Московский авиационный институт

(национальный исследовательский университет)

**Факультет прикладной математики и физики**

Кафедра вычислительной математики и программирования

**Лабораторные работы 1 — 16 по курсу ИСРППС:**

**Программирование интерактивного интерфейса и применение**

**паттернов на языке С#**

№1. Обработка исключений, преобразование типов

№2. Паттерны MVC, Facade

№3. Delegate Pattern, PictureBox

№4. Компонента MessageBox

№5. Паттерны Abstract Factory и Singleton

№6. Buttons, CheckBox, Labels, Scrolls, Track

№7. Паттерн Chain-responsibility

№8. Паттерн Command

№9. Menus, ToolStrip and StatusStrip

№10. Dialog Boxes, ClipBoard

№11. Tree View, List View

№12. Drag and Drop, Clip

№13. Lines, Curves and Area Fills.

№14. Обработка прерываний от таймера

№15. Компонента dll

№16. Типы проектов .Net

Работу выполнил:

М8О-205Б-19 Васильев Александр Владиславович

Руководитель: \_\_\_\_\_\_\_\_/Семенов А.С./

Подпись:

Дата: \_\_\_\_\_ 2021

№1. Обработка исключений, преобразование типов

using System;

namespace ISRPPS\_1

{

class Date

{

public int year\_A

{

set

{

if (value < 1 || value > 2020) throw new ArgumentOutOfRangeException("Year is out of range");

else year = value;

}

get { return year; }

}

public int month\_A

{

set

{

if (value < 1 || value >= 13) throw new ArgumentOutOfRangeException("Month is out of range");

else month = value;

}

get { return month; }

}

public int day\_A

{

set

{

if (value < 1 || value >= 32) throw new ArgumentOutOfRangeException("Day is out of range");

else day = value;

}

get { return day;}

private int year = 0;

private int month = 0;

private int day = 0;

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Step 1 : size (byte, int, short, float, double, long, decimal)");

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

Console.WriteLine("size byte: {0} to {1}", byte.MinValue, byte.MaxValue);

Console.WriteLine("size int: {0} to {1}", int.MinValue, int.MaxValue);

Console.WriteLine("size short: {0} to {1}", short.MinValue, short.MaxValue);

Console.WriteLine("size float: {0} to {1}", float.MinValue, float.MaxValue);

Console.WriteLine("size double: {0} to {1}", double.MinValue, double.MaxValue);

Console.WriteLine("size long: {0} to {1}", long.MinValue, long.MaxValue);

Console.WriteLine("size decimal: {0} to {1}", decimal.MinValue, decimal.MaxValue);

Console.WriteLine("");

string inform = "";

Console.WriteLine("input float value");

inform = Console.ReadLine();

var p = Convert.ToDouble(inform);

//var p = Single.Parse(inform.Replace(".", ","));

Console.WriteLine("output value:{0}",p);

Console.WriteLine("Step 2: Massive-> initialization-> output");

Console.WriteLine("");

float[] arr = new float[3] { 2.7f, 5.1f, 8.2f };

foreach (var f in arr) { Console.Write(" {0} ", f); }

Console.WriteLine("");

Console.WriteLine("");

Console.WriteLine("Step 3 : Output Date");

Console.WriteLine("");

Date date = new Date();

for (; ; )

{

Console.WriteLine("input year:");

try

{

date.year\_A = Convert.ToInt32(Console.ReadLine());

break;

}

catch (ArgumentOutOfRangeException) { Console.WriteLine("Out of Range, try again"); }

catch (FormatException) { Console.WriteLine("Wrong format, try again"); };

}

for(; ; )

{

Console.WriteLine("input month:");

try

{

date.month\_A = Convert.ToInt32(Console.ReadLine());

break;

}

catch (ArgumentOutOfRangeException) { Console.WriteLine("Out of Range, try again"); }

catch (FormatException) { Console.WriteLine("Wrong format, try again"); };

}

for (; ; )

{

Console.WriteLine("input day:");

try

{

date.day\_A = Convert.ToInt32(Console.ReadLine());

break;

}

catch (ArgumentOutOfRangeException) { Console.WriteLine("Out of Range, try again"); }

catch (FormatException) { Console.WriteLine("Wrong format, try again"); }

}

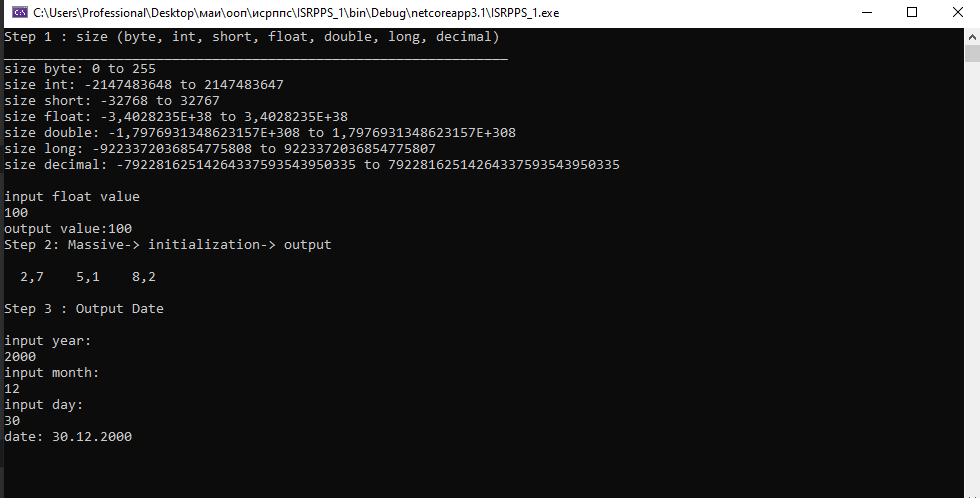
Console.WriteLine("date: {0}.{1}.{2}", date.day\_A, date.month\_A, date.year\_A);

Console.ReadKey();

}

}

}



№2. Паттерны MVC, Façade

Façade:

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1\_ISRPPS\_2\_F

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

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 WindowsFormsApp1\_ISRPPS\_2\_F

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

Facade.Run();

}

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

{

Facade.Run1();

}

public static class Facade

{

static S2 SubSYS2 = new S2();

static S3 SubSYS3 = new S3();

static Form2 f2 = new Form2();

public static void Run()

{

f2.ShowDialog();

f2.Hide();

SubSYS2.Run\_2();

SubSYS3.Run\_3();

}

public static void Run1()

{

SubSYS2.Run\_2();

SubSYS3.Run\_3();

SubSYS2.Run\_2();

}

internal class S2

{

Form2 SubSYS2 = new Form2();

public void Run\_2()

{

SubSYS2.ShowDialog();

}

}

internal class S3

{

Form3 SubSYS3 = new Form3();

public void Run\_3()

{

SubSYS3.ShowDialog();

}

}

}

}

}

**Form2.cs**

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 WindowsFormsApp1\_ISRPPS\_2\_F

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

}

}

**Form3.cs**

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 WindowsFormsApp1\_ISRPPS\_2\_F

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

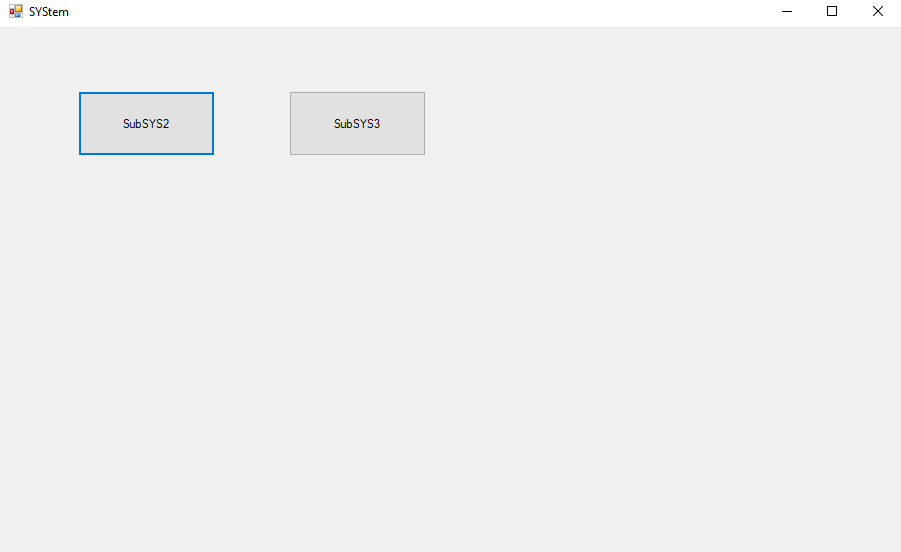
private void button1\_Click(object sender, EventArgs e)

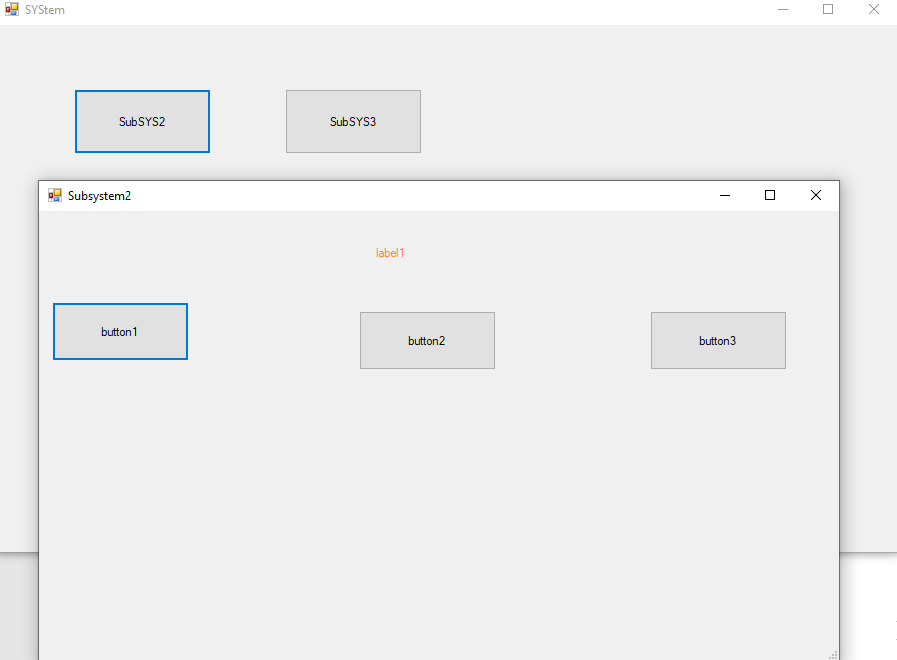
{

}

}

}





**MVC**

**Program.cs(Controller)**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

using WindowsFormsApp1\_ISRPPS\_2.View;

namespace WindowsFormsApp1\_ISRPPS\_2

{

static class Program

{

public static bool fb = true;

