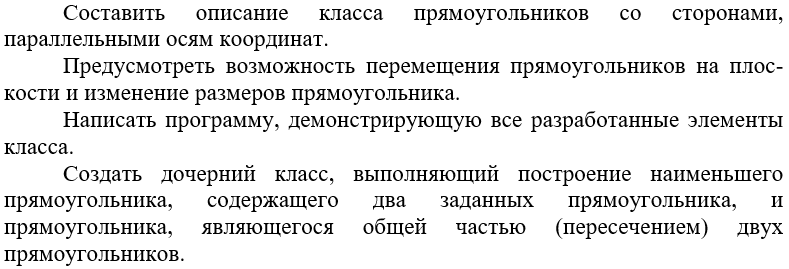
г. Смоленск ,2020

**Цель**:

1.Создать класс.

2.Создать дочерний класс.

1. **Задание на выполнение:**



**2.** **Анализ задания**: в данной лабораторной работе необходимо создать класс и дочерний класс и описать их методы. Создание прямоугольников возможность изменения размеров прямоугольников. Возможность построения общих прямоугольников.

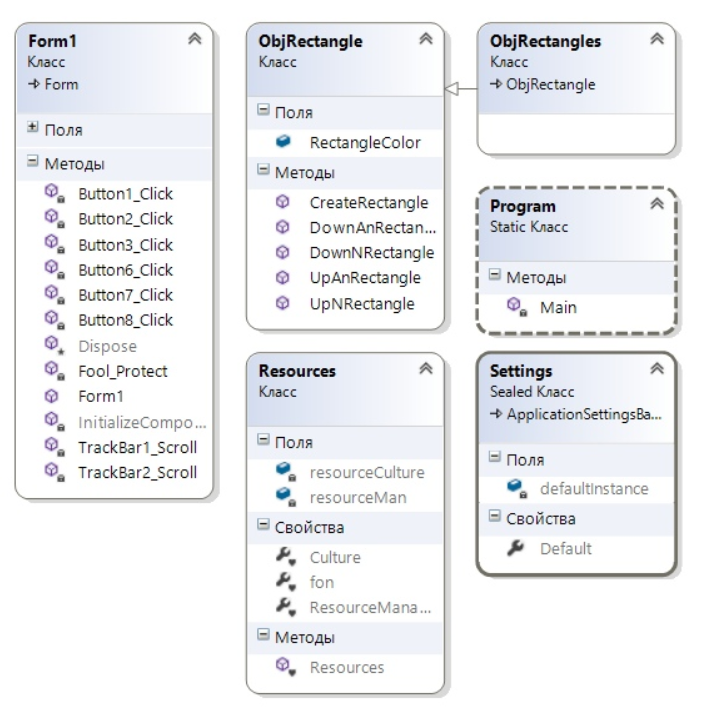


