

Московский авиационный институт
(национальный исследовательский университет)

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

Лабораторные работы 1 — 16 по курсу ИСРППС:

Инструментальные средства разработки прикладных программных систем.

- №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, масштабирование Шрифта
- №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. Кривые Безье для авионики
- №14. Обработка прерываний от таймера. Вращение Images and Bitmaps
- №15. Компонента .dll
- №16. Типы проектов .Net.

Работу выполнила:

М08-205-19Б Данилова Т.М.

Группа

ФИО:

Подпись

Руководитель: _____/Семенов А.С./

Подпись:

Дата: 23 апреля 2021 г.

Лабораторная работа №1. Обработка исключений, преобразование типов.

Базовым для всех типов исключений является тип **Exception**. Этот тип определяет ряд свойств, с помощью которых можно получить информацию об исключении.

- **InnerException**: хранит информацию об исключении, которое послужило причиной текущего исключения
- **Message**: хранит сообщение об исключении, текст ошибки
- **Source**: хранит имя объекта или сборки, которое вызвало исключение
- **StackTrace**: возвращает строковое представление стека вызовов, которые привели к возникновению исключения
- **TargetSite**: возвращает метод, в котором и было вызвано исключение

Однако так как тип **Exception** является базовым типом для всех исключений, то выражение `catch (Exception ex)` будет обрабатывать все исключения, которые могут возникнуть.

Но также есть более специализированные типы исключений, которые предназначены для обработки каких-то определенных видов исключений. Их довольно много, я приведу лишь некоторые:

- **DivideByZeroException**: представляет исключение, которое генерируется при делении на ноль
- **ArgumentOutOfRangeException**: генерируется, если значение аргумента находится вне диапазона допустимых значений
- **ArgumentException**: генерируется, если в метод для параметра передается некорректное значение
- **IndexOutOfRangeException**: генерируется, если индекс элемента массива или коллекции находится вне диапазона допустимых значений
- **InvalidCastException**: генерируется при попытке произвести недопустимые преобразования типов
- **NullReferenceException**: генерируется при попытке обращения к объекту, который равен `null` (то есть по сути неопределен)

Текст программы:

```
using System;

namespace lab1
{
    class Date
    {
        private int year = 0;
        private int month = 0;
        private int day = 0;
        public int Year_A
        {
            set
            {
                if (value < 1917 || 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 > 12)
                    throw new ArgumentOutOfRangeException("Month is out of range");
                else
                    month = value;
            }
            get { return month; }
        }
        public int Day_A
        {
            set
            {
                if (value < 1 || value > 31)
                    throw new ArgumentOutOfRangeException("Day is out of range");
                else
                    day = value;
            }
            get
            { return day; }
        }
    }
}

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Step 1: Types");
        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 long: {0} to {1}", long.MinValue, long.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 decimal: {0} to {1}", decimal.MinValue,
decimal.MaxValue);
        Console.WriteLine();

        Console.WriteLine("Step 2: Massive");
    }
}
```

```

float[] arr = new float[3] { 2.7f, 5.1f, 8.2f };
foreach (float f in arr)
    Console.WriteLine("{0} ", f);
Console.WriteLine();

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

}
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"); }

}
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"); }

}
Console.WriteLine("Date: {0}.{1}.{2}", date.Day_A, date.Month_A, date.Year_A);
Console.ReadKey();
}
}
}

```

Пример работы программы:

```
Step 1: Types
size byte: 0 to 255
size int: -2147483648 to 2147483647
size short: -32768 to 32767
size long: -9223372036854775808 to 9223372036854775807
size float: -3,4028235E+38 to 3,4028235E+38
size double: -1,7976931348623157E+308 to 1,7976931348623157E+308
size decimal: -79228162514264337593543950335 to 79228162514264337593543950335

Step 2: Massive
2,7
5,1
8,2

Step 3: Date
Input year:
56
Out of Range, try again
Input year:
2020
Input month:
hello
Wrong format
Input month:
12
Input day:
!
Wrong format
Input day:
1
Date: 1.12.2020
```

Лабораторная работа №2. Паттерны MVC, Facade.

Текст программы:

Model.cs:

```
namespace lab2
{
    public class Model
    {
        public Model() {}
        public void op()
        {
            MessageBox.Show("Hello");
        }
        public int mdl() { return 999; }
    }
}
```

Controller.cs:

```
namespace lab2
{
    static class Controller
    {
        public static bool fb = false;
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Model model = new Model();
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);

            Form1 form1 = new Form1();
            Application.Run(form1);

            if (fb)
            {
                int r = model.mdl();
            }
        }
    }
}
```

View.cs:

```
namespace lab2
{
    public partial class Form1 : Form
    {
        Model model = new Model();
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            model.op();
        }
    }
}
```

```

    }
    private void button2_Click(object sender, EventArgs e)
    {
        this.label1.Text = model.mdl().ToString();
    }
    public static class Facade
    {
        static F2 Sub2 = new F2();
        static F3 Sub3 = new F3();
        static Form2 f2 = new Form2();

        public static void Run()
        {
            Sub2.Run_2();
        }

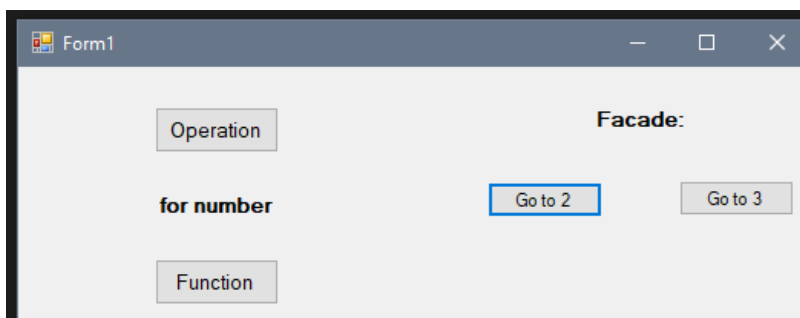
        public static void Run1()
        {
            Sub3.Run_3();
        }

        internal class F2
        {
            Form2 Sub2 = new Form2();
            public void Run_2()
            {
                Sub2.ShowDialog();
            }
        }
        internal class F3
        {
            Form3 Sub3 = new Form3();
            public void Run_3()
            {
                Sub3.ShowDialog();
            }
        }
    }
    private void button3_Click(object sender, EventArgs e)
    {
        Facade.Run();
    }

    private void button4_Click(object sender, EventArgs e)
    {
        Facade.Run1();
    }
}
}

```

Пример работы программы:



Лабораторная работа №3. Delegate Pattern, PictureBox

Текст программы:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Configuration;
using System.Data;
using System.Drawing;
using System.Globalization;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace lab3
{
    delegate void DelegateTool(MouseEventArgs e);
    public partial class Form1 : Form
    {
        Bitmap bitmap = null;
        Graphics gr = null;
        Pen pen = new Pen(Color.Black, 3.5f);
        Pen pBrush = new Pen(new SolidBrush(Color.Blue), 5.5f);

        private DelegateTool _operation;

        public Form1()
        {
            InitializeComponent();
            bitmap = new Bitmap(this.pictureBox1.Width, this.pictureBox1.Height);
            gr = Graphics.FromImage(bitmap);
            //gr = this.pictureBox1.CreateGraphics();
            this._operation = tBrush;
        }

        private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
        {
            //Point p = e.Location;
            //PointF pf1 = p; //new PointF(e.X, e.Y);
            //MessageBox.Show("X = " + e.X + "Y = " + e.Y + "");
            this.textBox1.Text = "X = " + e.X + "    Y = " + e.Y + "";

            if (e.Button != MouseButtons.Left) return;
            //gr.DrawLine(pBrush, e.X, e.Y, e.X + 1, e.Y + 1);
            this._operation(e);
            this.pictureBox1.Image = this.bitmap;
            button1.Visible = true;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.gr.Clear(this.pictureBox1.BackColor);
            this.pictureBox1.Image = this.bitmap;
            button1.Visible = false;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            ToolBrush(); // _operation = tBrush;
        }
    }
}
```



```

private void button3_Click(object sender, EventArgs e)
{
    ToolPen(); //_operation = tPen;
}
private void button4_Click(object sender, EventArgs e)
{
    _operation = DrawWeb;
}

public void ToolDraw(MouseEventArgs e) { _operation(e); }
public void ToolBrush() { _operation = tBrush; }
public void ToolPen() { _operation = tPen; }

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

private void DrawWeb(MouseEventArgs e)
{
    PointF pointf;
    int step = 6;

    this.gr.Clear(this.pictureBox1.BackColor);
    this.pictureBox1.Image = this.bitmap;

    for (int i = 0; i < step; i++)
    {
        pointf = new PointF(0, (i * (this.pictureBox1.Size.Height)/(step-1)));
        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-1));
        pointf.Y = 0;
        this.gr.DrawLine(pen, pointf, e.Location);