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

/// mvc - структурный тип

/// view - отобр информанции, model - алгоритмы, расчёты, контроллер- связующее звено между представлением и моделью

/// достоинства: поддержка нескольких представлений, можем использовать другой интерфейс не изменяя модель или наоборот

/// недостатки:сложность и стоимость частых обновлений

/// обращение:котроллер - модель - отображение

/// паттерн - часто используемая, повторяемая архитектурная конструкция, представляющая собой решение проблем проектирования

/// у них есть названия, типы ( порождающий, структурный, поведенческий), есть достоинства и недостатки и диаграммы классов

///

[STAThread]

static void Main()

{

Model model = new Model();//2

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

//Application.Run(new Form1());//1

Form2 form2 = new Form2();//2

Form1 form1 = new Form1();//2

Application.Run(form1);//2

if (fb)

{

//Application.Run(new Form2());//1

form2.Rungek=model.func();//2

form2.ShowDialog();//2

}

}

}

}

**Model.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1\_ISRPPS\_2

{

public class Model

{

public Model() { }

public void op() { MessageBox.Show("Model 1 operation"); }

public int func() { return 404; }

}

}

**View(Form1.cs)**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using WindowsFormsApp1\_ISRPPS\_2.View;

//Паттерн mvc - структурный

namespace WindowsFormsApp1\_ISRPPS\_2

{

public partial class Form1 : Form

{

//Model model = new Model();//1 - всегда создается при запуске

public Form1()

{

InitializeComponent();

}

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

{

}

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

{

MessageBox.Show("Button1");

//model.op();//1

}

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

{

MessageBox.Show("Button2");

// MessageBox.Show("Model1 fucntion"+" " +"label 1-> "+ model.func()) ;//1

//this.label1.Text = model.func().ToString();//1

Program.fb = false;

}

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

{

MessageBox.Show("label1");

}

}

}

**View(Form2.cs)**

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 WindowsFormsApp1\_ISRPPS\_2.View

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

public int Rungek { set; get; } = 0;//2

private void Form2\_Load(object sender, EventArgs e)

{

}

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

{

MessageBox.Show("" + Rungek);//2

this.label1.Text = Rungek.ToString();//2

if (Rungek >= 999) { this.label1.BackColor = Color.Violet; Rungek = 404; return ; }//2

if (Rungek !=0) this.label1.BackColor = Color.Red;//2

else this.label1.BackColor = Color.Green;//2

Rungek = 0;//2

}

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

{

MessageBox.Show("label: Фиолетовый->красный->зеленый");

Rungek = 999;//2

}

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

{

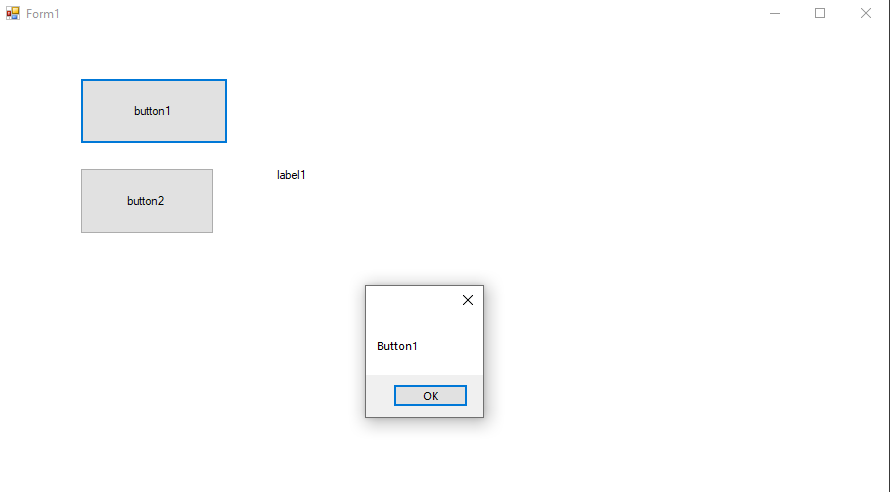
MessageBox.Show("label1: красный->зеленый");

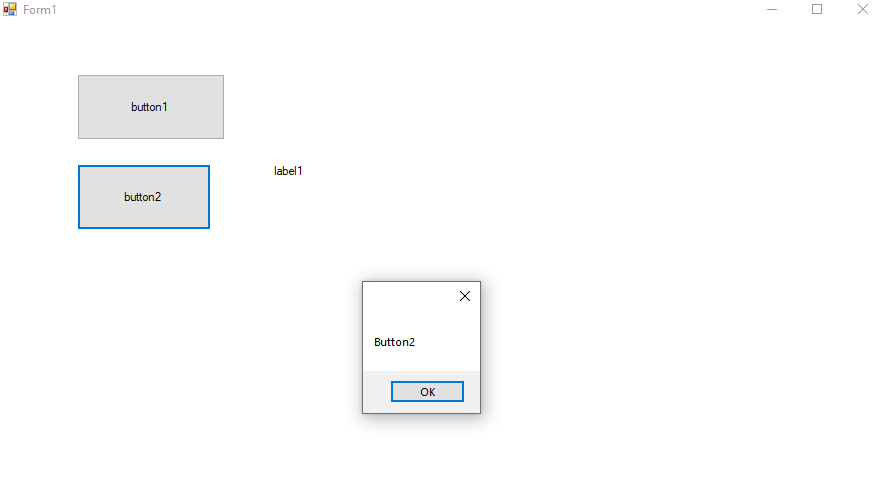
Rungek = 404;//2

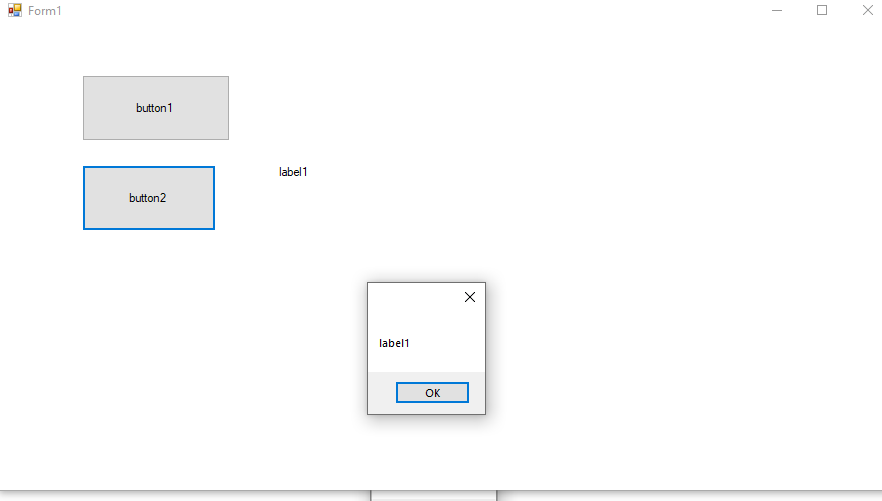
}

}

}







№3. Delegate Pattern, PictureBox

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp1\_ISRPPS\_3

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

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 WindowsFormsApp1\_ISRPPS\_3

{

delegate void DelegateTool(MouseEventArgs e);//делегат тип поведенческий

public partial class Form1 : Form

{

Bitmap bitmap=null;//bitmap для работы с растровыми изображениями

Graphics gr = null;//создаем объект графики, с помощью которого можно рисовать в picturebox

Pen pen = new Pen(Color.DarkBlue, 3.5f);

Pen brush = new Pen(new SolidBrush(Color.IndianRed), 9.5f);

private DelegateTool operation;

public Form1()

{

InitializeComponent();

gr = this.pictureBox1.CreateGraphics();

this.operation = tPen;

this.pictureBox1.Image = this.bitmap;

bitmap = new Bitmap(this.pictureBox1.Width, this.pictureBox1.Height);

gr = Graphics.FromImage(bitmap);

}

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

{

}

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

{

this.gr.Clear(this.pictureBox1.BackColor);

this.pictureBox1.Image = this.bitmap;

//this.pictureBox1.Refresh();

}

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

{

ToolBrush();

}

//методы, которые меняют делегат(инструмент работы)

public void ToolDraw(MouseEventArgs e){operation(e);}

public void ToolBrush() { operation = tBrush; }

public void ToolPen() { operation = tPen; }

private void tPen(MouseEventArgs e) { gr.DrawLine(pen, e.X, e.Y, e.X +1 , e.Y + 1); }

private void tBrush(MouseEventArgs e) { gr.DrawLine(brush, e.X, e.Y, e.X + 1, e.Y + 1); }

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

{

ToolPen();

}

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

{

DrawWeb();

}

public void DrawWeb() { operation = tDrawWeb; }

private void tDrawWeb(MouseEventArgs e)

{

int step =4;

this.gr.Clear(this.pictureBox1.BackColor);

this.pictureBox1.Image = this.bitmap;

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

{

PointF pointf = new PointF(0, (i \* (this.pictureBox1.Size.Height)) / step);

this.gr.DrawLine(pen,pointf,e.Location);

pointf.X = this.pictureBox1.Size.Width;

this.gr.DrawLine(pen, pointf, e.Location);

pointf.X = (i \* (this.pictureBox1.Size.Width)) / step;

pointf.Y = 0;

this.gr.DrawLine(pen, pointf, e.Location);

pointf.Y = this.pictureBox1.Size.Height;

this.gr.DrawLine(pen, pointf, e.Location);

}

}

//при движении и нажатии мыши выполняется делегат

private void pictureBox1\_MouseMove(object sender, MouseEventArgs e)

{

// Point p = e.Location;

this.label1.Text = ("X = " + e.X + " " + "Y = " + e.Y);

if (e.Button != MouseButtons.Left) return;

//Graphics gr = this.pictureBox1.CreateGraphics();

// PointF pf1 = new PointF(e.X, e.Y);

// Pen pen = new Pen(Color.DarkBlue, 3.5f);

// gr.DrawLine(pen, pf1, new PointF(e.X + 1, e.Y + 1));

// Pen brush = new Pen(new SolidBrush(Color.IndianRed), 9.5f);

// gr.DrawLine(brush, pf1, new PointF(e.X + 1, e.Y + 1));

this.operation(e);

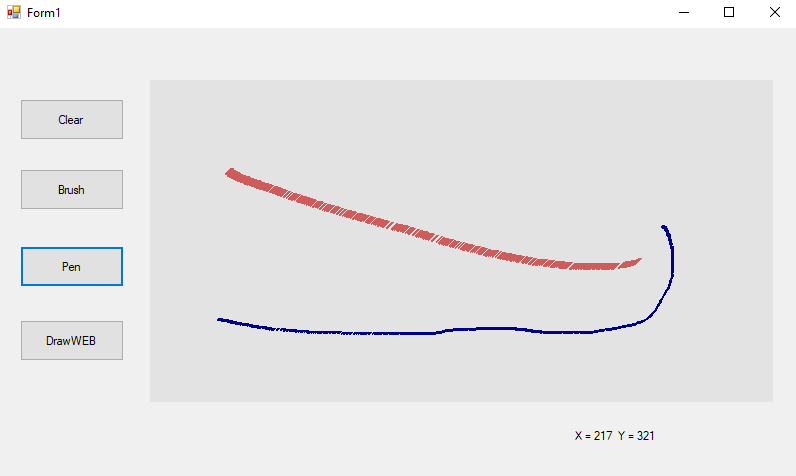
this.pictureBox1.Image = this.bitmap;//задаем bitmap картинке