Рисунок 2 – Диаграмма классов

**3. Схема алгоритма**

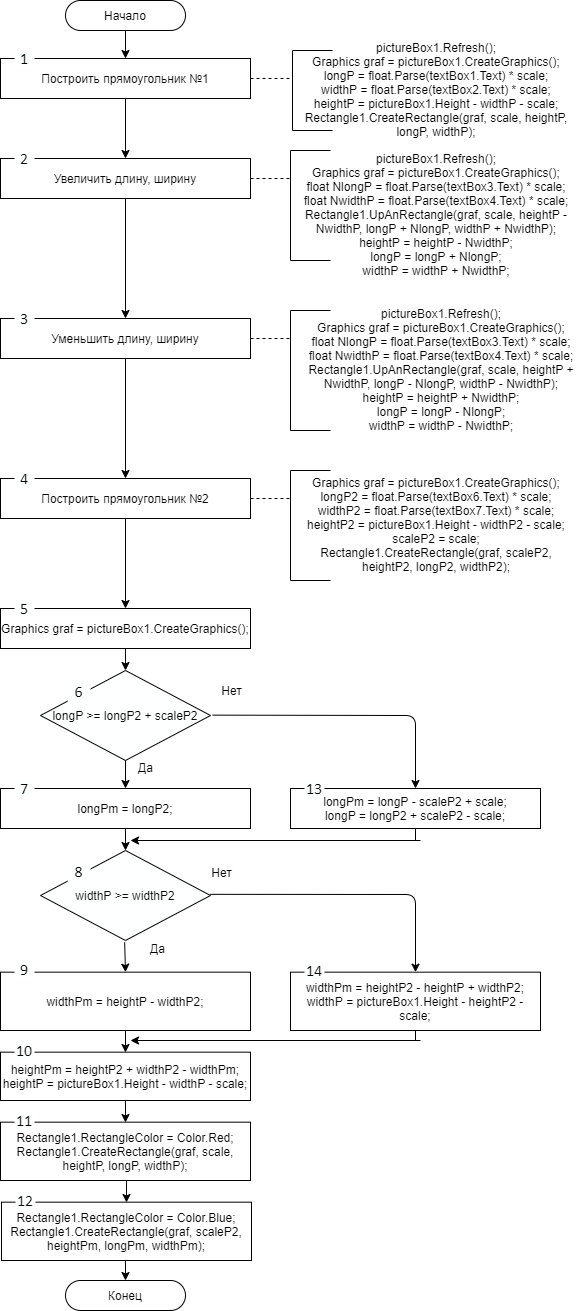


Рисунок 3.1 – Алгоритм программы

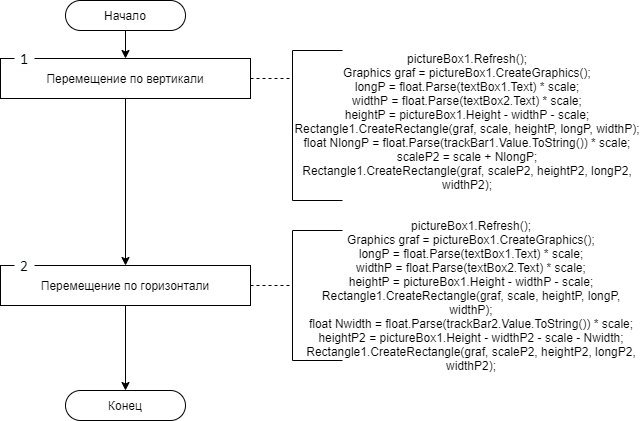


Рисунок 3.2 – Алгоритм перемещения прямоугольника

**4. Тестирование программы**

Таблица 1 – Тестирование программы

|  |  |  |
| --- | --- | --- |
| Набор данных | Ожидаемый результат | Результат |
| Корректный ввод | Вывод результата |  |
| Некорректный ввод (невозможен) | – | – |