        pointf.Y = this.pictureBox1.Size.Height;
        this.gr.DrawLine(pen, pointf, e.Location);
    }
} // end class
}

```

Пример работы программы:

Form1

ToolBrush

ToolPen

DrawWeb

X = 493 Y = 79

Form1

ToolBrush

ToolPen

DrawWeb

Clear

X = 502 Y = 281

Form1

ToolBrush

ToolPen

DrawWeb

Clear

X = 156 Y = 186

Лабораторная работа №4. Компонента MessageBox.

Параметры:

owner [IWin32Window](#)

Реализация [IWin32Window](#), которой будет принадлежать данное модальное диалоговое окно.

text [String](#)

Текст, отображаемый в окне сообщения.

caption [String](#)

Текст для отображения в строке заголовка окна сообщения.

buttons [MessageBoxButtons](#)

Одно из значений [MessageBoxButtons](#), указывающее, какие кнопки отображаются в окне сообщения.

icon [MessageBoxIcon](#)

Одно из значений [MessageBoxIcon](#), указывающее, какой значок отображается в окне сообщения.

defaultButton [MessageBoxDefaultButton](#)

Одно из значений [MessageBoxDefaultButton](#), определяющее кнопку по умолчанию для данного окна сообщения.

options [MessageBoxOptions](#)

Одно из значений [MessageBoxOptions](#), указывающее, какие параметры отображения и связи будут использованы для этого окна сообщения. Чтобы использовать значения по умолчанию, следует передать значение 0.

helpFilePath [String](#)

Полный путь к файлу справки, который должен отображаться, когда пользователь щелкнет кнопку "Справка".

keyword [String](#)

Ключевое слово справки, отображаемое по щелчку кнопки "Справка".

Текст 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 lab4
{
```

```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        MessageBox.Show("hi 1");
        MessageBox.Show("hi", "test 2");
        MessageBox.Show("hi", "test 3", MessageBoxButtons.OKCancel);
        MessageBox.Show("hi", "test 4", MessageBoxButtons.OKCancel,
        MessageBoxIcon.Information);
        MessageBox.Show("hi", "test 5", MessageBoxButtons.OKCancel,
        MessageBoxIcon.Information, MessageBoxDefaultButton.Button2);
    }

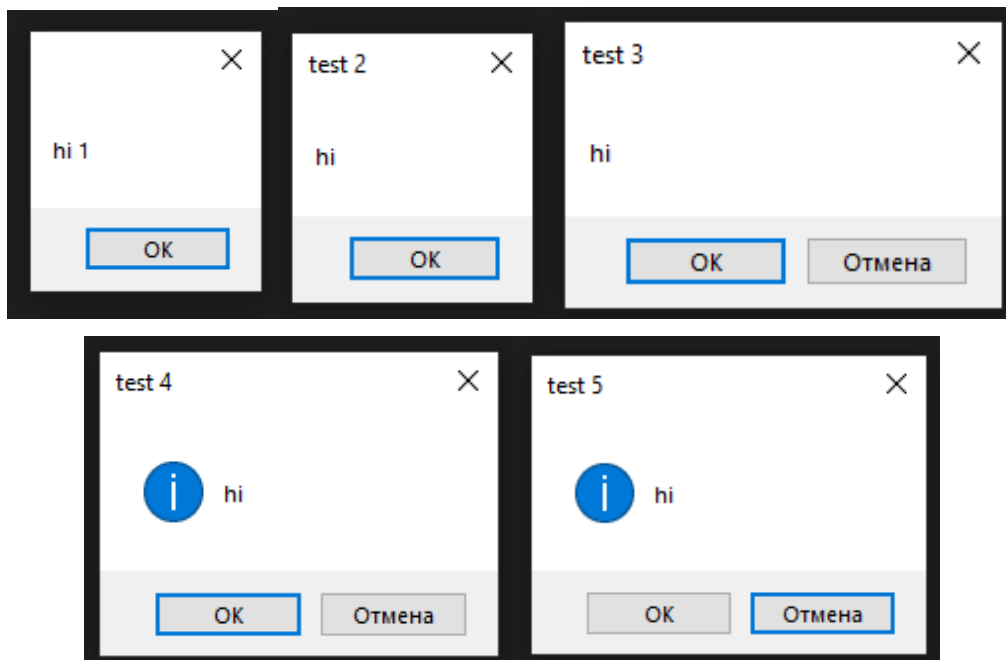
    private void button2_Click(object sender, EventArgs e)
    {
        DialogResult result = MessageBox.Show("Are you sure?", "Exit",
        MessageBoxButtons.OKCancel, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);
        switch (result)
        {
            case DialogResult.OK:
                Application.Exit();
                break;
            case DialogResult.Cancel:
                MessageBox.Show("Cancel");
                break;
            default: break;
        }
    }

    private void button3_Click(object sender, EventArgs e)
    {
        Random random = new Random();
        this.BackColor = Color.FromArgb(random.Next(255), random.Next(255),
        random.Next(255));
        DialogResult result = MessageBox.Show("Maybe you want to change the color else?",
        "Color", MessageBoxButtons.RetryCancel, MessageBoxIcon.Question);
        switch (result)
        {
            case DialogResult.Retry:
                this.button3_Click(sender, e);
                break;
            case DialogResult.Cancel:
                MessageBox.Show("Well, OK");
                break;
            default: break;
        }
    }
}

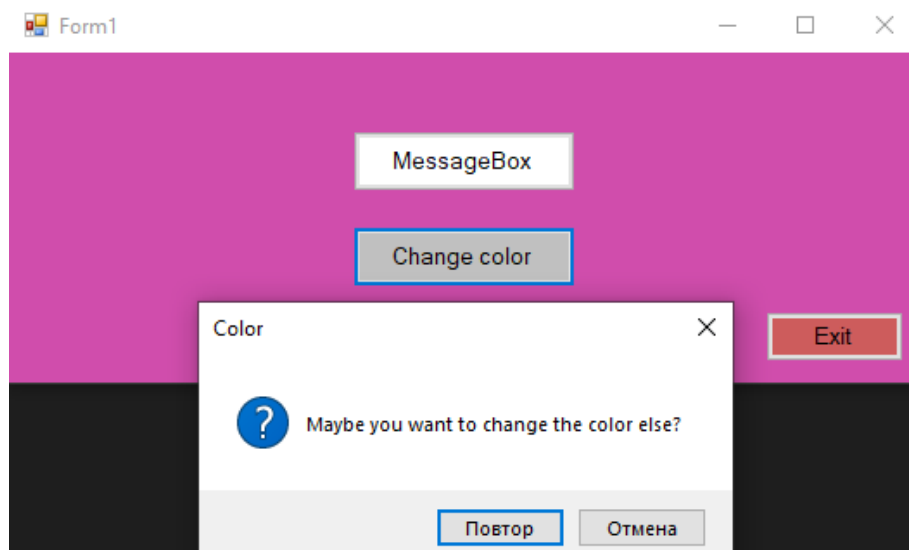
```

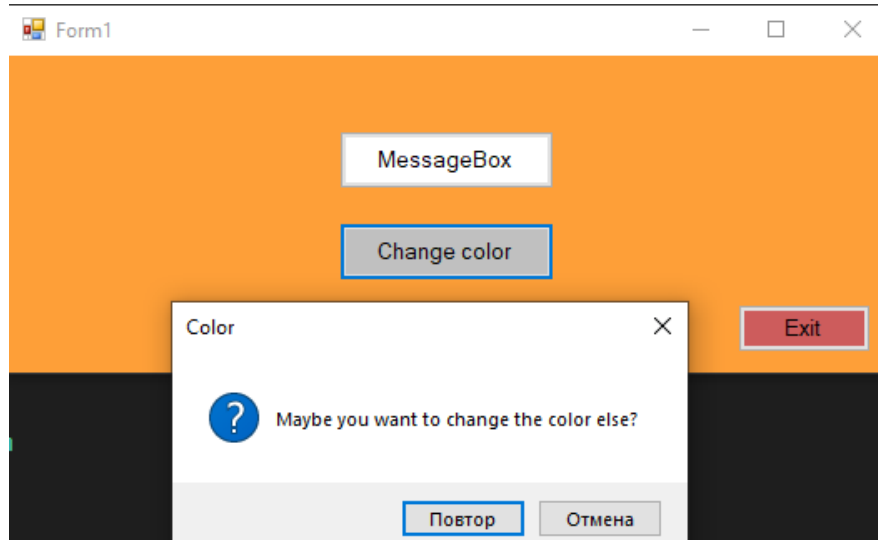
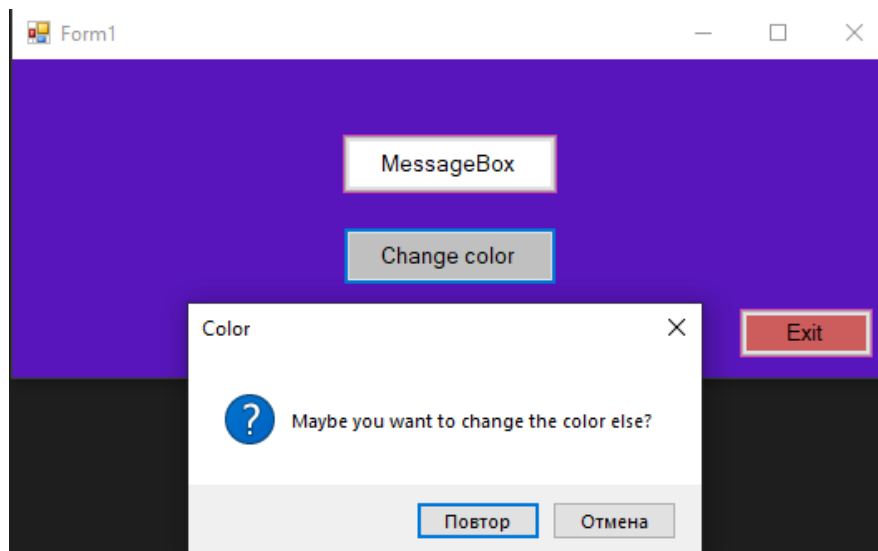
Пример работы программы:

1) Разные перегрузки:

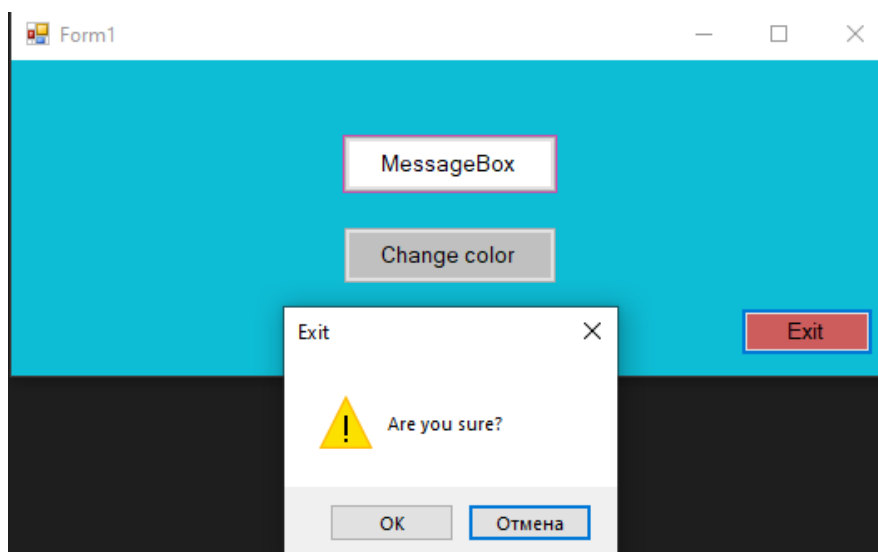


2) Диалог с пользователем (возможность повторить действие):





3) Запрос подтверждения действия:



Лабораторная работа №5. Паттерны Abstract Factory и Singleton.