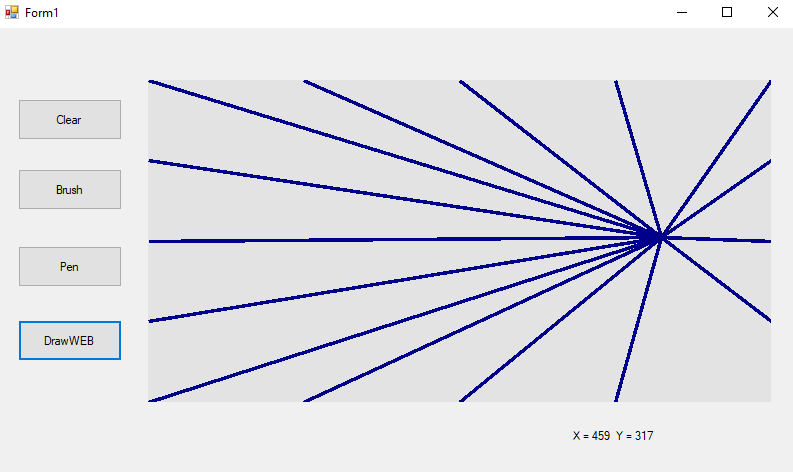
}

}

}







№4. Компонента MessageBox

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ISRPPS\_Lab\_4

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

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 ISRPPS\_Lab\_4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)//функция которая возвращает результат нажатия на кнопку диалога

{

MessageBox.Show("hi");

MessageBox.Show("hi","testBox2");

MessageBox.Show("hi", "testBox3",MessageBoxButtons.OKCancel);

MessageBox.Show("Information", "testBox4", MessageBoxButtons.YesNoCancel,MessageBoxIcon.Information);

MessageBox.Show("Information", "testBox5", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information,MessageBoxDefaultButton.Button3);

}

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

{

DialogResult result = MessageBox.Show("Are you sure?","Exit",

MessageBoxButtons.YesNoCancel,

MessageBoxIcon.Question,

MessageBoxDefaultButton.Button1);

switch (result)//обработка результата

{

case DialogResult.Yes:

MessageBox.Show("Have a nice day!");

Application.Exit();

break;

case DialogResult.No:

MessageBox.Show("No");

break;

case DialogResult.Cancel:

MessageBox.Show("Cancel ");

break;

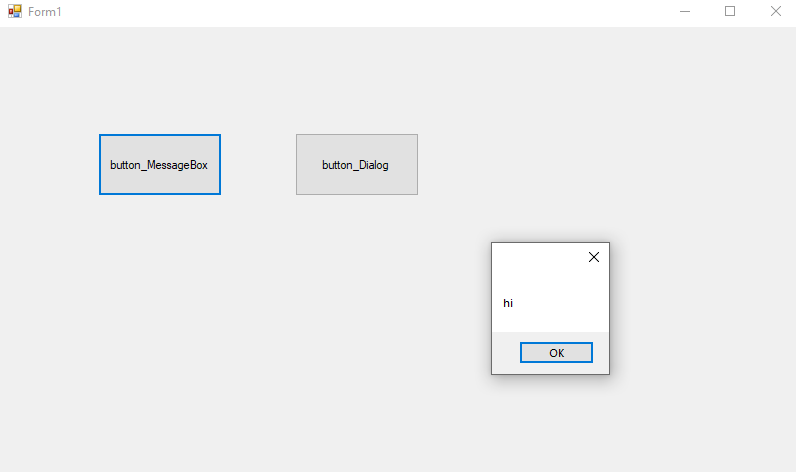
default: break;

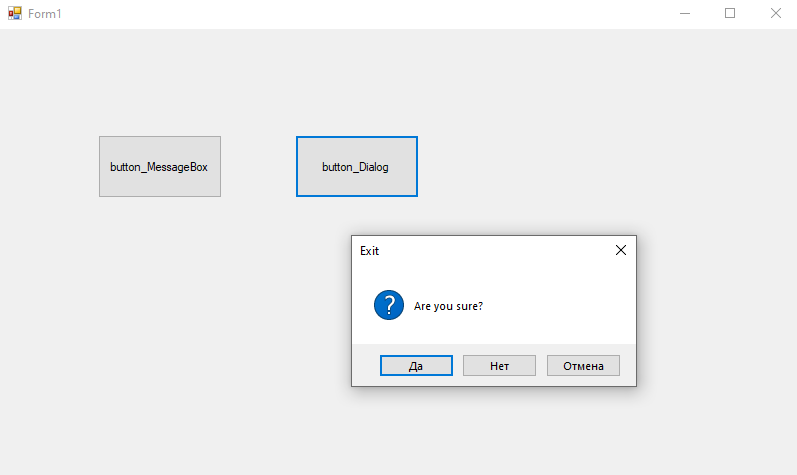
}

}

}

}





№5. Паттерны Abstract Factory и Singleton

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ISRPPS\_Lab\_5

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(Form1.fa) ;//форма через атрибут доступа

}

}

}

**Form1.cs**

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.Reflection;

namespace ISRPPS\_Lab\_5

{

public partial class Form1 : Form

{

private static Form1 f1 = null;//атрибут (2)

private Test test = null;

AbsFac bank = null;

Client client = null;

private Form1()//(1)конструктор формы private

{

InitializeComponent();

bank = new Sber();

test = Test.Instance;

}

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

{

}

public static Form1 fa

{

get { if (f1 == null) f1 = new Form1(); return f1; }

}

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

{

this.client = new Client(new Sber());

this.client.Run();

this.label1.Text = this.client.abstractProductA.account.ToString();

}

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

{

this.client = new Client(new Alpha());

this.client.Run();

this.label2.Text = this.client.abstractProductB.account.ToString();

}

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

{

this.label1.Text = "account\_sber";

}

private void label2\_Click(object sender, EventArgs e)

{

this.label2.Text = "account\_alpha";

}

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

{

MessageBox.Show(fa.test.TestF());

}

}

public class Singleton<T> where T : class //порождающий класс, нужен для работы с одним объектом, отношение клиент-утилита

{

protected Singleton() { MessageBox.Show("Singleton"); }

private sealed class SingletonCreator<S> where S : class

{

private static readonly S instance =

(S)typeof(S).GetConstructor(BindingFlags.Instance | BindingFlags.NonPublic,

null, new Type[0], new ParameterModifier[0]).Invoke(null);

public static S CreatorInstance { get { return instance; } }

}

public static T Instance

{

get

{

return SingletonCreator<T>.CreatorInstance;

}

}

}

public class Test : Singleton<Test>

{

private Test() { MessageBox.Show("Test"); }

public string TestF() { return "singleton "; }

}

/////////////////////////////////////////////////////////////////////

abstract class AbsFac//по сути расширенная спецификация

{

public abstract AbsProdA CreatProdA();

public abstract AbsProdB CreatProdB();

}

abstract class AbsProdA { public int account { set; get; } = 100559; }

abstract class AbsProdB//абстрактные продукты

{

public int account { set; get; } = 200420;

public abstract void Inter(AbsProdA a);

}

class Sber : AbsFac//абстрактное производство

{

public override AbsProdA CreatProdA() { return new CardA(); }

public override AbsProdB CreatProdB() { return new CardB(); }

}

class CardA : AbsProdA { }

class CardB : AbsProdB

{

public override void Inter(AbsProdA a) { MessageBox.Show(this.GetType().Name + " interact with " + a.GetType().Name); }

}

class Alpha : AbsFac

{

public override AbsProdA CreatProdA() { return new CardAlpha1(); }

public override AbsProdB CreatProdB() { return new CardAlpha2(); }

}

class CardAlpha1 : AbsProdA { }

class CardAlpha2 : AbsProdB

{

public override void Inter(AbsProdA a) { MessageBox.Show(this.GetType().Name + " interact with " + a.GetType().Name); }

}

class Client

{

public AbsProdA abstractProductA;

public AbsProdB abstractProductB;

public Client(AbsFac factory)

{

abstractProductB = factory.CreatProdB();

abstractProductA = factory.CreatProdA();

}

public void Run()

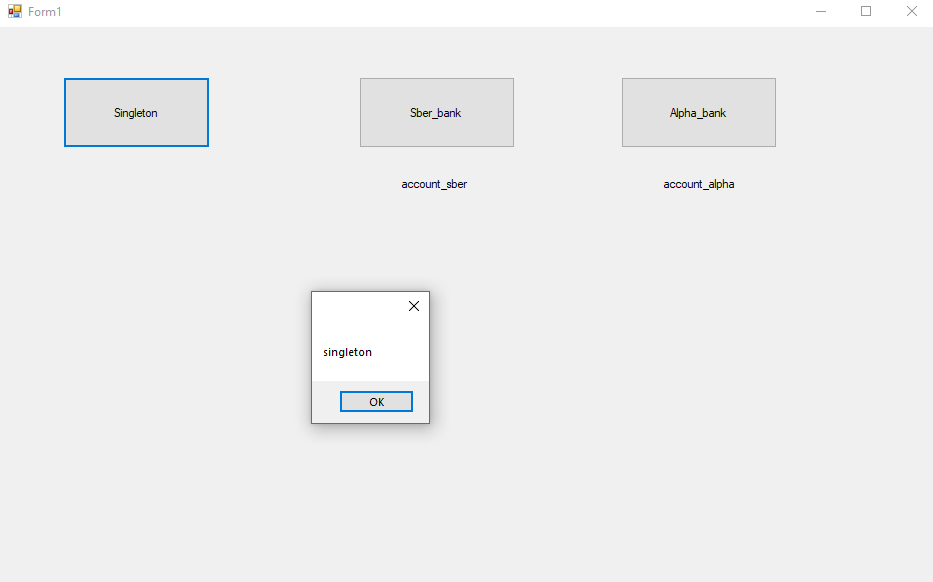
{

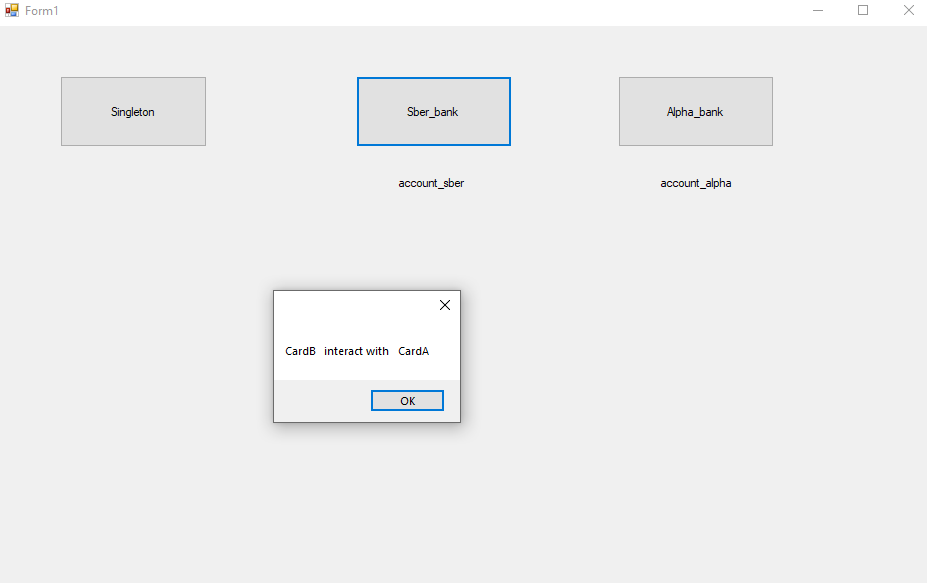
abstractProductB.Inter(abstractProductA);

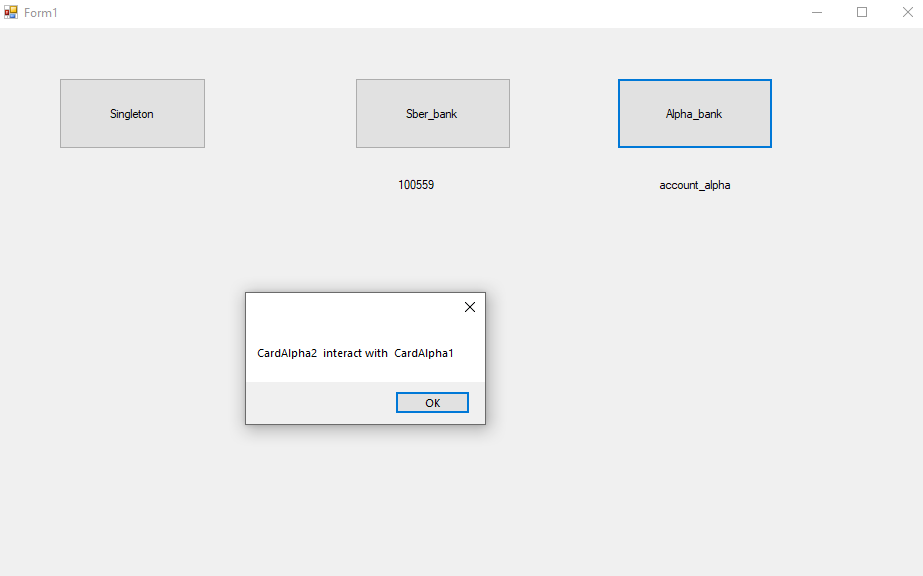
}

}

}







№6. Buttons, CheckBox, Labels, Scrolls, Track, масштабирование Шрифта

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ISRPPS\_Lab\_6

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

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 ISRPPS\_Lab\_6

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

trackBar1.Minimum = 0;

trackBar1.Maximum = int.MaxValue;

}

private void checkedListBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

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

{

button1.BackColor = Color.FromArgb(trackBar1.Value);

this.label1.Text = button1.BackColor.Name;

this.richTextBox1.SelectionColor = this.button1.BackColor;

}

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

{

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (this.checkBox1.Checked)

{

this.button1.Visible = false;

this.trackBar1.Visible = false;

this.label1.Visible = false;

}

else

{

this.button1.Visible = true;

this.trackBar1.Visible = true;

this.label1.Visible = true;

}

}

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

{

MessageBox.Show("This button show selected color.\n" +

""+ "By the means slider select the color.","Information",

MessageBoxButtons.OK,MessageBoxIcon.Information);

}

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

{

//this.trackBar2.Value = 9;

this.richTextBox1.SelectionColor = this.button1.BackColor;

this.richTextBox1.Font = new Font(this.richTextBox1.Font.FontFamily, trackBar2.Value);

this.label2.Text ="Font : "+Convert.ToString( trackBar2.Value);

}

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

{

this.richTextBox1.Text = "C# syntax is highly expressive, " +

"yet it's also simple and easy to learn. The curly brace syntax" +

" of C# will be instantly recognizable to anyone familiar with C," +

" C++, Java or JavaScript. Developers who know any of these languages" +

" are typically able to work productively in C# within a short time. " +

"C# provides powerful features such as nullable types, delegates," +

" lambda expressions, pattern matching, and safe direct memory access." +

" C# supports generic methods and types, which provide increased type safety " +

"and performance. C# provides iterators, which enable implementers of collection" +

" classes to define custom behaviors for client code. Language-Integrated Query (LINQ)" +

" expressions make the strongly typed query a first-class language construct.";

}

private void checkBox2\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox2.Checked)

{

this.richTextBox1.RightToLeft =RightToLeft.Yes;

}

else this.richTextBox1.RightToLeft = RightToLeft.No;

}

private void richTextBox1\_TextChanged(object sender, EventArgs e)

{

this.richTextBox1.SelectionColor = this.button1.BackColor;

}

private void checkBox3\_CheckedChanged(object sender, EventArgs e)

{

if (this.checkBox3.Checked)

{

this.button2.Visible = false;

this.trackBar2.Visible = false;

this.label2.Visible = false;

this.richTextBox1.Visible = false;

this.checkBox2.Visible = false;

this.button3.Visible = false;

}

else

{

this.button2.Visible = true;

this.trackBar2.Visible = true;

this.label2.Visible = true;

this.richTextBox1.Visible = true;

this.checkBox2.Visible = true;

this.button3.Visible = true; ;

}

}

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

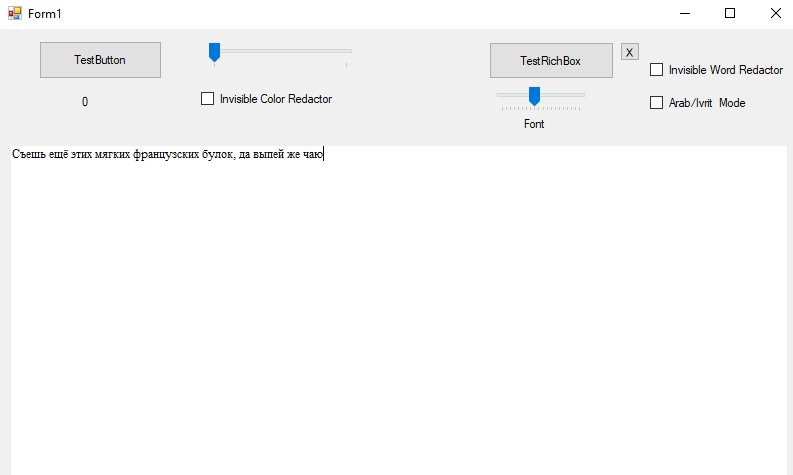
{

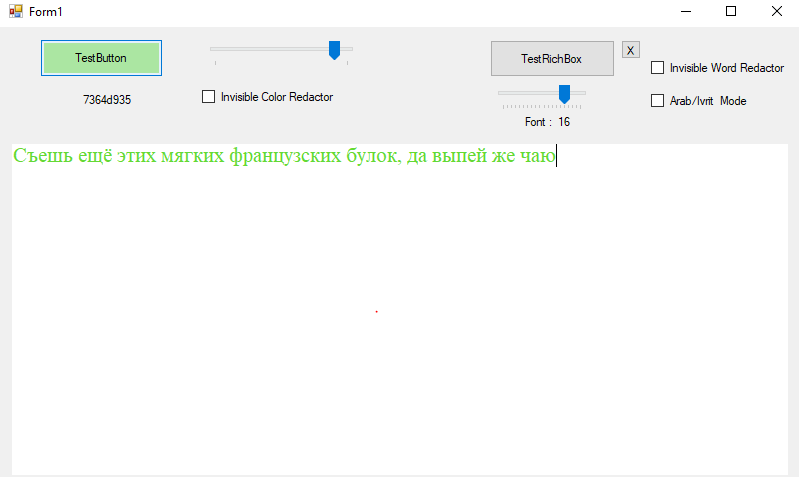
this.richTextBox1.Text = default;

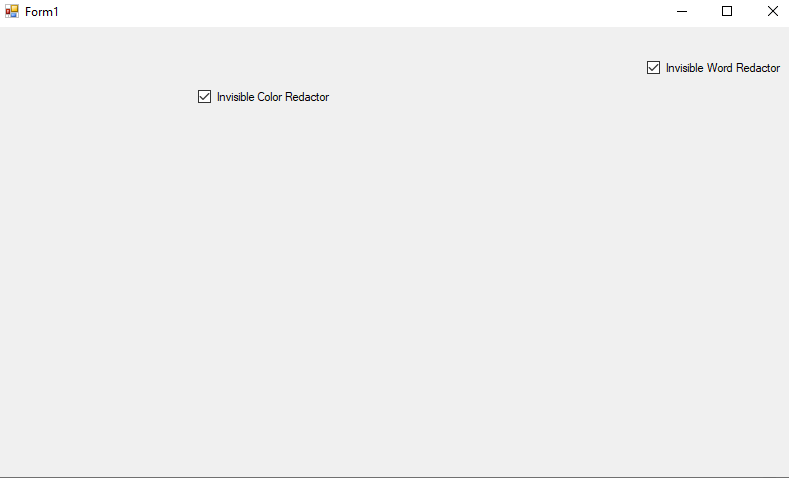
}

}

}







№7. Паттерн Chain-responsibility

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ISRPPS\_Lab\_7

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

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;

//поведенческий паттерн

namespace ISRPPS\_Lab\_7

{

public partial class Form1 : Form

{

Handler h1 =null;

int[] sequencce = new int[3];

public Form1()

{

InitializeComponent();

this.h1 = new H1();

Handler h2 = new H2();

this.h1.next = h2;

h2.next = new H3();

int[] sequencce = { 2, 1, 1, 1, 2, 3 }; //запросы

foreach (int act in sequencce) { h1.HandlerRequest(act);}

}

abstract class Handler

{

public Handler next { set; get; } = null;// ссылка на последующий объект

public abstract void HandlerRequest(int act);

}

class H1 : Handler

{

public override void HandlerRequest(int act)

{

if (act == 1) { MessageBox.Show("H1 handled request 1 ", this.GetType().Name); }

else if (next != null)

{

next.HandlerRequest(act);

}

}

}

class H2 : Handler

{

public override void HandlerRequest(int act)

{

if (act == 2) { MessageBox.Show("H2 handled request 2 ", this.GetType().Name); }

else if (next != null)

{

next.HandlerRequest(act);

}

}

}

class H3 : Handler

{

public override void HandlerRequest(int act)