**5. Экранные формы**

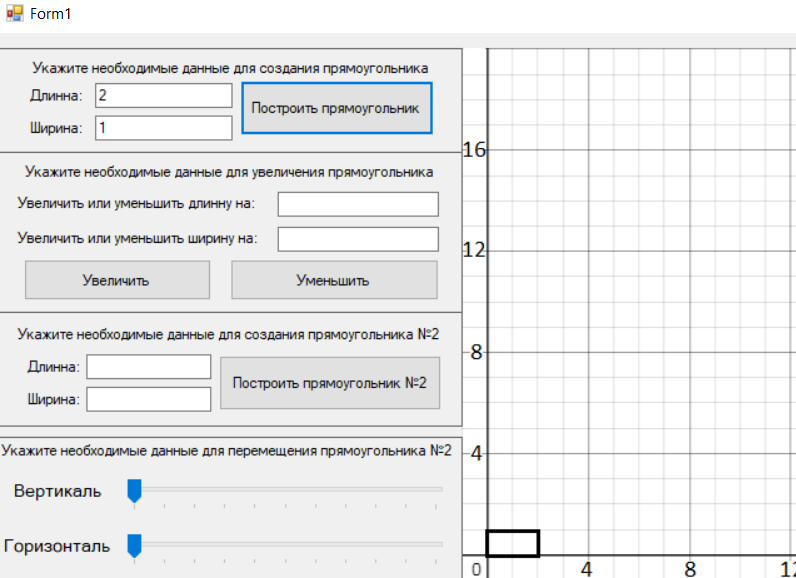


Рисунок 5.1 – Экранная форма. Создание прямоугольника №1

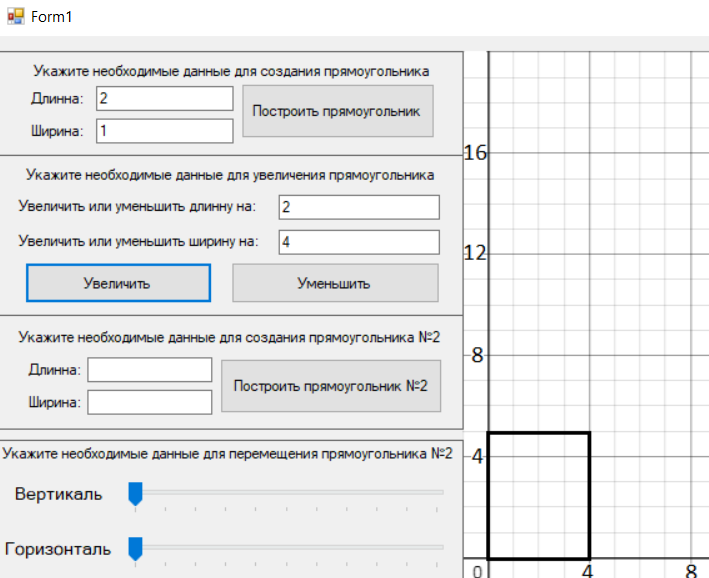


Рисунок 5.2 – Экранная форма. Увеличить длину, ширину

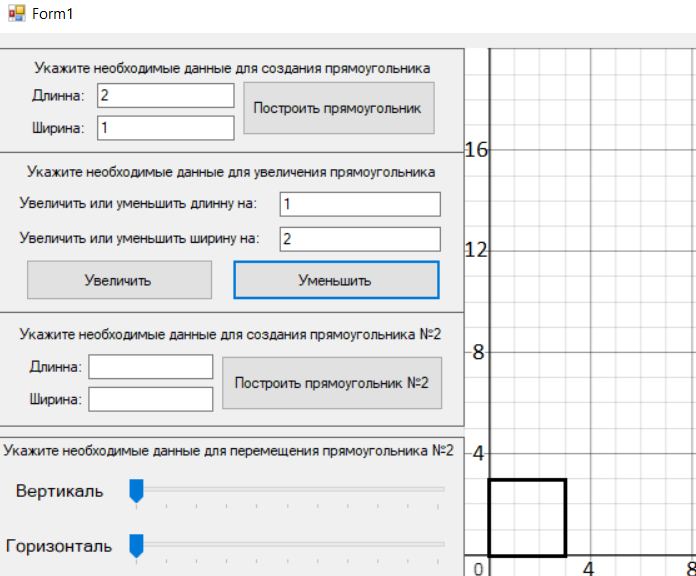


Рисунок 5.3 – Экранная форма. Уменьшить длину, ширину

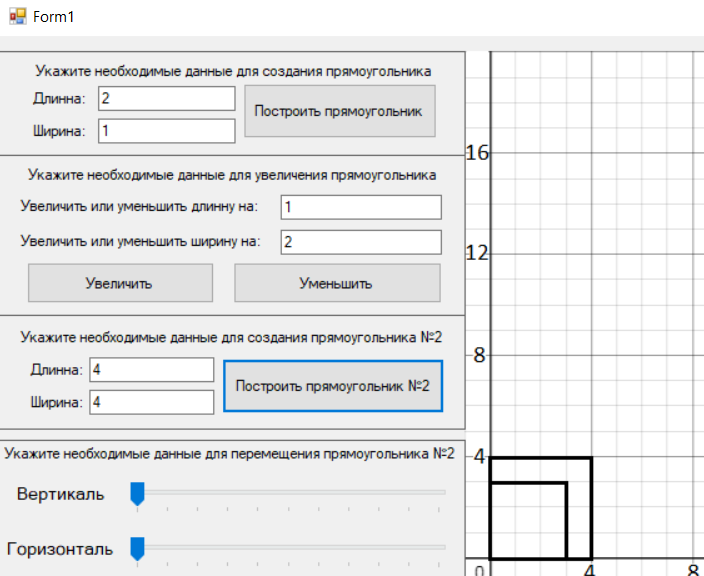


Рисунок 5.4 – Экранная форма. Построение прямоугольника №2

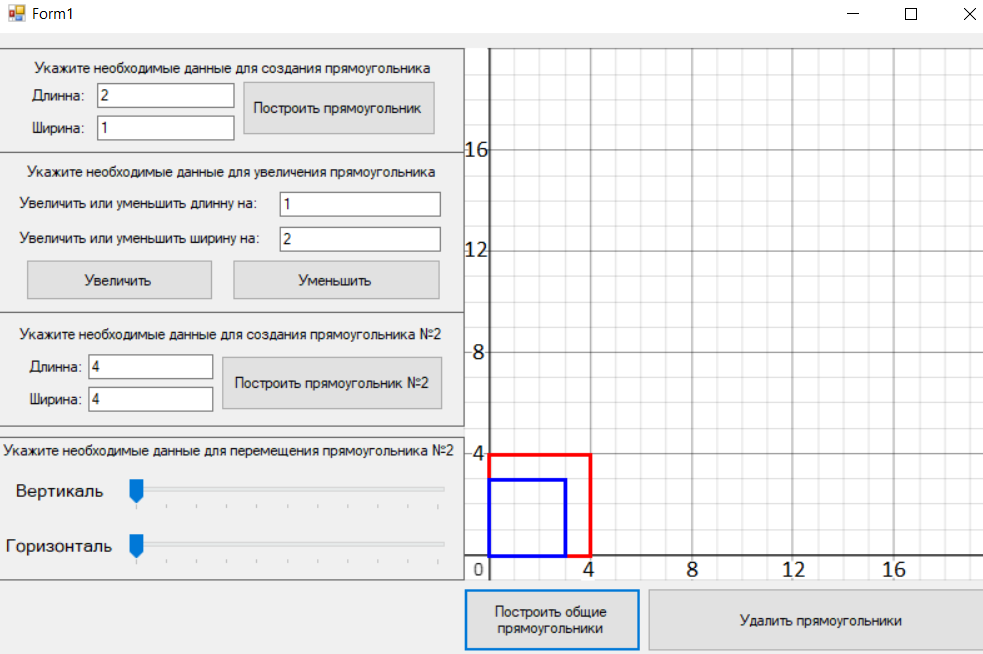


Рисунок 5.5 – Экранная форма. Построить общие прямоугольники.

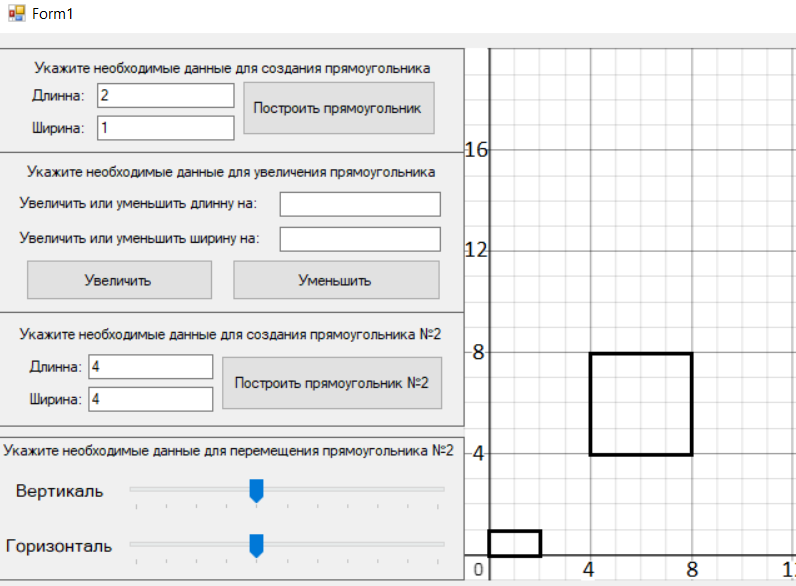


Рисунок 5.6 – Экранная форма. Перемещение прямоугольника.

**Вывод:** в ходе выполнения работы получен опыт в работе с классами C#. Выполнены поставленные задачи.

**Приложение А. Программный код**

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.Drawing.Imaging;

namespace Лр6

{

public partial class Form1 : Form

{

const float scale = 20.2F;

public Form1()

{

InitializeComponent();

trackBar1.Scroll += TrackBar1\_Scroll;

trackBar2.Scroll += TrackBar2\_Scroll;

}

ObjRectangle Rectangle1 = new ObjRectangle();

float longP2;

float widthP2;

float scaleP2;

float heightP2;

float longP;

float widthP;

float longPm;

float widthPm;

float heightP;

float heightPm;

private void Button1\_Click(object sender, EventArgs e)

{

pictureBox1.Refresh();

Graphics graf = pictureBox1.CreateGraphics();

longP = float.Parse(textBox1.Text) \* scale;

widthP = float.Parse(textBox2.Text) \* scale;

heightP = pictureBox1.Height - widthP - scale;