Текст программы:

```
namespace lab5
{
    public partial class Form1 : Form
    {
        private static Form1 f1 = null;    //step2

        AbstractFactory bank = null;
        Client client = null;
        TestClass testclass = null;

        private Form1()    //step 1: Скрытый конструктор
        {
            InitializeComponent();
            bank = new BankSber();
            testclass = Singleton<TestClass>.CreatorInstance;
        }

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

        private void button1_Click(object sender, EventArgs e)
        {
            this.client = new Client(new BankSber());
            this.client.Run();
            this.label1.Text = this.client.abstractProductA.account.ToString();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            MessageBox.Show(fA.testclass.TestProc());
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.client = new Client(new BankVTB());
            this.client.Run();
            this.label2.Text = this.client.abstractProductB.account.ToString();
        }
    } //end Form1

    public class Singleton<T> where T : class
    {
        protected Singleton(){ MessageBox.Show("Constructor Singleton<T>"); }

        private sealed class SingletonCreator<S> where S : class
        {
            public static S Instance { get; } = (S)typeof(S).GetConstructor(
                BindingFlags.Instance | BindingFlags.NonPublic,
                null,
                new Type[0],

```

```

        new ParameterModifier[0]).Invoke(null);
    } //end class SingletonCreator
    public static T CreatorInstance { get { return SingletonCreator<T>.Instance; } }
} //end Singleton<T>

public class TestClass : Singleton<TestClass>
{
    private TestClass() { MessageBox.Show("Constructor TestClass"); }
    public string TestProc() { return "Hello World!"; } //надо вызывать
}

// *****AbstractFactory*****

abstract class AbstractFactory // factory1, factory2, ...
{
    public abstract AbstractProductA CreateProductA();
    public abstract AbstractProductB CreateProductB();
}

abstract class AbstractProductA
{
    public int account { set; get; } = 222;
}
abstract class AbstractProductB
{
    public int account { set; get; } = 555;
    public abstract void Interact(AbstractProductA a);
}

class BankSber : AbstractFactory
{
    public override AbstractProductA CreateProductA()
    {
        return new CardA();
    }
    public override AbstractProductB CreateProductB()
    {
        return new CardB();
    }
}

class CardA : AbstractProductA
{
    //int account = 111;
}
class CardB : AbstractProductB
{
    public override void Interact(AbstractProductA a)
    {
        MessageBox.Show(this.GetType().Name + " interact with " + a.GetType().Name);
    }
    //int account = 120;
}

class BankVTB : AbstractFactory
{
    public override AbstractProductA CreateProductA()
    {
        return new CardVTB();
    }
}

```



```

        public override AbstractProductB CreateProductB()
        {
            return new CardVTBB();
        }
    }
    class CardVTB : AbstractProductA
    {
        //int account = 222;
    }
    class CardVTBB : AbstractProductB
    {
        public override void Interact(AbstractProductA a)
        {
            MessageBox.Show(this.GetType().Name + " interact with " + a.GetType().Name);
        }
        //int account = 200;
    }
}

```

```

class Client
{
    public AbstractProductA abstractProductA;
    public AbstractProductB abstractProductB;

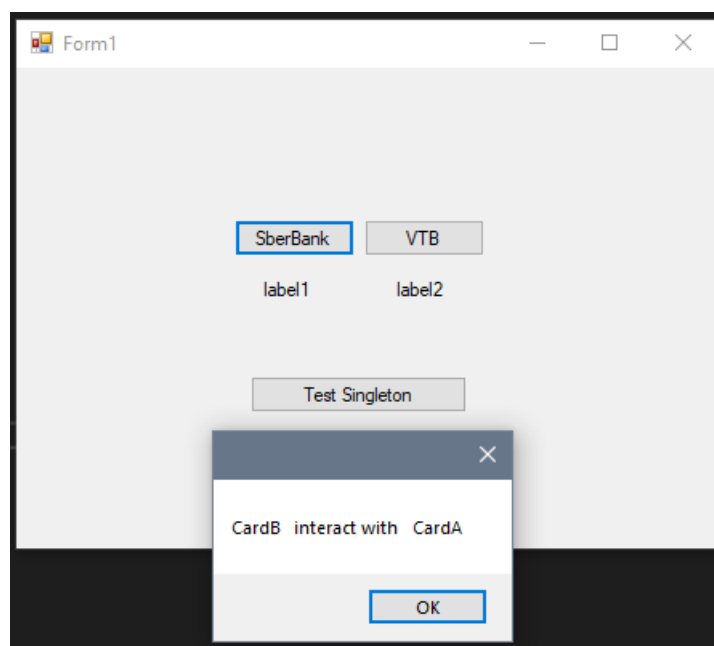
    public Client(AbstractFactory factory)
    {
        abstractProductA = factory.CreateProductA();
        abstractProductB = factory.CreateProductB();
    }

    public void Run()
    {
        abstractProductB.Interact(abstractProductA);
    }
}

```

}

Пример работы программы:



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

Текст программы:

```
namespace lab6
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            this.button1.Visible = true;
        }

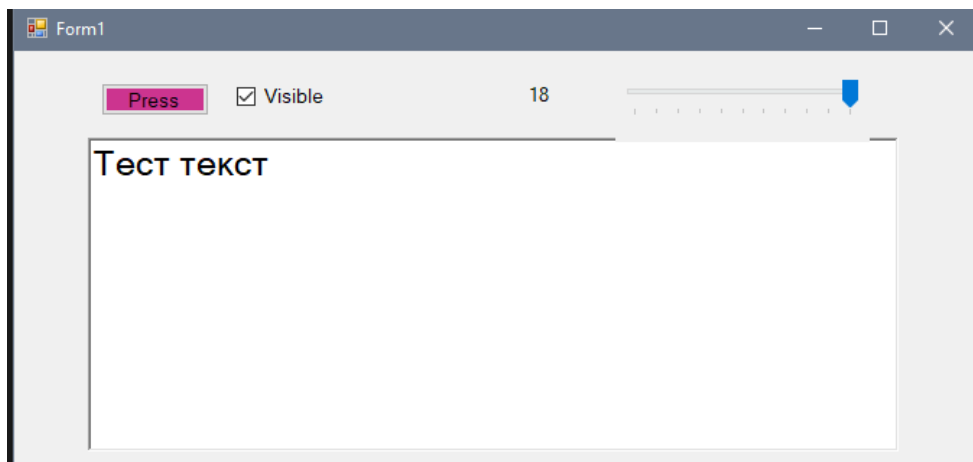
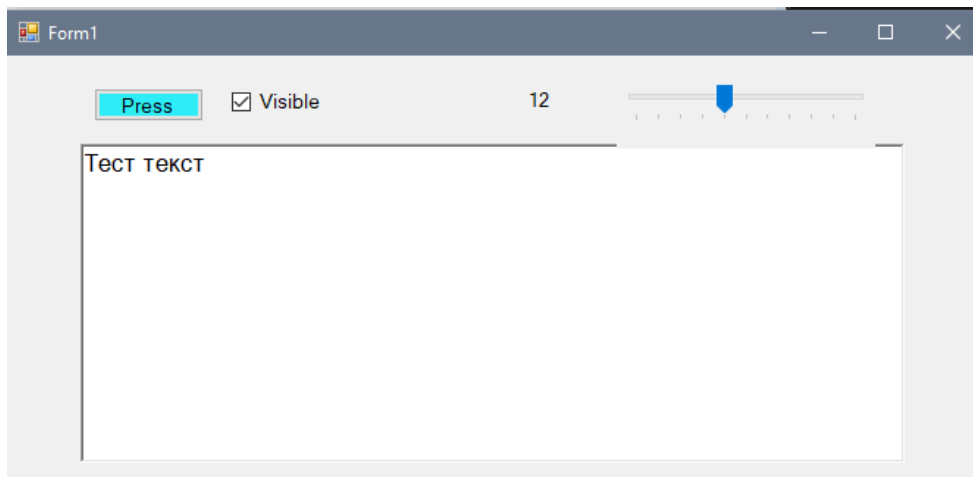
        private void button1_Click(object sender, EventArgs e)
        {
            if (this.richTextBox1.Text == "") { this.richTextBox1.Text =
                "High above the city, on a tall column, stood the statue of the Happy
                Prince. He was gilded all over with thin leaves of fine gold, for eyes he had two bright
                sapphires, and a large red ruby glowed on his sword-hilt"; }
            else
            {
                Random random = new Random();
                if (this.button1.BackColor != Color.Gold) { this.richTextBox1.SelectionColor
                    = this.button1.BackColor; }
                Color color = Color.FromArgb(random.Next(255), random.Next(255),
                    random.Next(255));
                this.button1.BackColor = color;
            }
        }

        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            if (this.checkBox1.Checked)
                //MessageBox.Show("checkBox1.Checked");
                this.button1.Visible = true;
            else
                //MessageBox.Show("un checkBox1.Checked");
                this.button1.Visible = false;
        }

        private void label1_Click(object sender, EventArgs e)
        {
        }

        private void trackBar1_Scroll(object sender, EventArgs e)
        {
            this.richTextBox1.Font = new Font(this.richTextBox1.Font.FontFamily,
                trackBar1.Value);
            this.label1.Text = this.trackBar1.Value.ToString();
        }
    }
}
```

Пример работы программы:



Лабораторная работа №7. Паттерн Chain-responsibility

Текст программы:

```
namespace lab7
{
    public partial class Form1 : Form
    {
        int[] request = new int[3];
        ChainofResponsibility chr = new ChainofResponsibility();

        public Form1()
        {
            InitializeComponent();
        }
        public class ChainofResponsibility
        {
            Handler h1 = new H1();
            Handler h2 = new H2();
            Handler h3 = new H3();

            public ChainofResponsibility()
            {
                h1.successor = h2;
                h2.successor = h3;
                h3.successor = null;
            }

            public void run(int[] request)
            {
                foreach (int i in request)
                {
                    h1.HandlerRequest(i);
                }
            }
        }
        abstract class Handler
        {
            public Handler successor { set; get; } = null;

            public abstract void HandlerRequest(int action);
        }
        class H1 : Handler
        {
            public override void HandlerRequest(int action)
            {
                if (action == 1)
                    MessageBox.Show("1 request");
                else if (successor != null)
                    successor.HandlerRequest(action);
            }
        }
        class H2 : Handler
        {
            public override void HandlerRequest(int action)
            {
                if (action == 2)
                    MessageBox.Show("2 request");
                else if (successor != null)
                    successor.HandlerRequest(action);
            }
        }
    }
}
```

```

        successor.HandlerRequest(action);
    }
}
class H3 : Handler
{
    public override void HandlerRequest(int action)
    {
        if (action == 3)
            MessageBox.Show("3 request");
        else if (successor != null)
            successor.HandlerRequest(action);
    }
}
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox1.Checked) { int i = 0; request[i] = 1; }
    else { int i = 0; request[i] = 0; }
}

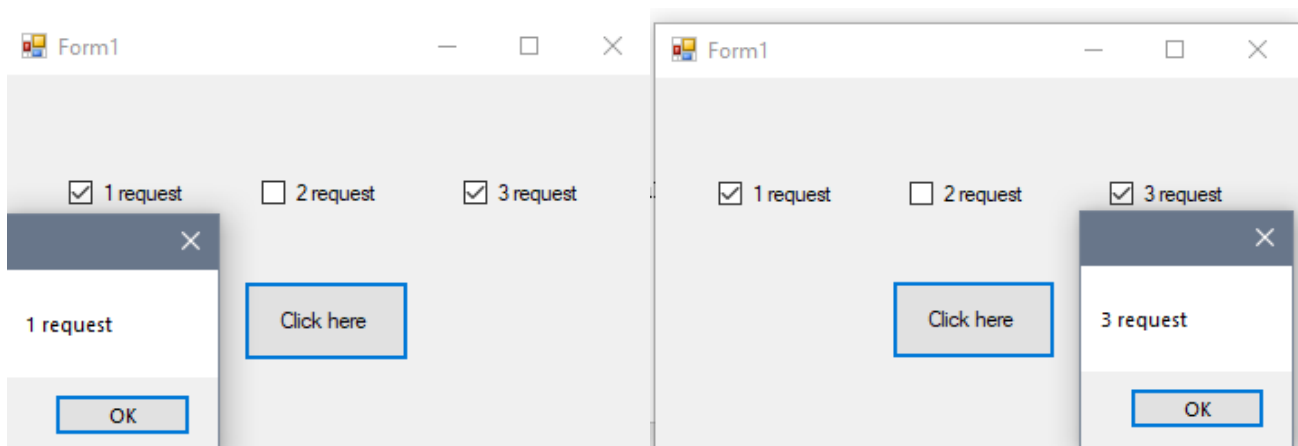
private void checkBox2_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox2.Checked) { int i = 1; request[i] = 2; }
    else { int i = 1; request[i] = 0; }
}

private void checkBox3_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox3.Checked) { int i = 2; request[i] = 3; }
    else { int i = 2; request[i] = 0; }
}

private void button1_Click(object sender, EventArgs e)
{
    if ((checkBox1.Checked) || (checkBox2.Checked) || (checkBox3.Checked))
        chr.run(request);
    else MessageBox.Show("Please, select a mark");
}
}
}

```

Пример работы программы:



Лабораторная работа №8. Паттерн Command

Текст программы:

Command.cs

```
namespace lab8
{
    interface Command
    {
        void Execute();
        void UnExecute();
    }

    class ConcretCommand : Command
    {
        char @operator;
        int operand;
        Calculator calculator;
        // Constructor
        public ConcretCommand(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; }
        }

        // +++ interface
        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;
            }
            return undo;
        }
    }
}
```

```

        undo = '*';
        break;
    default:
        undo = ' ';
        break;
    }
    return undo;
}
}

class Calculator
{
    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 '/':
                curr /= operand;
                break;
        }

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

// "Invoker" : вызывающий
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)
    {
        //Console.WriteLine("\n---- Redo {0} levels ", levels);
        // Делаем возврат операций
        for (int i = 0; i < levels; i++)
            if (_current < _commands.Count - 1)
                _commands[_current++].Execute();
    }
    public void Undo(int levels)
    {
        // Console.WriteLine("\n---- Undo {0} levels ", 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 ConcretCommand(
        _calculator, @operator, operand);
    command.Execute();
    // Добавляем операцию к списку отмены
    _commands.Add(command);
    _current++;
}
}

```

```

}

```

Form1.cs

```

namespace lab8
{
    public partial class Form1 : Form
    {
        User user = new User();
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //MessageBox.Show(" " + Convert.ToInt32(this.textBox1.Text));
            user.Compute('+', Convert.ToInt32(this.textBox1.Text));
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.textBox1.Text = user.getResult().ToString();
        }

        private void Undo_Click(object sender, EventArgs e)
        {
            user.Undo(1);
            this.textBox1.Text = user.getResult().ToString();
        }

        private void Redo_Click(object sender, EventArgs e)
        {
            user.Redo(1);
            this.textBox1.Text = user.getResult().ToString();
        }

        private void button4_Click(object sender, EventArgs e) //-
        {
            user.Compute('-', Convert.ToInt32(this.textBox1.Text));
        }

        private void button3_Click(object sender, EventArgs e) /**
        {
            user.Compute('*', Convert.ToInt32(this.textBox1.Text));
        }

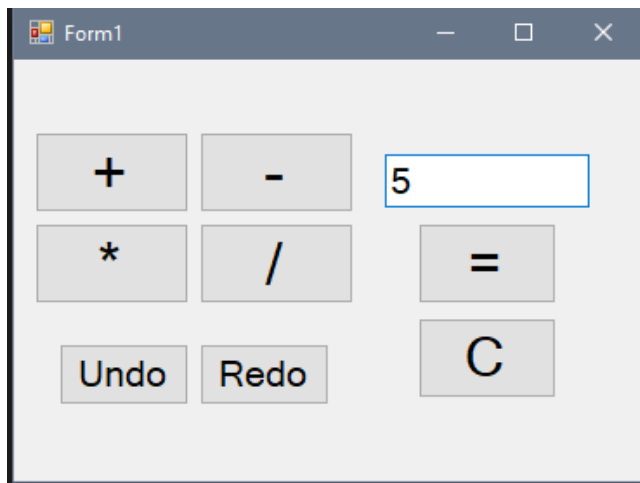
        private void button5_Click(object sender, EventArgs e) ///
        {

```



```
        user.Compute('/', Convert.ToInt32(this.textBox1.Text));  
    }  
  
    private void button6_Click(object sender, EventArgs e)  
    {  
        this.textBox1.Clear();  
    }  
}  
}
```

Пример работы программы:



Лабораторная работа №9. Menus, ToolStrip and StatusStrip

Текст программы:

```
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;
    }

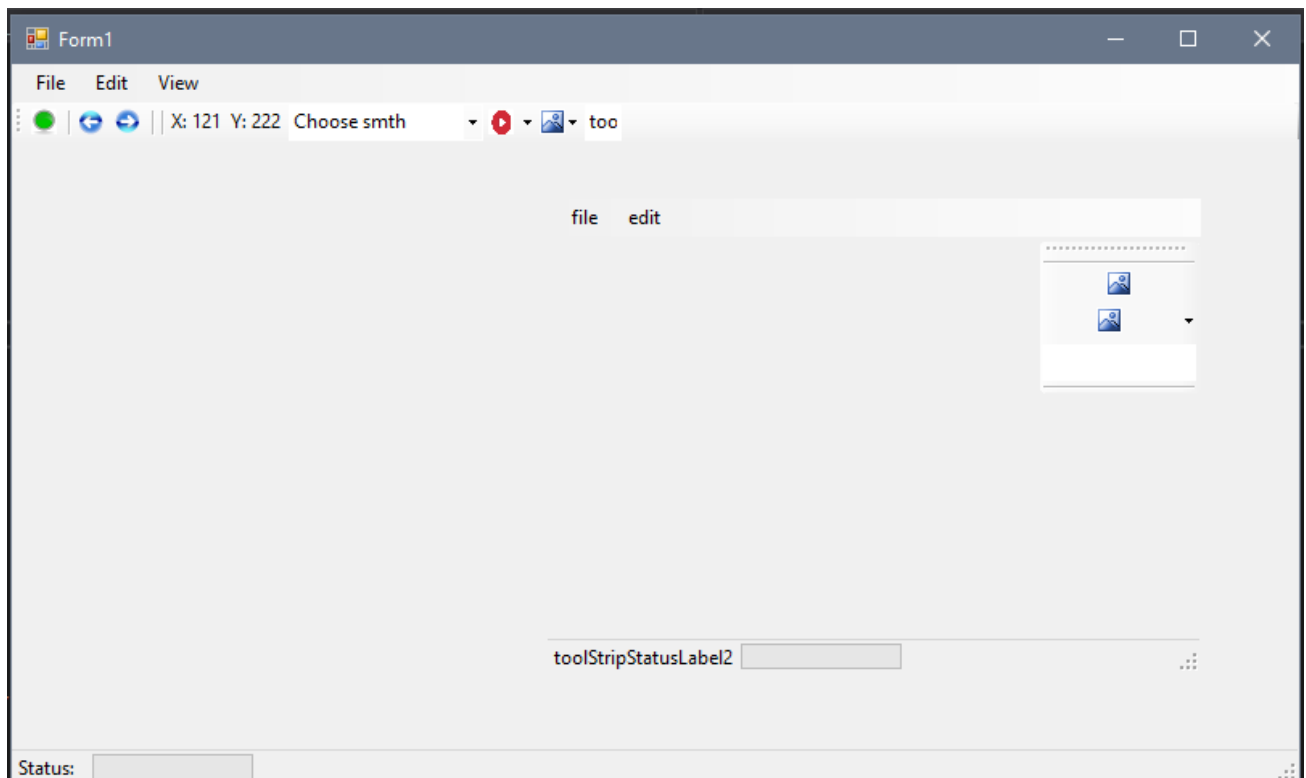
    private void toolStripProgressBar1_Click(object sender, EventArgs e)
    {
        while (this.toolStripProgressBar1.Value < 100)
        { this.toolStripProgressBar1.Value += 50; }
    }

    private void Form1_KeyDown(object sender, KeyEventArgs e)
    {
        if((Control.ModifierKeys==Keys.Control) && (e.KeyCode==Keys.Space))
        {
            Application.Exit();
        }
    }

    private void 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));
    }
}
}