{

if (act == 3) { MessageBox.Show("H3 handled request 3 ", this.GetType().Name); }

else if (next != null)

{

next.HandlerRequest(act);

}

}

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked) { int i = 0; sequencce[i]=1;}

else { int i = 0; sequencce[i] = 0; }

}

private void checkBox2\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox2.Checked) { int i = 1; sequencce[i] = 2; }

else { int i = 1; sequencce[i] = 0; }

}

private void checkBox3\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox3.Checked) { int i = 2; sequencce[i] = 3; }

else { int i = 2; sequencce[i] = 0; }

}

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

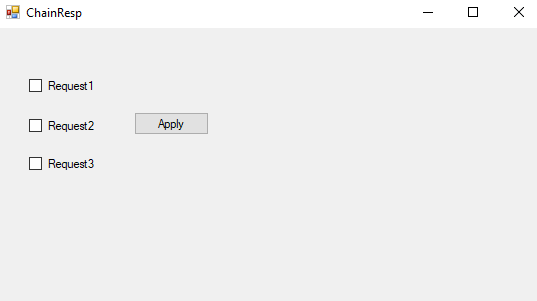
{

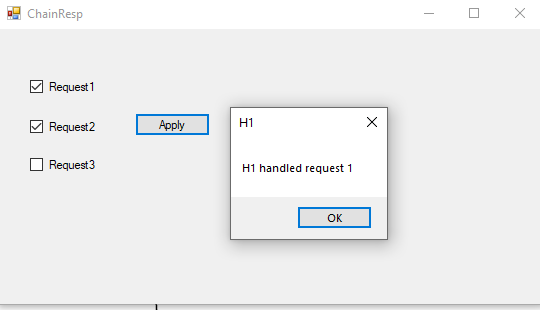
foreach (int act in sequencce) { h1.HandlerRequest(act); }

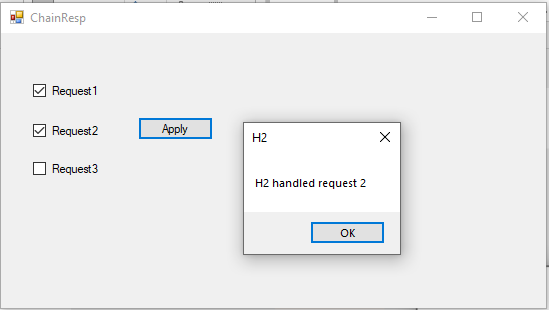
}

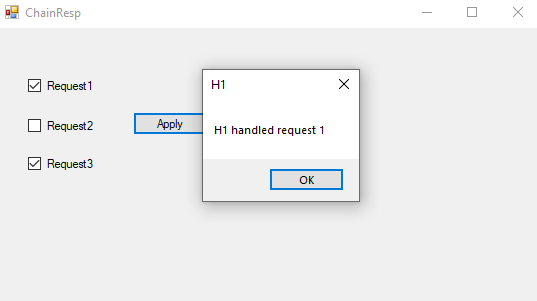
}

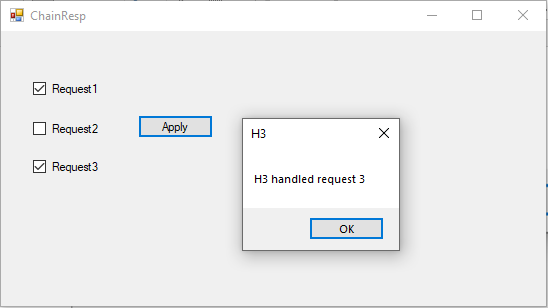
}











№8. Паттерн Command

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ISRPPS\_Lab\_8

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Command0.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

//поведенческий паттерн,нужен, когда класс получатель и класс калькулятор напрямую не связаны

//Паттерн Command предоставляет возможность ставить запросы в очередь, протоколировать запросы, а также поддерживать отмену операций

namespace ISRPPS\_Lab\_8

{

interface Command//в виде интерфейса

{

void Execute();

void UnExecute();

}

class ConcreteCommand : Command

{

char @operator;

int operand;

Calculator calculator;

// Constructor

public ConcreteCommand(Calculator calculator, char @operator, int operand)

{

this.calculator = calculator;

this.@operator = @operator;

this.operand = operand;

}

public char Operator { set { @operator = value; } }

public int Operand { set { operand = value; } }

public void Execute()

{

calculator.Operation(@operator, operand);

}

public void UnExecute()

{

calculator.Operation(Undo(@operator), operand);

}

// Private helper function : приватные вспомогательные функции

private char Undo(char @operator)

{

char undo;

switch (@operator)

{

case '+':

undo = '-';

break;

case '-':

undo = '+';

break;

case '\*':

undo = '/';

break;

case '/':

undo = '\*';

break;

default:

undo = ' ';

break;

}

return undo;

}

}

public class Calculator

{

static public int curr { set; get; } = 0; // левый operand

public void Operation(char @operator, int operand)

{

switch (@operator)

{

case '+':

curr += operand;

break;

case '-':

curr -= operand;

break;

case '\*':

curr \*= operand;

break;

case '/':

if (operand != 0)

curr /= operand;

else MessageBox.Show("You want to save your PC? Don't divide a number by zero !!!", "Warning");

break;

}

// MessageBox.Show("" + curr + " " + @operator + " " + operand);

}

}

class User

{

// Initializers

private Calculator \_calculator = new Calculator();

private List<Command> \_commands = new List<Command>();

private int \_current = 0;

public int getResult() {{ return Calculator.curr; } }

public void Redo(int levels)

{

// Делаем возврат операций

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

if (\_current < \_commands.Count)

\_commands[\_current++].Execute();

}

public void Undo(int levels)

{

// Делаем отмену операций

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

if (\_current > 0) \_commands[--\_current].UnExecute();

}

Command command = null;

public void Compute(char @operator, int operand)

{

// Создаем команду операции и выполняем её

command = new ConcreteCommand(

\_calculator, @operator, operand);

command.Execute();

// Добавляем операцию к списку отмены

\_commands.Add(command);

\_current++;

}

}

}

**Form1.cs**

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 ISRPPS\_Lab\_8

{

public partial class Form1 : Form

{

User user = new User();

char @operator;

public Form1()

{

InitializeComponent();

this.textBox1.Text ="0";

}

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

{

}

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

{

@operator = '-';//то что присваивается в оператор

//user.Compute(@operator, Convert.ToInt32(this.textBox1.Text));

this.textBox1.Clear();

}

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

{

@operator = '+';

//user.Compute('+', Convert.ToInt32(this.textBox1.Text));

this.textBox1.Clear();

}

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

{

//this.textBox1.Clear();

user.Compute(@operator, Convert.ToInt32(this.textBox1.Text));//тут command

this.textBox1.Text = user.getResult().ToString();

}

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

{

@operator = '\*';

// user.Compute('\*', Convert.ToInt32(this.textBox1.Text));

this.textBox1.Clear();

}

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

{

@operator = '/';

//user.Compute('/', Convert.ToInt32(this.textBox1.Text));

this.textBox1.Clear();

}

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

{

}

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

{

user.Undo(1);

this.textBox1.Text = user.getResult().ToString();

}

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

{

user.Redo(1);

this.textBox1.Text =user.getResult().ToString();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button8\_Click\_1(object sender, EventArgs e)// C

{

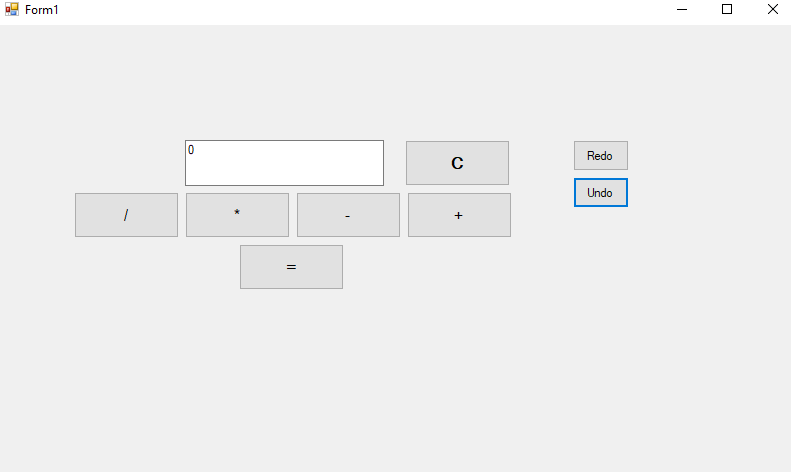
Calculator.curr = 0;

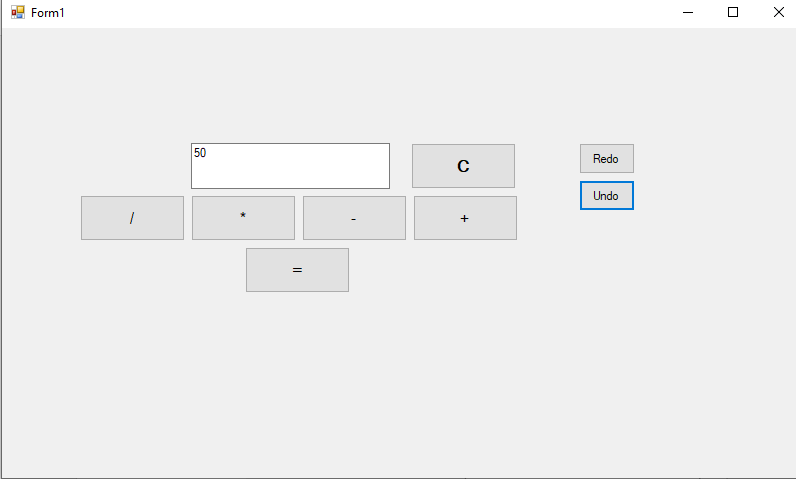
this.textBox1.Text = "0";

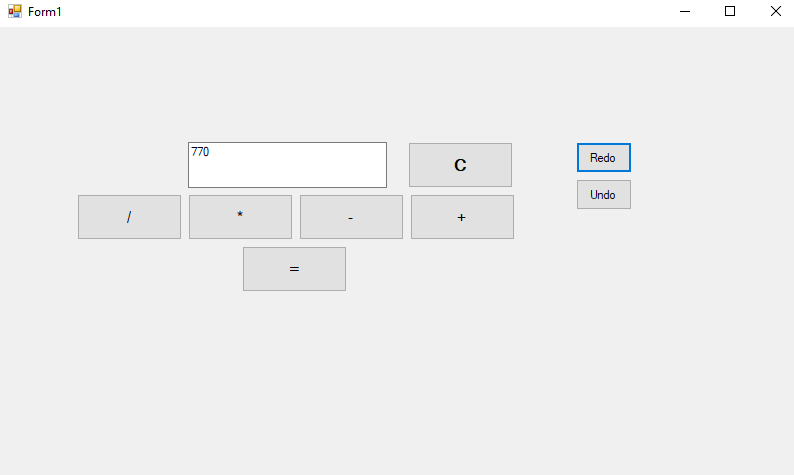
}

}

}







№9. Menus, ToolStrip and StatusStrip

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ISRPPS\_Lab\_9

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

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 ISRPPS\_Lab\_9

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

this.toolStripLabel1.Text = "X: " + e.X.ToString() + " Y: " + e.Y.ToString();

this.toolStripStatusLabel1.Text = "Status: ";

}

private void toolStripComboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

int s = toolStripComboBox1.SelectedIndex;

switch (s)

{

case 0: { if (s == 0) MessageBox.Show("Choosed: Name\_Line\_1"); } break;

case 1: { if (s == 1) MessageBox.Show("Choosed: Name\_Line\_2"); } break;

case 2: { if (s == 2) MessageBox.Show("Choosed: Name\_Line\_3"); } break;

case 3: { if (s == 3) MessageBox.Show("Choosed: Name\_Line\_4"); } break;

case 4: { if (s == 4) MessageBox.Show("Choosed: Name\_Line\_5"); } break;

}

}

private void toolStripButton2\_Click(object sender, EventArgs e)

{

MessageBox.Show("Operation Undo");

}

private void toolStripButton3\_Click(object sender, EventArgs e)

{

MessageBox.Show("Operation Redo");

}

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

{

MessageBox.Show("Created");

}

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

{

MessageBox.Show("Opened");

}

private void exitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

MessageBox.Show("Exit");

Application.Exit();

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

MessageBox.Show("Program launched");

this.toolStripProgressBar1.Value = 0;

do this.toolStripProgressBar1.Value += 50;

while (this.toolStripProgressBar1.Value == 100);

MessageBox.Show("Error.Try again!");

this.toolStripStatusLabel1.Text += "Error";

}

private void toolStripLabel1\_Click(object sender, EventArgs e)

{

MessageBox.Show("toolStripLabel1\_Click");

this.toolStripComboBox1.Text = "Label1 Selected";

}

private void toolStripSplitButton1\_ButtonClick(object sender, EventArgs e)

{

MessageBox.Show("Launch");

//this.toolStripProgressBar1.Value = 0;

while (this.toolStripProgressBar1.Value < 100)

{ this.toolStripProgressBar1.Value += 1; }

if (this.toolStripProgressBar1.Value == 100)

{

MessageBox.Show("Complete");

this.toolStripStatusLabel1.Text += "Complete";

}

this.toolStripProgressBar1.Value = 0;

}

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

{

}

private void toolStripProgressBar1\_Click(object sender, EventArgs e)

{

while (this.toolStripProgressBar1.Value < 100)

{ this.toolStripProgressBar1.Value += 50; }

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if((Control.ModifierKeys==Keys.Control) && (e.KeyCode==Keys.Space))

{

Application.Exit();

}

}

private void toolStripButton4\_Click(object sender, EventArgs e)

{

}

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

{

}

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

{

if (e.Button != MouseButtons.Right) return;

if (e.X > 200) { this.contextMenuStrip2.Show(PointToScreen(e.Location)); }

else this.contextMenuStrip1.Show(PointToScreen(e.Location));

}

private void toolStripMenuItem8\_Click(object sender, EventArgs e)

{

}

private void toolStripMenuItem10\_Click(object sender, EventArgs e)

{

}

private void contextMenuStrip1\_Opening(object sender, CancelEventArgs e)

{

//MessageBox.Show("contextMenuStrip1");

}

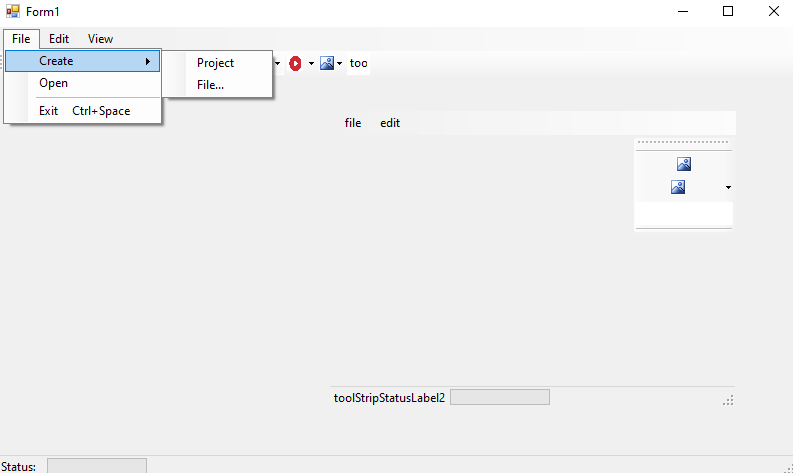
private void toolStripContainer1\_RightToolStripPanel\_Click(object sender, EventArgs e)

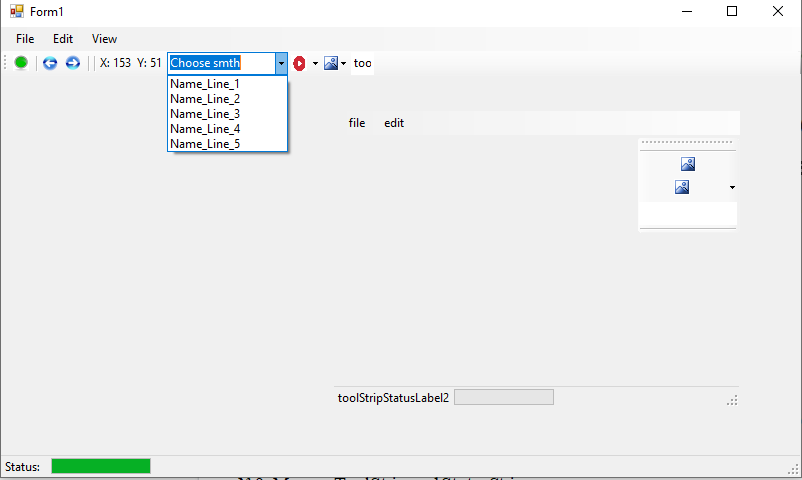
{

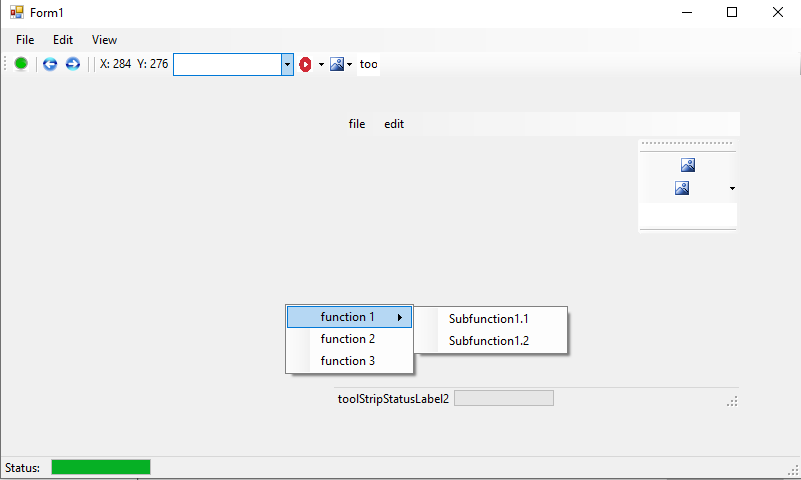
}

}

}







№10. Dialog Boxes.Clipboard.

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ISRPPS\_Lab\_10

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

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 ISRPPS\_Lab\_10

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

this.richTextBox1.Text = "Example";

}

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

{

}

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

{

DialogResult dr = this.colorDialog1.ShowDialog();

switch (dr)

{

case DialogResult.OK:

this.label1.BackColor = this.colorDialog1.Color;

break;

}

}

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

{

DialogResult dr = this.fontDialog1.ShowDialog();

switch (dr)

{

case DialogResult.OK:

this.label1.Font = this.fontDialog1.Font;

break;

}

}

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

{

this.label1.BackColor = DefaultBackColor;

this.label1.Font = DefaultFont;

}

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

{

DialogResult dr = this.saveFileDialog1.ShowDialog();

switch (dr)

{

case DialogResult.OK:

MessageBox.Show(this.saveFileDialog1.FileName);

break;

default:

break;

}

try

{

using (Stream fileopen = this.saveFileDialog1.OpenFile())

{

this.richTextBox1.SaveFile(fileopen, RichTextBoxStreamType.RichText);

}

}

catch(IOException exc)

{

MessageBox.Show(exc.Message, "Error");

return;

}

}

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

{

DialogResult dr = this.openFileDialog1.ShowDialog();

switch (dr)

{

case DialogResult.OK:

MessageBox.Show( this.openFileDialog1.FileName);

break;

}

using (Stream fileopen = this.openFileDialog1.OpenFile())

{

this.richTextBox1.LoadFile(fileopen, RichTextBoxStreamType.RichText);

}

}

private void richTextBox1\_TextChanged(object sender, EventArgs e)

{

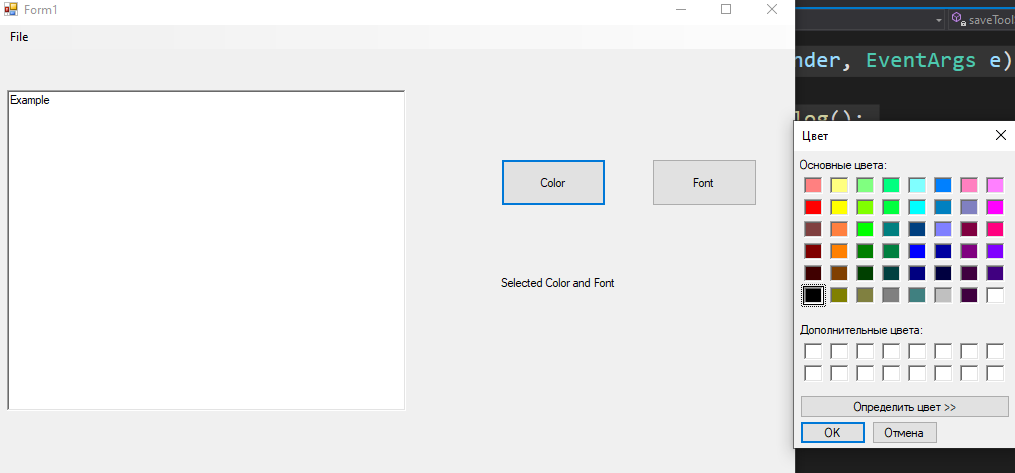
this.richTextBox1.SelectionColor = this.label1.BackColor;