Rectangle1.CreateRectangle(graf, scale, heightP, longP, widthP);

}

private void Button2\_Click(object sender, EventArgs e)

{

pictureBox1.Image = null;

Rectangle1.RectangleColor = Color.Black;

}

private void Button3\_Click(object sender, EventArgs e)

{

pictureBox1.Refresh();

Graphics graf = pictureBox1.CreateGraphics();

float NlongP = float.Parse(textBox3.Text) \* scale;

float NwidthP = float.Parse(textBox4.Text) \* scale;

Rectangle1.UpAnRectangle(graf, scale, heightP - NwidthP, longP + NlongP, widthP + NwidthP);

heightP = heightP - NwidthP;

longP = longP + NlongP;

widthP = widthP + NwidthP;

}

private void Button6\_Click(object sender, EventArgs e)

{

pictureBox1.Refresh();

Graphics graf = pictureBox1.CreateGraphics();

float NlongP = float.Parse(textBox3.Text) \* scale;

float NwidthP = float.Parse(textBox4.Text) \* scale;

Rectangle1.UpAnRectangle(graf, scale, heightP + NwidthP, longP - NlongP, widthP - NwidthP);

heightP = heightP + NwidthP;

longP = longP - NlongP;

widthP = widthP - NwidthP;

}

private void Button7\_Click(object sender, EventArgs e)

{

Graphics graf = pictureBox1.CreateGraphics();

longP2 = float.Parse(textBox6.Text) \* scale;

widthP2 = float.Parse(textBox7.Text) \* scale;

heightP2 = pictureBox1.Height - widthP2 - scale;

scaleP2 = scale;

Rectangle1.CreateRectangle(graf, scaleP2, heightP2, longP2, widthP2);

}

private void Fool\_Protect(object sender, KeyPressEventArgs e) //Относительно универсальный ввод только чисел в текстовое поле

{

TextBox senderTextBox = sender as TextBox;

char number = e.KeyChar; //Нажатая кнопка.

if (!Char.IsDigit(number) && number != 8)

{

e.Handled = true; // Ввод отменяется

}

}

private void Button8\_Click(object sender, EventArgs e)

{

Graphics graf = pictureBox1.CreateGraphics();

if (longP >= longP2 + scaleP2)

{

longPm = longP2;

}

else

{

longPm = longP - scaleP2 + scale;

longP = longP2 + scaleP2 - scale;

}

if (widthP >= widthP2)

{

widthPm = heightP - widthP2;

}

else

{

widthPm = heightP2 - heightP + widthP2;

widthP = pictureBox1.Height - heightP2 - scale;

}

heightPm = heightP2 + widthP2 - widthPm;

heightP = pictureBox1.Height - widthP - scale;

Rectangle1.RectangleColor = Color.Red;

Rectangle1.CreateRectangle(graf, scale, heightP, longP, widthP);

Rectangle1.RectangleColor = Color.Blue;

Rectangle1.CreateRectangle(graf, scaleP2, heightPm, longPm, widthPm);

}

private void TrackBar1\_Scroll(object sender, EventArgs e)

{

pictureBox1.Refresh();

Graphics graf = pictureBox1.CreateGraphics();

longP = float.Parse(textBox1.Text) \* scale;

widthP = float.Parse(textBox2.Text) \* scale;

heightP = pictureBox1.Height - widthP - scale;

Rectangle1.CreateRectangle(graf, scale, heightP, longP, widthP);

float NlongP = float.Parse(trackBar1.Value.ToString()) \* scale;

scaleP2 = scale + NlongP;

Rectangle1.CreateRectangle(graf, scaleP2, heightP2, longP2, widthP2);

}

private void TrackBar2\_Scroll(object sender, EventArgs e)

{

pictureBox1.Refresh();

Graphics graf = pictureBox1.CreateGraphics();

longP = float.Parse(textBox1.Text) \* scale;

widthP = float.Parse(textBox2.Text) \* scale;

heightP = pictureBox1.Height - widthP - scale;

Rectangle1.CreateRectangle(graf, scale, heightP, longP, widthP);

float Nwidth = float.Parse(trackBar2.Value.ToString()) \* scale;

heightP2 = pictureBox1.Height - widthP2 - scale - Nwidth;

Rectangle1.CreateRectangle(graf, scaleP2, heightP2, longP2, widthP2);

}

}

}