```

Пример работы программы:



Лабораторная работа №10. Dialog Boxes, ClipBoard

Текст программы:

```
namespace lab10
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            //this.richTextBox1.Text = " Example with richTextBox ";
        }

        private void button1_Click(object sender, EventArgs e)
        {
            DialogResult dr = this.colorDialog1.ShowDialog();
            switch (dr)
            {
                case DialogResult.OK:
                    this.label1.ForeColor = 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 openToolStripMenuItem_Click(object sender, EventArgs e)
        {
            DialogResult dr = this.openFileDialog1.ShowDialog();
            switch (dr)
            {
                case DialogResult.OK:
                    using (Stream fileopen = this.openFileDialog1.OpenFile())
                        this.richTextBox1.LoadFile(fileopen, RichTextBoxStreamType.RichText);
                    MessageBox.Show(this.openFileDialog1.FileName);
                    break;
            }
        }

        private void saveToolStripMenuItem_Click(object sender, EventArgs e)
        {
            DialogResult dr = this.saveFileDialog1.ShowDialog();
            switch (dr)
            {
                case DialogResult.OK:
                    MessageBox.Show(this.saveFileDialog1.FileName);
                    try
                    {
                        using (Stream fileopen = this.saveFileDialog1.OpenFile())
```

```

        this.richTextBox1.SaveFile(fileopen,
RichTextBoxStreamType.RichText);
    }
    catch (IOException exc)
    {
        MessageBox.Show(exc.Message, "Error");
        return;
    }
    break;
default:
    break;
}
}

private void button3_Click(object sender, EventArgs e)
{
    DialogResult dr = this.folderBrowserDialog1.ShowDialog();
    switch (dr)
    {
        case DialogResult.OK:
            MessageBox.Show(this.folderBrowserDialog1.RootFolder.ToString());
            MessageBox.Show(this.folderBrowserDialog1.SelectedPath.ToString());
            break;
    }
}

private void richTextBox1_TextChanged(object sender, EventArgs e)
{
    this.richTextBox1.SelectionColor = this.label1.ForeColor;
    this.richTextBox1.Font = this.label1.Font;
}

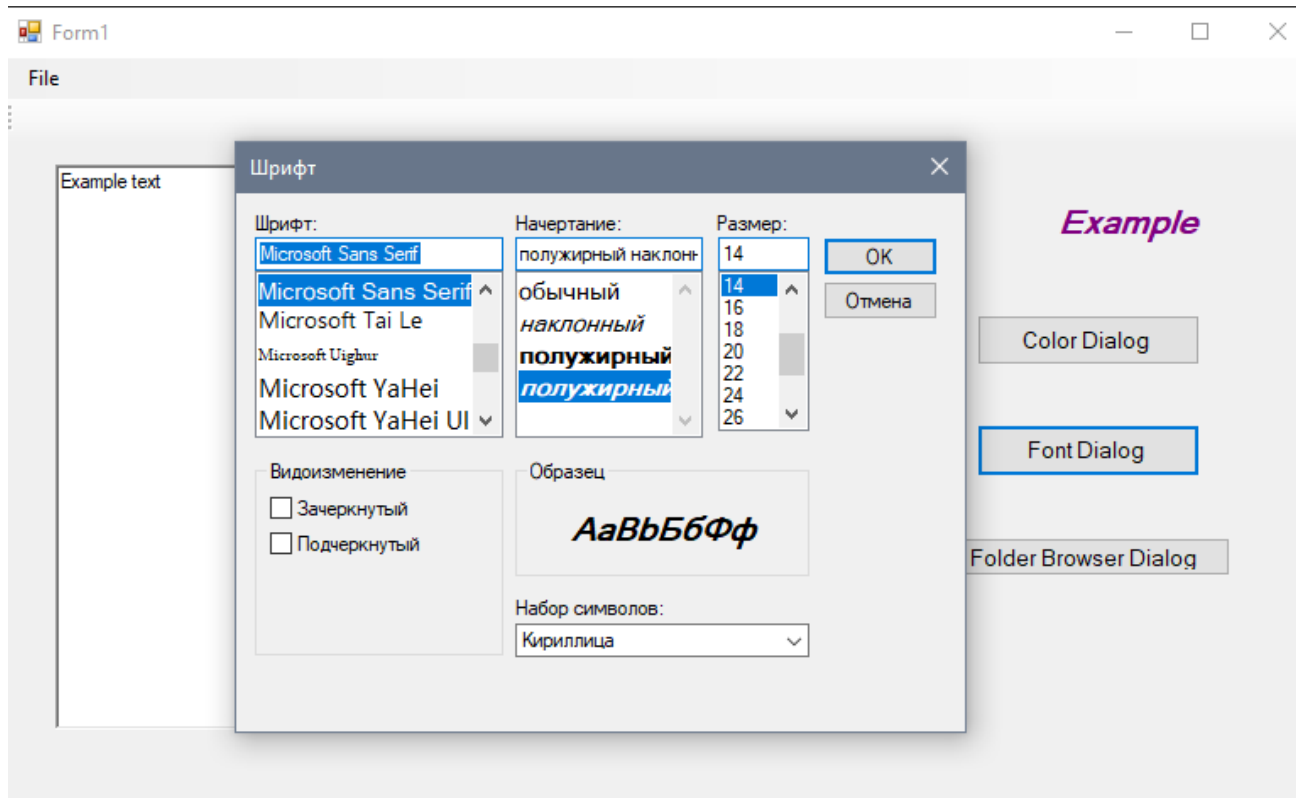
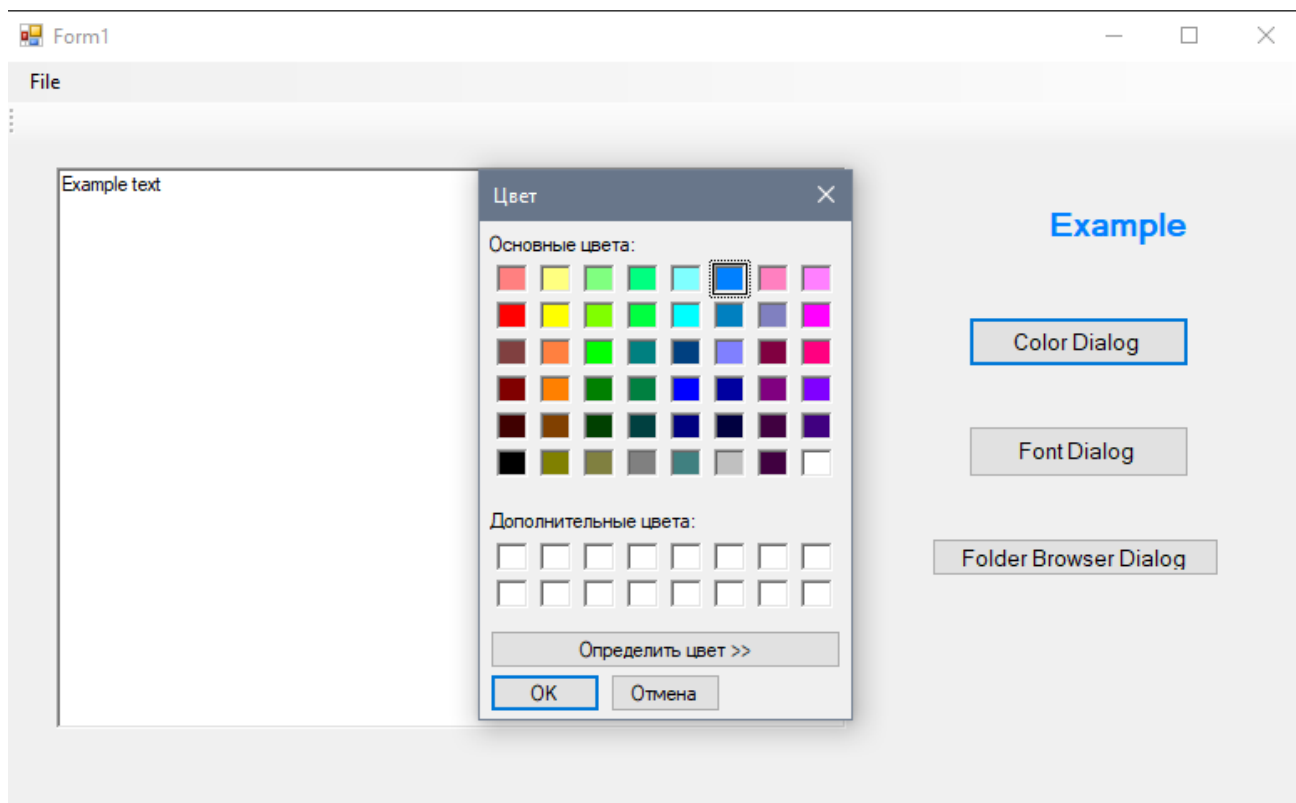
private void copyToolStripMenuItem_Click(object sender, EventArgs e)
{
    Clipboard.SetText(this.richTextBox1.Text);
}

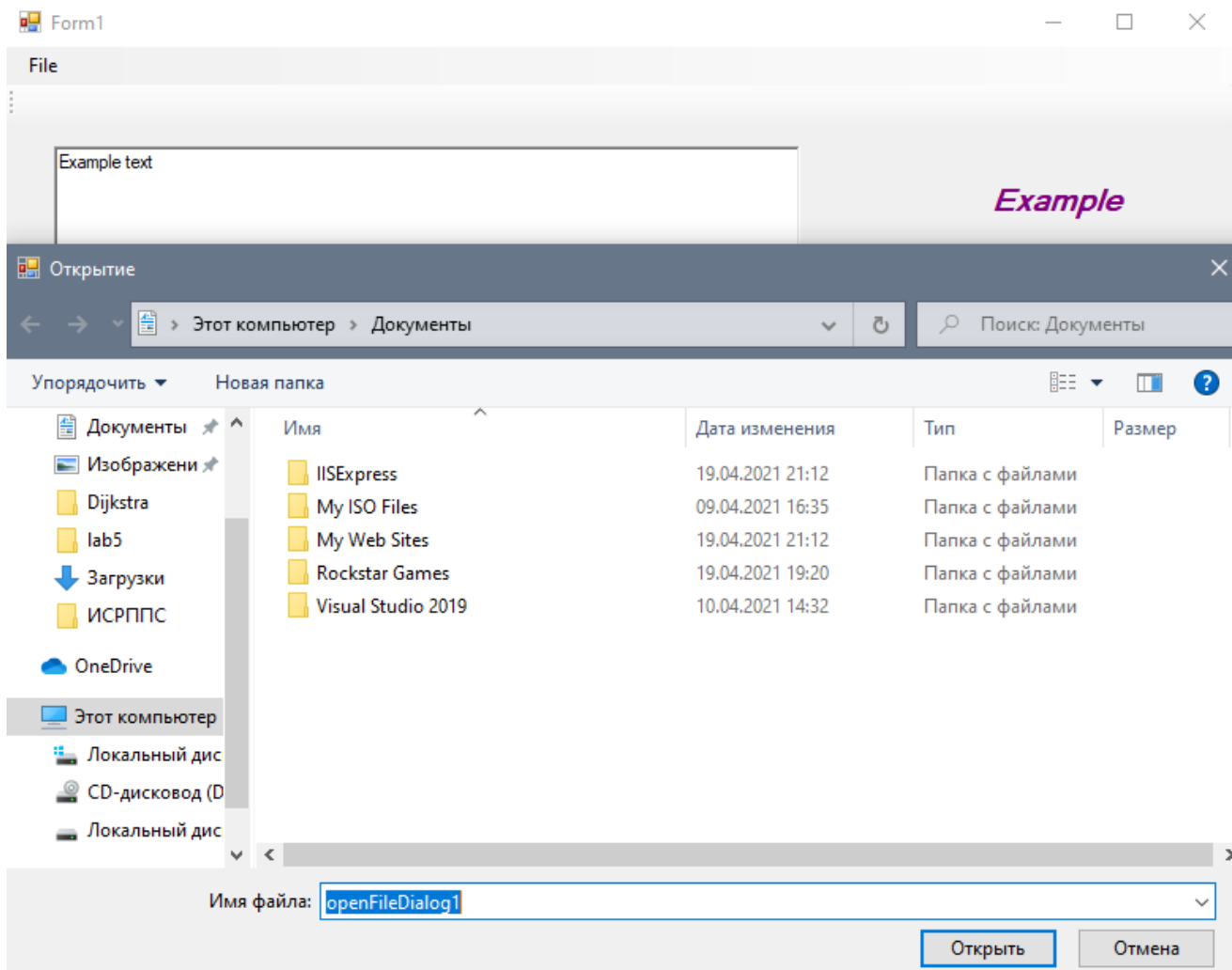
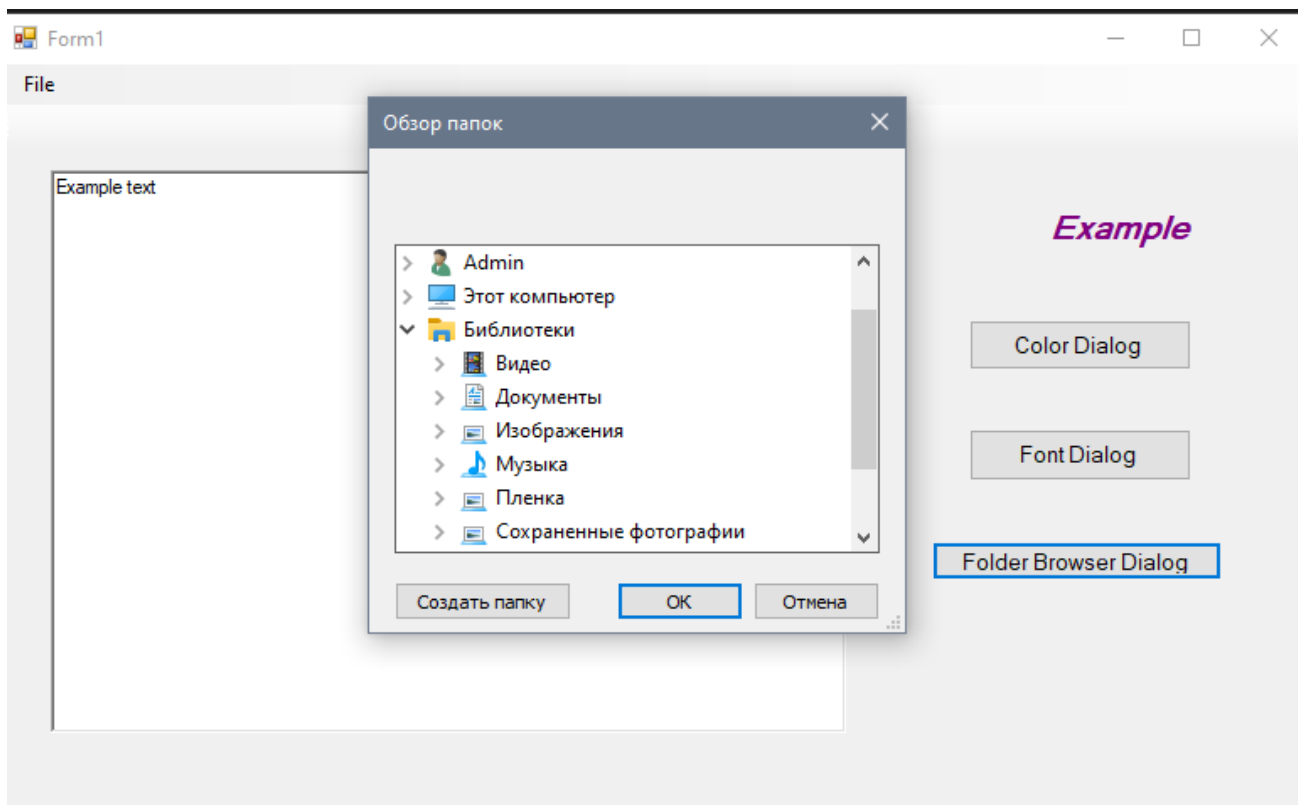
private void addToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.richTextBox1.Text = Clipboard.GetText();
}

private void label1_Click(object sender, EventArgs e)
{
    this.label1.ForeColor = DefaultBackColor;
    this.label1.Font = DefaultFont;
}
}
}