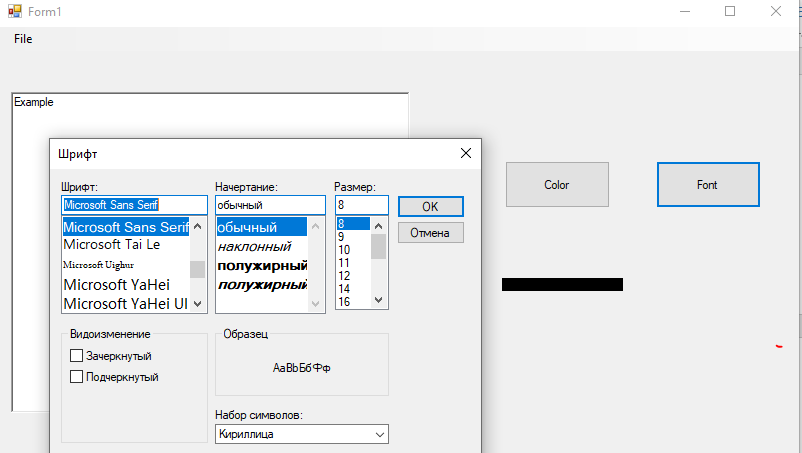
this.richTextBox1.Font = this.label1.Font;

}

}

}





№11. Tree View, List View

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab\_i\_11\_2

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab\_i\_11\_2

{

public partial class Form1 : Form

{

string fullPath; // Полный путь

public Form1()

{

InitializeComponent();

TreeNode Node3 = new TreeNode("Node3");

Node3.Nodes.Add(new TreeNode("node3.1"));

treeView1.Nodes.Add(Node3);

treeView1.Nodes[2].Nodes.Add(new TreeNode("node3.2"));

treeView1.Nodes[2].Nodes[1].Nodes.Add(new TreeNode("node3.2.1"));

// УДАЛЕНИЕ УЗЛОВ

treeView1.ImageList = imageList1;

treeView1.Nodes[0].ImageIndex = 1;

DeviceTreeIntit();

}

TreeNode device = null;

public void DeviceTreeIntit()

{

string[] Array = Directory.GetLogicalDrives(); // получ. список лог. дисков системы

this.treeView1.BeginUpdate(); // блок перерисовки окна до снятия блокировки

this.treeView1.Nodes.Clear(); // удаление всех узлов дерева

foreach (string s in Array)

{

device = new TreeNode(s, 0, 0); // Для всех устройств создается узел дерева

this.treeView1.Nodes.Add(device);// add узла в коллекцию узлов дерева

GetDirs(device); //Доб. в дерево списка содержимого корневого каталога //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

}

treeView1.EndUpdate();

}

// Получение списка всех подкаталогов заданого каталога node

public void GetDirs(TreeNode node)

{

DirectoryInfo[] diArray;

node.Nodes.Clear();

string fullPath = node.FullPath;

DirectoryInfo di = new DirectoryInfo(fullPath);

try

{

// Запись инф. о всех подкаталогах данного каталога массив diArray

diArray = di.GetDirectories();

}

catch { return; }

//diArray исп. для заполнение узла дерева содержимым каталога

foreach (DirectoryInfo dirinfo in diArray)

{

TreeNode dir = new TreeNode(dirinfo.Name, 0, 0);

node.Nodes.Add(dir);

}

}

private void button1\_Click(object sender, EventArgs e) { treeView1.Nodes[0].Expand(); }

private void button2\_Click(object sender, EventArgs e) { treeView1.Nodes[0].ExpandAll();}

private void button3\_Click(object sender, EventArgs e) { treeView1.Nodes[0].Collapse(); }// Метод класса TreeNode

private void button4\_Click(object sender, EventArgs e) { treeView1.Nodes[0].Toggle(); }

private void treeView1\_AfterSelect(object sender, TreeViewEventArgs e)

{

listView1.Items.Clear();

label4.Text = "";

label1.Text = e.Node.Text; // e.Node - ссылка на выбранный узел

label6.Text = e.Node.FullPath; // полный путь к узлу

DirectoryInfo di = new DirectoryInfo(e.Node.FullPath);

FileInfo[] fiArray;

DirectoryInfo[] diArray;

// получение списка всех каталогов и файлов из выбранного каталога

try

{

fiArray = di.GetFiles();

diArray = di.GetDirectories();

} catch { return; }

//Наполнение списка файлами с информацией о них

foreach (FileInfo fileInfo in fiArray)

{

ListViewItem lvi = new ListViewItem(fileInfo.Name);

lvi.SubItems.Add(fileInfo.Length.ToString());

// время последнего изменения

lvi.SubItems.Add(fileInfo.LastAccessTime.ToString());

listView1.Items.Add(lvi);

}

}

private void treeView1\_BeforeExpand(object sender, TreeViewCancelEventArgs e)

{

treeView1.BeginUpdate();

foreach(TreeNode node in e.Node.Nodes) { GetDirs(node); }

treeView1.EndUpdate();

}

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

{

if (this.textBox1.TextLength != 0)

treeView1.SelectedNode.Nodes.Add(new TreeNode(this.textBox1.Text));

}

private void listView1\_ItemChecked(object sender, ItemCheckedEventArgs e)

{

Int64 size = 0;

foreach(ListViewItem lvi in listView1.Items)

{

if (lvi.Checked){

size += Convert.ToInt64(lvi.SubItems[1].Text);

}

}

label4.Text = size.ToString();

}

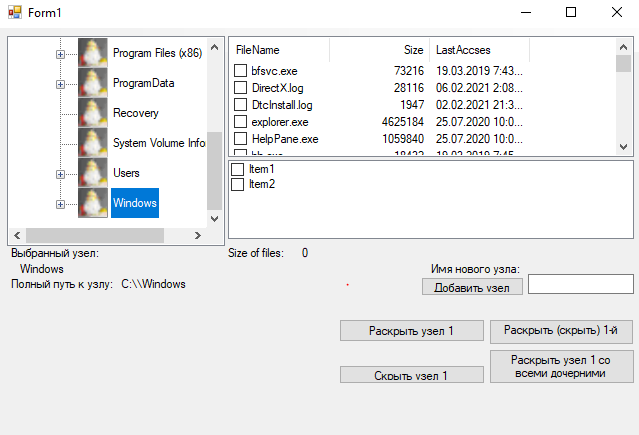
private void label7\_Click(object sender, EventArgs e)

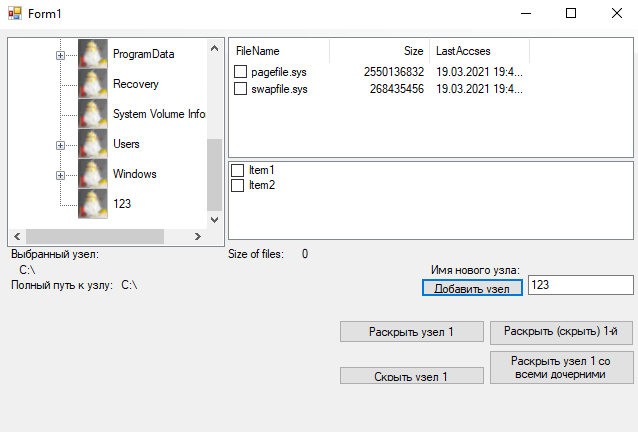
{

}

}

}





№12. Drag and Drop, Clip

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab\_i\_12

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

using System;

using System.Drawing;

using System.Windows.Forms;

namespace lab\_i\_12

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_MouseMove(object sender, MouseEventArgs e)

{

this.button1.MouseDown += new MouseEventHandler(this.button1\_MouseMove);//запуск события MouseDown-тащим

}

private void button1\_MouseDown(object sender, MouseEventArgs e)

{

// при нажатии на кнопку вызывается перетаскивание указанного объекта

this.button1.DoDragDrop(button1, DragDropEffects.Move); // метод, начинает операцию перетаскивания.

}

private void Panels\_DragEntre(object sender, DragEventArgs e)

{

e.Effect = DragDropEffects.Move;

}

private void Pane1\_DragDrop(object sender, DragEventArgs e)

{

this.button1.Parent = (Panel)sender;

this.button1.Location = splitContainer1.Panel1.PointToClient(new Point (e.X, e.Y));

}

private void Pane2\_DragDrop(object sender, DragEventArgs e)

{

this.button1.Parent = (Panel)sender;

this.button1.Location = splitContainer1.Panel2.PointToClient(new Point(e.X, e.Y));

}

private void splitContainer1\_DragEnter(object sender, DragEventArgs e)

{

e.Effect = DragDropEffects.Move;

}

private void splitContainer1\_Panel1\_DragOver(object sender, DragEventArgs e)

{

//MessageBox.Show("DragOver");

}

private void checkBox1\_MouseDown(object sender, MouseEventArgs e)

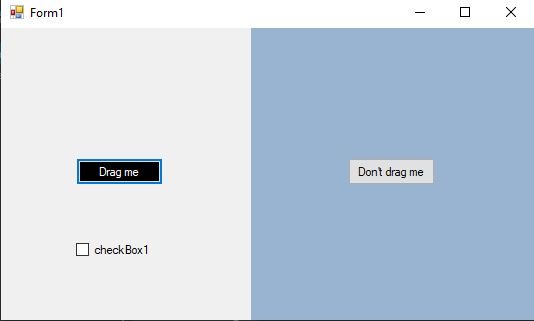
{

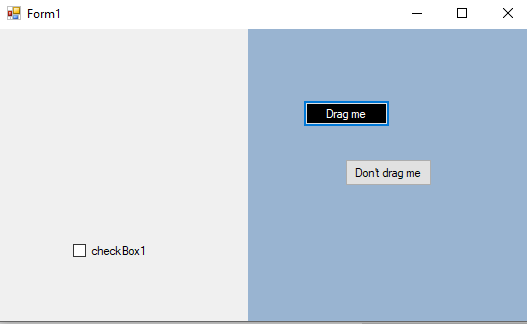
this.checkBox1.DoDragDrop(checkBox1, DragDropEffects.Move);

}

}

}





№13. Lines, Curves and Area Fills.

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab\_i\_13

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Lab\_i\_13

{

public partial class Form1 : Form

{

Graphics grfx;

int cx;

int cy;

public Form1()

{

InitializeComponent();

//this.Handle;

}

Pen penRed5 = new Pen(Color.Red, 5);

Pen penPink2 = new Pen(Color.Pink, 2);

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

{

// берем дискриптор окна

grfx = Graphics.FromHwnd(this.pictureBox1.Handle);

cx = pictureBox1.Size.Width;

cy = pictureBox1.Size.Height - 60;

grfx.DrawLine(penRed5, 0,0,cx,cy);

grfx.DrawLine(penRed5, cx,0,0,cy);

grfx.DrawLine(penRed5, 0, cy, cx, pictureBox1.Size.Height);

grfx.DrawLine(penRed5, cx, cy, 0, pictureBox1.Size.Height);

grfx.DrawLine(penPink2, 0, 0, cx, cy);

grfx.DrawLine(penPink2, cx, 0, 0, cy);

grfx.DrawLine(penPink2, 0, cy, cx, pictureBox1.Size.Height);

grfx.DrawLine(penPink2, cx, cy, 0, pictureBox1.Size.Height);

}

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

{

grfx = Graphics.FromHwnd(this.pictureBox2.Handle);

cx = pictureBox2.Size.Width;

cy = pictureBox2.Size.Height;

PointF[] aptf = new PointF[cx];

//grfx.Clear(Color.SkyBlue);

for (int i = 0; i < cx; i++) {

aptf[i].X = i;

aptf[i].Y = cy/2\*(1-(float)Math.Sin(i\*2\*Math.PI/(cx-1))); // синусоида разбита на 2 части, i - шаг

}

grfx.DrawLines(new Pen(Color.Orange,5), aptf);

grfx.DrawLines(new Pen(Color.LightYellow,2), aptf);

}

SolidBrush solidBrushY = new SolidBrush(Color.Yellow);

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

{

grfx = Graphics.FromHwnd(this.pictureBox3.Handle);

cx = pictureBox3.Size.Width;

cy = pictureBox3.Size.Height;

//grfx.Clear(Color.Silver);

grfx.FillEllipse(solidBrushY, 0, 0, cx, cy);

grfx.FillEllipse(new SolidBrush(Color.White), 0+5, 0+5, cx-10, cy-10);

grfx.FillEllipse(solidBrushY, 0+10, 0+10, cx-20, cy-20);

grfx.FillEllipse(new SolidBrush(Color.Black), 0+20, 0+20, cx-40, cy-40);

}

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

{

grfx = Graphics.FromHwnd(this.pictureBox4.Handle);

cx = pictureBox4.Size.Width;

cy = pictureBox4.Size.Height;

//grfx.Clear(Color.Silver);

grfx.DrawEllipse(new Pen(Color.Green,7), 0, 0, cx, cy);

grfx.DrawEllipse(new Pen(Color.LightSeaGreen,4), 0, 0, cx, cy);

grfx.DrawEllipse(new Pen(Color.White,1), 0, 0, cx, cy);

}

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

{

grfx = Graphics.FromHwnd(this.pictureBox5.Handle);

cx = pictureBox5.Size.Width;

cy = pictureBox5.Size.Height;

//grfx.Clear(Color.Silver);

grfx.DrawBezier(new Pen(Color.SkyBlue,8),

new PointF(0, 0),

new PointF(cx+40, cy + 40),

new PointF(cx+30 , 0 - 100),

new PointF(0, cy));

grfx.DrawBezier(new Pen(Color.White, 2),

new PointF(0, 0),

new PointF(cx + 40, cy + 40),

new PointF(cx + 30, 0 - 100),

new PointF(0, cy));

//grfx.DrawBezier(new Pen(Color.White, 2),

// new PointF(cx / 2 + 2, cy / 2 + 2),

// new PointF(cx / 2, cy / 4),

// new PointF(cx / 4, cy / 2),

// new PointF(cx / 2 + 2, cy / 2 + 2));

}

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

{

grfx = Graphics.FromHwnd(this.pictureBox6.Handle);

cx = pictureBox6.Size.Width;

cy = pictureBox6.Size.Height;

//grfx.Clear(Color.Silver);

PointF[] points = new PointF[4] {new PointF(0+2,0+2),

new PointF(cx/2-20,cy-20),

new PointF(cx/2+20,cy-20),

new PointF(cx,0)};

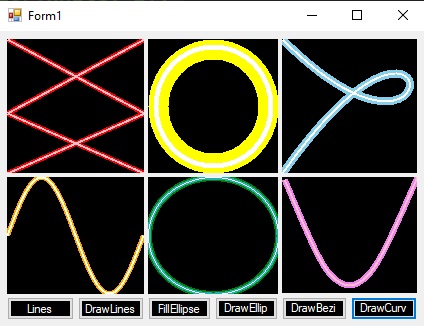
grfx.DrawCurve(new Pen(Color.Violet, 6), points);

grfx.DrawCurve(new Pen(Color.Pink, 2), points);

}

}

}



№14. Обработка прерываний от таймера. Вращение Images and Bitmap

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Lab\_i\_14.\_4

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Lab\_i\_14.\_4

{

public partial class Form1 : Form

{

public float Angle;

private System.Drawing.Drawing2D.Matrix rotatematrix = new System.Drawing.Drawing2D.Matrix();

public static readonly float ToRadin = (float)(Math.PI/180);

public Form1()

{

InitializeComponent();

this.pictureBox1.Bounds = new Rectangle(0,0, this.Width, this.Height+100);// Bounds - границы

}

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

{

this.Angle = trackBar1.Value;

pictureBox1.Refresh();

}

private void pictureBox1\_Paint(object sender, PaintEventArgs e)

{

e.Graphics.Clear(Color.White);

if (((PictureBox)sender).Image == null) return;

// Выбор высокого качества интерполяции

// Интерполяция - метод увеличения размера пикселей внутри изображения.

e.Graphics.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.HighQualityBilinear;

// Линейное пр-е сдвиг начала координат на вектр(х/2, y/2)

e.Graphics.TranslateTransform(((PictureBox)sender).Width / 2+50, ((PictureBox)sender).Height / 2-70);

e.Graphics.RotateTransform(Angle);

//Рисование изображ. в центре picBox

if (((PictureBox)sender).Image.Width < ((PictureBox)sender).Image.Width / 2F && ((PictureBox)sender).Image.Height < ((PictureBox)sender).Image.Height / 2F)

{

e.Graphics.DrawImage(((PictureBox)sender).Image, -((PictureBox)sender).Image.Width / 2F,

-((PictureBox)sender).Image.Height / 2F);

}

else

{

e.Graphics.DrawImage(((PictureBox)sender).Image, -2 \* ((PictureBox)sender).Width / 6F, -2 \* ((PictureBox)sender).Height / 9F,

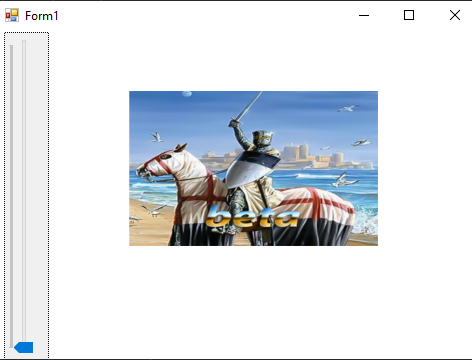
((PictureBox)sender).Width / 2F, ((PictureBox)sender).Height / 3F);

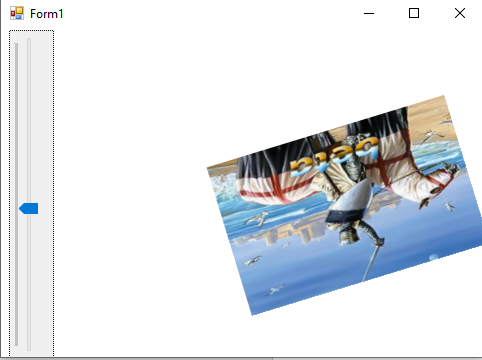
}

}

}

}





№15.Компонента dll

**DLL Project:**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ClassLibrary1\_ISRPPS\_Lab\_15

{

public static class Trigonometry

{

public static double doubleSin(double parameter) { return Math.Sin(2\*parameter); }

public static double doubleCos(double parameter) { return Math.Cos(2\*parameter); }

public static double doubletg(double parameter) { return Math.Tan(2\*parameter); }

}

}

**Main Program:**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace \_15\_1\_

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

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 ClassLibrary1\_ISRPPS\_Lab\_15;

namespace \_15\_1\_

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

}

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

{

label1.Text = Trigonometry.doubleSin(Convert.ToDouble(textBox1.Text)).ToString();

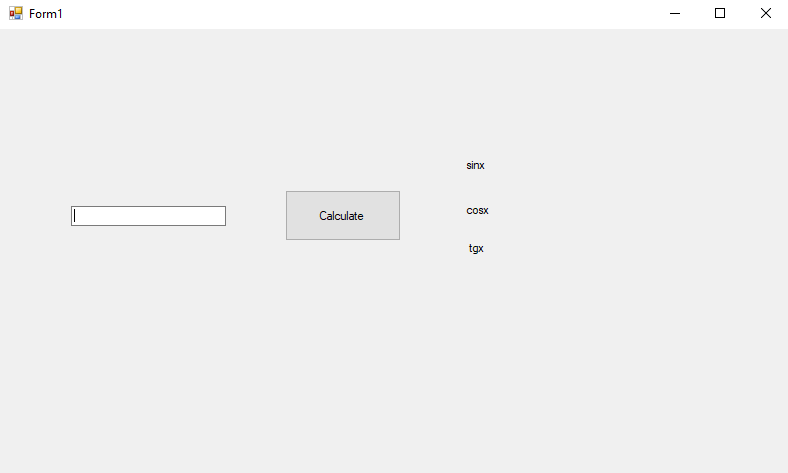
label2.Text = Trigonometry.doubleCos(Convert.ToDouble(textBox1.Text)).ToString();

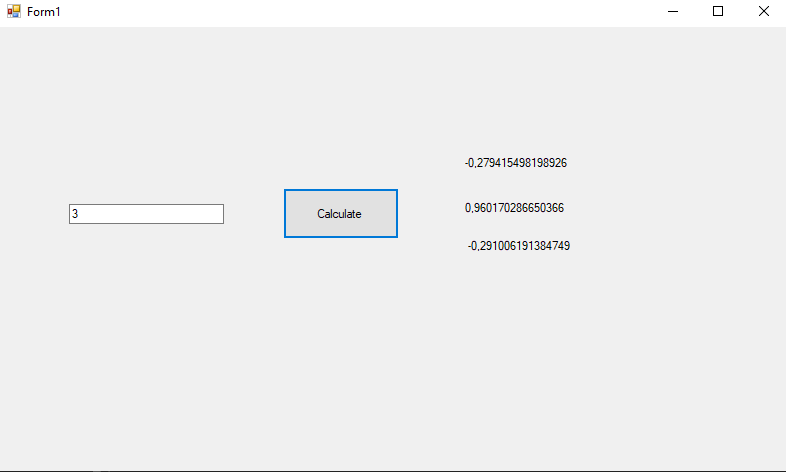
label3.Text = Trigonometry.doubletg(Convert.ToDouble(textBox1.Text)).ToString();

}

}

}





№16. Типы проектов .Net

**Проект Windows Forms Control Library – проект для создания библиотек пользовательских элементов интерфейса**

**UserControl1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Drawing;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsControlLibrary1

{

public partial class UserControl1: UserControl

{

public UserControl1()

{

InitializeComponent();

}

private void UserControl1\_Load(object sender, EventArgs e)

{

}

}

}

**UserControl2.cs**

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 WindowsFormsControlLibrary1

{

public partial class UserControl2 : UserControl

{

public UserControl2()

{

InitializeComponent();

}

private void UserControl2\_Load(object sender, EventArgs e)

{

}

}

}