```

Пример работы программы:





Лабораторная работа №11. Tree View, List View

Текст программы:

```
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"));

            //treeView1.Nodes[2].Clone(treeView1.Nodes[0].Clone());!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
            //!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

            // УДАЛЕНИЕ УЗЛОВ
            //treeView1.Nodes[0].Nodes.RemoveAt(0);
            //treeView1.Nodes.Remove(Node3);
            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 заданного 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 заполнения 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 = "";
    label11.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();
}
}
}

```

Пример работы программы:

Form1

FileName	Size	LastAccses
<input type="checkbox"/> DumpStack.Lo...	8192	21.04.2021 11:2...
<input type="checkbox"/> pagefile.sys	9148891136	23.04.2021 16:0...
<input type="checkbox"/> swapfile.sys	16777216	21.04.2021 11:2...

☐ Item1
☐ Item2

Выбранный узел: C:\
 Полный путь к узлу: C:\

Size of files: 0

Имя нового узла:

Раскрыть (скрыть) 1-й
 Раскрыть узел 1
 Раскрыть узел 1 со всеми дочерними
 Скрыть узел 1

Добавить узел

Лабораторная работа №12. Drag and Drop, Clip

Drag and Drop

Текст программы:

```
namespace lab12
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            //this.DragButton.MouseDown += new MouseEventHandler(this.Button_MouseDown);

            //this.splitContainer1.Panel1.DragEnter += new
            DragEventHandler(this.Panels_DragEnter);
            //this.splitContainer1.Panel1.DragDrop += new
            DragEventHandler(this.Panels_DragDrop);

            //this.splitContainer1.Panel2.DragEnter += new
            DragEventHandler(this.Panels_DragEnter);
            //this.splitContainer1.Panel2.DragDrop += new
            DragEventHandler(this.Panels_DragDrop);
        }

        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_DragEnter(object sender, DragEventArgs e)
        {
            e.Effect = DragDropEffects.Move;
        }

        /*private void Panels_DragDrop(object sender, DragEventArgs e)
        {
            ((Button)e.Data.GetData(typeof(Button))).Parent = (Panel)sender;

            if (sender == this.splitContainer1.Panel1)
                this.button1.Location = splitContainer1.Panel1.PointToClient(new Point(e.X,
            e.Y));
            if (sender == this.splitContainer1.Panel2)
                this.button1.Location = splitContainer1.Panel2.PointToClient(new Point(e.X,
            e.Y));
        }*/

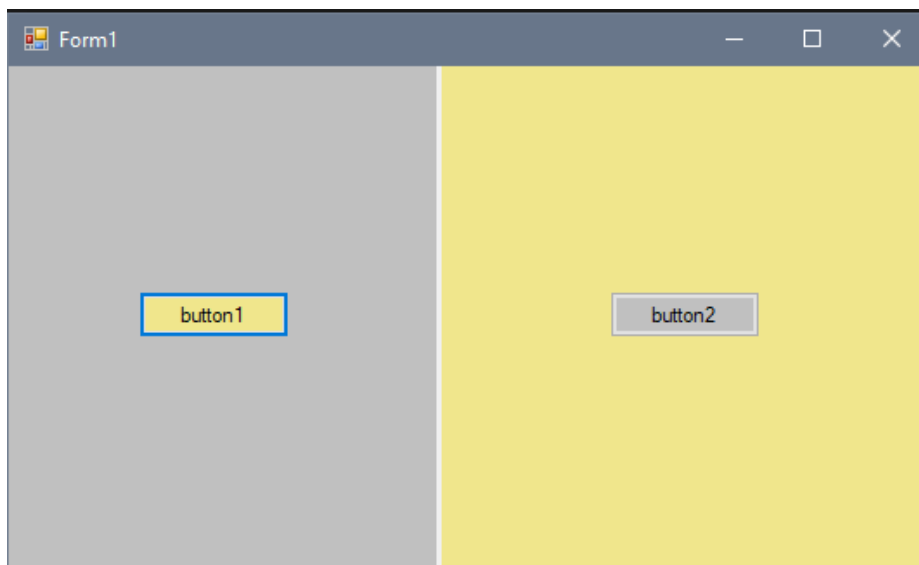
        private void Panel1_DragDrop(object sender, DragEventArgs e)
        {
            this.button1.Parent = (Panel)sender;
        }
    }
}
```

```

        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));
    }
}
}

```

Пример работы программы:



Clip

Текст программы:

```

namespace lab12_2
{
    public partial class Form1 : Form
    {
        private Rectangle clipRect = new Rectangle(0, 0, 80, 80);
        public Form1()
        {
            InitializeComponent();
        }
        private double GetDistance(Point p1, Point p2)
        {
            int a = p2.X - p1.X;
            int b = p2.Y - p1.Y;
            return Math.Sqrt(a*a + b*b);
        }
        private Image ClipImage(Image origIm, Rectangle clipRect)
        {
            Bitmap clipBmp = new Bitmap(clipRect.Width, clipRect.Height);
            using (Graphics g = Graphics.FromImage(clipBmp))
            {

```

```

        g.DrawImage(origIm, new Rectangle(0, 0, clipRect.Width, clipRect.Height),
clipRect, GraphicsUnit.Pixel);
    }
    return (Image)clipBmp;
}

```

```

private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left || (SystemInformation.MouseButtonsSwapped &&
e.Button == MouseButtons.Right))
    {
        Point topLeft = new Point(clipRect.Left, clipRect.Top);
        Point topRight = new Point(clipRect.Right, clipRect.Top);
        Point bottomLeft = new Point(clipRect.Left, clipRect.Bottom);
        Point bottomRight = new Point(clipRect.Right, clipRect.Bottom);

        Point mouse = e.Location;

        int threshold = 15;

        if (GetDistance(mouse, topLeft) <= threshold)
        {
            topLeft = mouse;
            bottomLeft.X = topLeft.X;
            topRight.Y = topLeft.Y;
        }
        else if (GetDistance(mouse, topRight) <= threshold)
        {
            topRight = mouse;
            bottomRight.X = topLeft.X;
            topLeft.Y = topLeft.Y;
        }
        else if (GetDistance(mouse, bottomLeft) <= threshold)
        {
            bottomLeft = mouse;
            topLeft.X = bottomLeft.X;
            bottomRight.Y = bottomLeft.Y;
        }
        else if (GetDistance(mouse, bottomRight) <= threshold)
        {
            bottomRight = mouse;
            topLeft.X = bottomRight.X;
            bottomLeft.Y = bottomRight.Y;
        }
        else if (clipRect.Contains(mouse))
        {
            topLeft.X = mouse.X - (clipRect.Width / 2);
            topLeft.Y = mouse.Y - (clipRect.Height / 2);
            topRight.X = mouse.X + (clipRect.Width / 2);
            topRight.Y = topLeft.Y;
            bottomLeft.X = topLeft.X;
            bottomLeft.Y = mouse.Y + (clipRect.Height / 2);
            bottomRight.X = topLeft.X;
            bottomRight.Y = bottomLeft.Y;
        }

        int width = topRight.X - topLeft.X;
        int height = bottomLeft.Y - topLeft.Y;
        clipRect = new Rectangle(topLeft.X, topLeft.Y, width, height);
        this.Refresh();
    }
}

```

```

    }

    //Load file
    private void button2_Click(object sender, EventArgs e)
    {
        OpenFileDialog dl = new OpenFileDialog();
        if (dl.ShowDialog() == DialogResult.OK)
        {
            try
            {
                if (pictureBox1.Image != null)
                    pictureBox1.Image.Dispose();
                pictureBox1.Image = Image.FromFile(dl.FileName);
            }
            catch (Exception) { MessageBox.Show("Error"); }
        }
    }

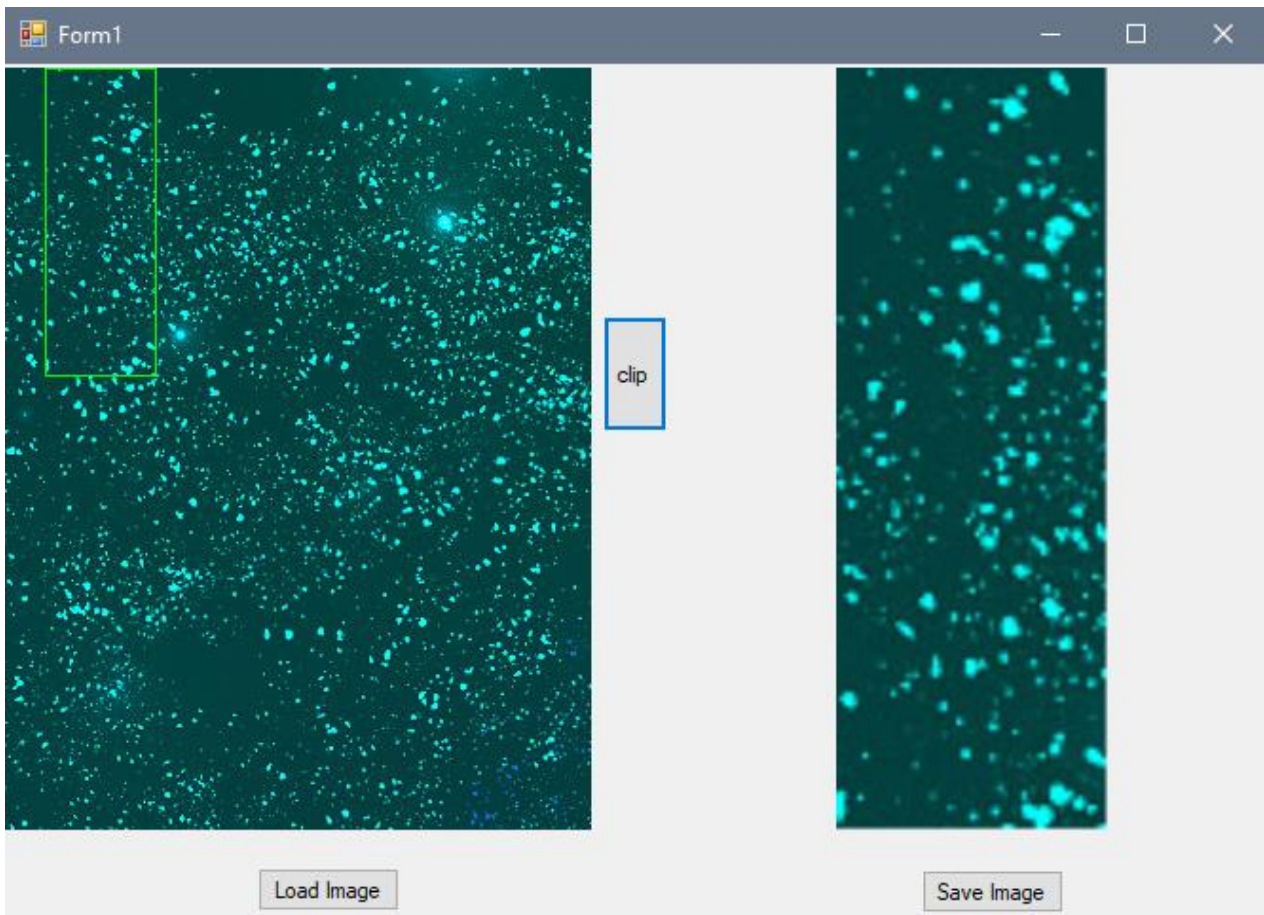
    //Save file
    private void button3_Click(object sender, EventArgs e)
    {
        if (pictureBox2.Image != null)
        {
            SaveFileDialog dl = new SaveFileDialog();
            if (dl.ShowDialog() == DialogResult.OK)
            {
                if (!dl.FileName.EndsWith(".bmp"))
                    dl.FileName += ".bmp";
                pictureBox2.Image.Save(dl.FileName);
            }
        }
    }

    private void pictureBox1_Paint(object sender, PaintEventArgs e)
    {
        e.Graphics.DrawRectangle(Pens.Lime, clipRect);
    }

    //Clip
    private void button1_Click(object sender, EventArgs e)
    {
        if (pictureBox1.Image != null)
            pictureBox2.Image = ClipImage(pictureBox1.Image, clipRect);
    }
}

```

Пример работы программы:



Лабораторная работа №13. Lines, Curves and Area Fills. Кривые Безье для авионики

Текст программы:

```
namespace lab13
{
    public partial class Form1 : Form
    {
        Graphics graph;
        int cx;
        int cy;
        public Form1()
        {
            InitializeComponent();
        }

        Pen penRed5 = new Pen(Color.Red, 5);
        Pen penPink2 = new Pen(Color.Pink, 2);
        private void button1_Click(object sender, EventArgs e)
        {
            // берем дескриптор окна
            graph = Graphics.FromHwnd(this.pictureBox1.Handle);
            cx = pictureBox1.Size.Width;
            cy = pictureBox1.Size.Height - 60;
            graph.DrawLine(penRed5, 0, 0, cx, cy);
            graph.DrawLine(penRed5, cx, 0, 0, cy);
            graph.DrawLine(penRed5, 0, cy, cx, pictureBox1.Size.Height);
            graph.DrawLine(penRed5, cx, cy, 0, pictureBox1.Size.Height);
            graph.DrawLine(penPink2, 0, 0, cx, cy);
            graph.DrawLine(penPink2, cx, 0, 0, cy);
            graph.DrawLine(penPink2, 0, cy, cx, pictureBox1.Size.Height);
            graph.DrawLine(penPink2, cx, cy, 0, pictureBox1.Size.Height);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            graph = 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
            }
            graph.DrawLines(new Pen(Color.Orange, 5), aptf);
            graph.DrawLines(new Pen(Color.LightYellow, 2), aptf);
        }

        SolidBrush solidBrushY = new SolidBrush(Color.Yellow);

        private void button3_Click(object sender, EventArgs e)
        {
            graph = Graphics.FromHwnd(this.pictureBox3.Handle);
            cx = pictureBox3.Size.Width;
            cy = pictureBox3.Size.Height;
            //grfx.Clear(Color.Silver);
            graph.FillEllipse(solidBrushY, 0, 0, cx, cy);
        }
    }
}
```

```

graph.FillEllipse(new SolidBrush(Color.White), 0 + 5, 0 + 5, cx - 10, cy - 10);
graph.FillEllipse(solidBrushY, 0 + 10, 0 + 10, cx - 20, cy - 20);
graph.FillEllipse(new SolidBrush(Color.Black), 0 + 20, 0 + 20, cx - 40, cy - 40);
}

private void button4_Click(object sender, EventArgs e)
{
    graph = Graphics.FromHwnd(this.pictureBox4.Handle);
    cx = pictureBox4.Size.Width;
    cy = pictureBox4.Size.Height;
    //grfx.Clear(Color.Silver);
    graph.DrawEllipse(new Pen(Color.Green, 7), 0, 0, cx, cy);
    graph.DrawEllipse(new Pen(Color.LightSeaGreen, 4), 0, 0, cx, cy);
    graph.DrawEllipse(new Pen(Color.White, 1), 0, 0, cx, cy);
}

private void button5_Click(object sender, EventArgs e)
{
    graph = Graphics.FromHwnd(this.pictureBox5.Handle);
    cx = pictureBox5.Size.Width;
    cy = pictureBox5.Size.Height;
    //grfx.Clear(Color.Silver);

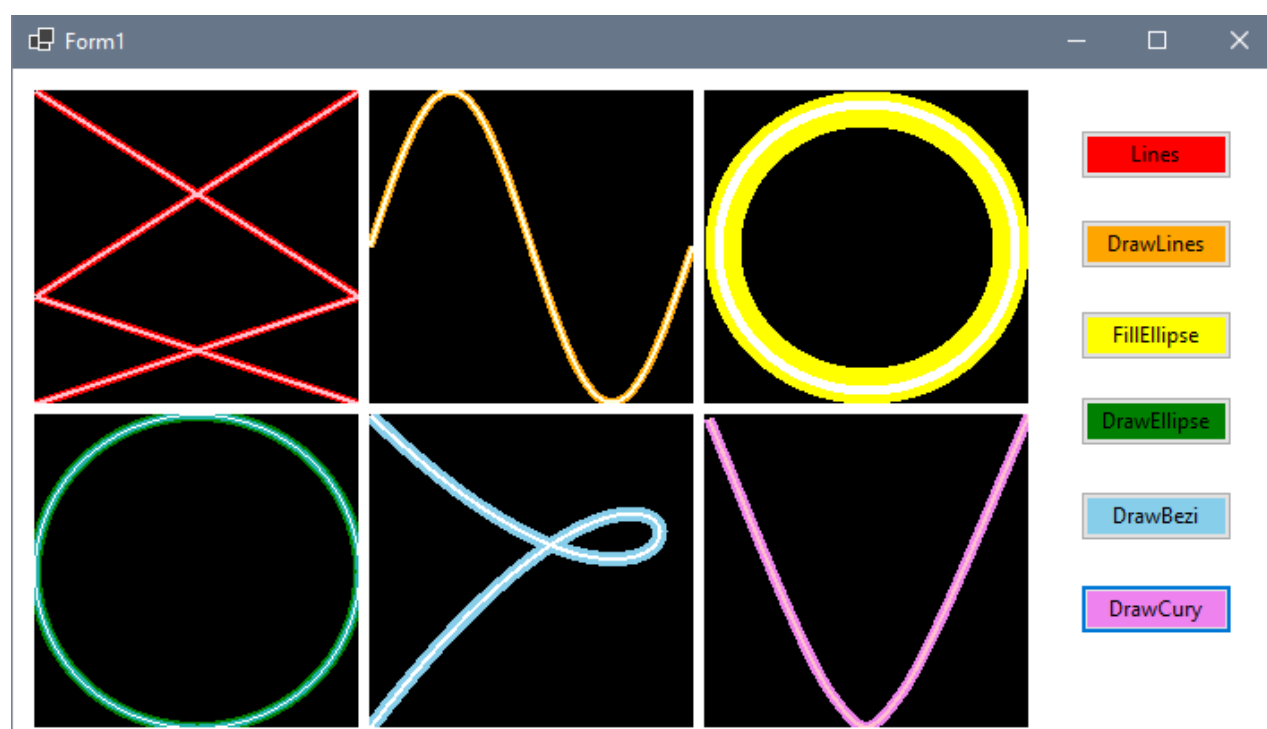
    graph.DrawBezier(new Pen(Color.SkyBlue, 8),
        new PointF(0, 0),
        new PointF(cx + 40, cy + 40),
        new PointF(cx + 30, 0 - 100),
        new PointF(0, cy));
    graph.DrawBezier(new Pen(Color.White, 2),
        new PointF(0, 0),
        new PointF(cx + 40, cy + 40),
        new PointF(cx + 30, 0 - 100),
        new PointF(0, cy));
}

private void button6_Click(object sender, EventArgs e)
{
    graph = 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)};

    graph.DrawCurve(new Pen(Color.Violet, 6), points);
    graph.DrawCurve(new Pen(Color.Pink, 2), points);
}
}
}

```

Пример работы программы:



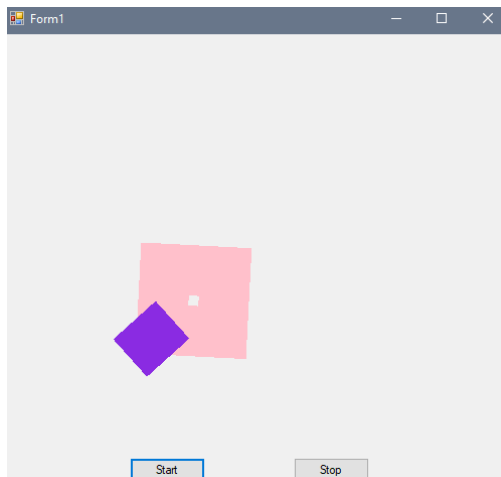
Лабораторная работа №14. Обработка прерываний от таймера. Вращение Images and Bitmaps

Текст программы:

```
namespace lab14
{
    public partial class Form1 : Form
    {
        float angle = 30;
        public Form1()
        {
            InitializeComponent();
            this.timer1.Stop();
            //this.timer1.Interval = 5;
            //this.timer1.Enabled = true;
            //this.timer1.Tick += new EventHandler(timer1_Tick);
        }
        private void pictureBox1_Paint(object sender, PaintEventArgs e)
        {
            e.Graphics.TranslateTransform(this.pictureBox1.Width/2,
            this.pictureBox1.Height/2);
            e.Graphics.RotateTransform(angle);
            e.Graphics.DrawRectangle(new Pen(Color.Pink, 50), 30, 30, 60, 60);
            e.Graphics.DrawLine(new Pen(Color.BlueViolet, 50), 120, 120, 80, 80);

        }
        private void timer1_Tick(object sender, EventArgs e)
        {
            angle += 0.5f;
            this.pictureBox1.Invalidate();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            this.timer1.Start();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            this.timer1.Stop();
        }
    }
}
```

Пример работы программы:



Лабораторная работа №15. Компонента .dll

Текст программы:

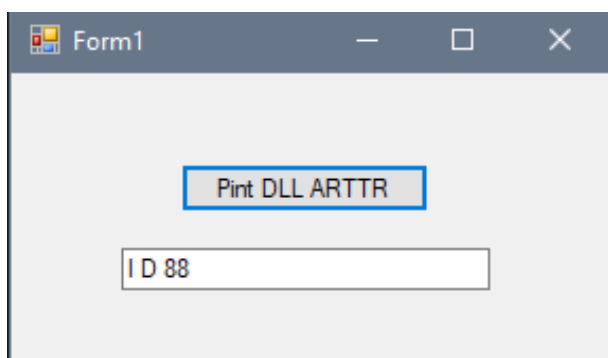
```
namespace lab15_ClassLibrary
{
    public class Reestr
    {
        public string F { set; get; }
        public string I { set; get; }
        public int year { set; get; }

        public Reestr()
        {
            this.F = "I";
            this.I = "D";
            this.year = 88;
        }
    }
}

namespace lab15_WindowsFormsApp
{
    public partial class Form1 : Form
    {
        Reestr reestr = new Reestr();
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.textBox1.Text = reestr.F + " " + reestr.I + " " + reestr.year;
        }
    }
}
```

Пример работы программы:



Лабораторная работа №16. Типы проектов .Net.

ASP.NET

Пример работы программы:

