



**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ
ГОРОДА МОСКВЫ**

**Государственное бюджетное профессиональное
образовательное учреждение города Москвы
«Колледж малого бизнеса № 4»
(ГБПОУ КМБ № 4)**

Отчёты по Лабораторным работам

разработка и тестирование кода

Специальность: 09.02.07 Информационные системы и программирование

Форма обучения: очная

Студент(ка): Геронок Артём Игоревич

Группа: ИПО-22.24

Руководитель: Рыбаков Александр Сергеевич

Отчётная работа защищена с оценкой «___» _____

Москва, 2026 г.

1 Лабораторная работа

Упражнение 2 Создание непрямоугольной формы Windows

В этом упражнении вы создадите треугольную форму Windows.

1 Откройте Visual Studio и создайте новый проект Windows Forms.

Проект откроется с формой по умолчанию с именем Form1 в конструкторе.

2 В окне Properties задайте свойству FormBorderStyle значение

None, а свойству BackColor значение Red. В этом случае форму легче

будет увидеть при тестировании приложения.

3 Перетащите кнопку из Toolbox в левый верхний угол формы.

Задайте свойству Text кнопки значение Close Form.

4 Дважды щелкните кнопку Close Form и добавьте в обработчик

события Button1 Click следующий код:

```
this.Close();
```

5 В конструкторе дважды щелкните форму, чтобы открыть

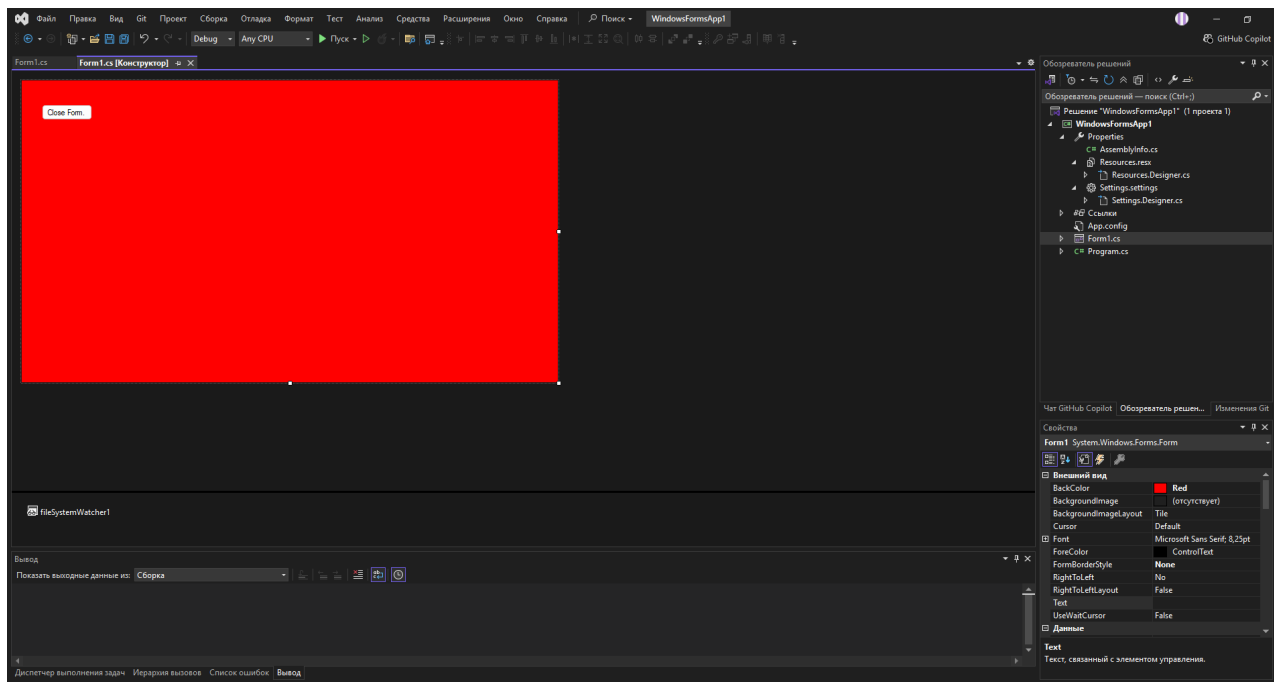
обработчик события Form1 Load. Добавьте в этот метод следующий код

(он задает области формы треугольную форму указанием многоугольника

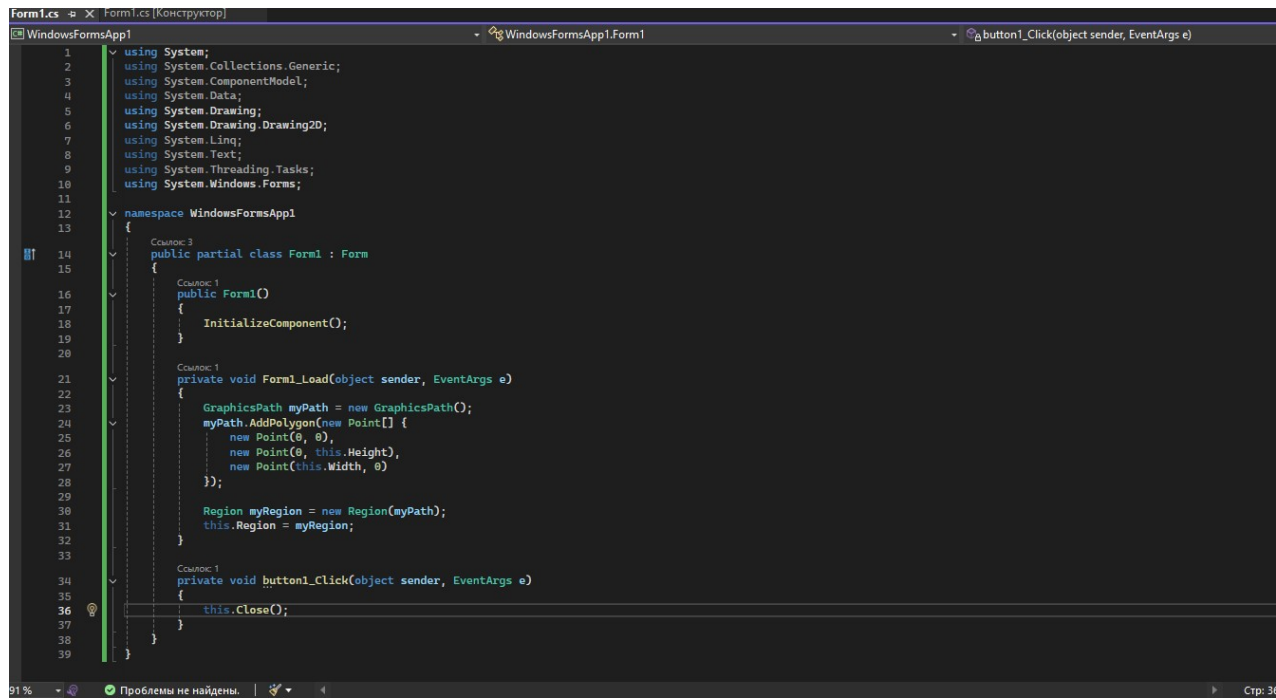
с тремя углами):

```
System.Drawing.Drawing2D.GraphicsPath myPath =  
new System.Drawing.Drawing2D.GraphicsPath();  
myPath.AddPolygon(new Point[] { new Point(0, 0),  
new Point(0, this.Height),  
new Point(this.Width, 0) });  
Region myRegion = new Region(myPath);  
this.Region = myRegion;
```

6 Постройте и запустите приложение. Появится треугольная форма.



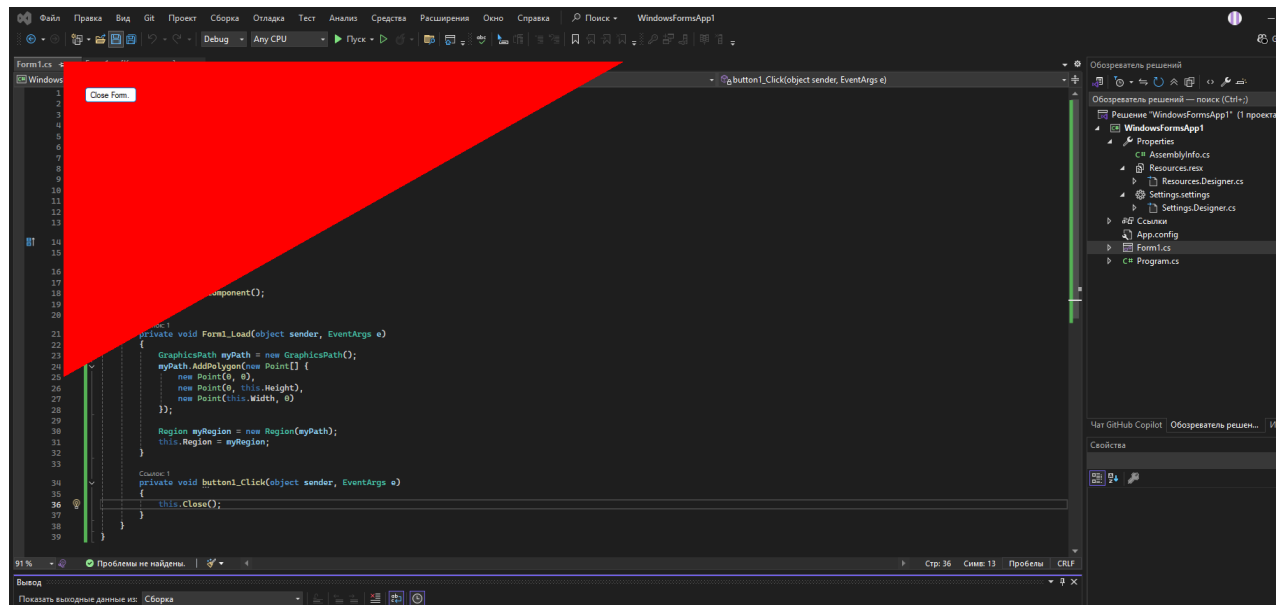
Отчет по Разработка и тестирование кода



The screenshot shows the Visual Studio code editor with the file `Form1.cs` open. The code is for a Windows Forms application. It includes the following code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Drawing.Drawing2D;
7 using System.Linq;
8 using System.Text;
9 using System.Threading.Tasks;
10 using System.Windows.Forms;
11
12 namespace WindowsFormsApp1
13 {
14     public partial class Form1 : Form
15     {
16         public Form1()
17         {
18             InitializeComponent();
19         }
20
21         private void Form1_Load(object sender, EventArgs e)
22         {
23             GraphicsPath myPath = new GraphicsPath();
24             myPath.AddPolygon(new Point[] {
25                 new Point(0, 0),
26                 new Point(0, this.Height),
27                 new Point(this.Width, 0)
28             });
29
30             Region myRegion = new Region(myPath);
31             this.Region = myRegion;
32         }
33
34         private void button1_Click(object sender, EventArgs e)
35         {
36             this.Close();
37         }
38     }
39 }
```

The status bar at the bottom indicates "91%" zoom and "Проблемы не найдены." (No problems found).



Упражнение 3 Создание наследуемой формы

Если у вас имеется уже готовая форма, которую вы собираетесь

использовать в нескольких приложениях, удобно создать наследуемую

(производную) форму. В этом упражнении вы создадите новую форму и

унаследуете ее от существующей базовой формы, а затем измените

производную форму, настроив ее для конкретной работы.

1 Откройте проект из предыдущего упражнения. Базовой формой

для создания производной будет треугольная форма.

2 Для кнопки Close Form задайте свойство Modifiers как protected.

3 Добавьте производную форму: меню Project (Проект) | Add

Windows Form...(Добавить форму Windows), в окне Categories

(Категории) укажите Windows Form, в окне Templates (Шаблоны)

выберите Inherited Form (Наследуемая форма).

4 В окне Add New Item в поле Name укажите название формы:

nForm.cs и нажмите Add для добавления формы.

5 В появившемся окне Inheritance Picker, в котором отображаются

все формы текущего проекта, выберите базовую форму Form1 и нажмите

ОК.

6 Постройте проект.

7 Откройте форму nForm в режиме конструктора. Проверьте, что

она имеет треугольную форму и свойства базовой формы и элемента

управления наследованы.

8 Настройте свойства производной формы:

а. для кнопки:

i. свойство Text – Hello!!!

ii. свойство BackColor – Brown

b. для формы: свойство BackColor – Blue

9 Постройте проект.

10 Задайте производную форму в качестве стартовой, указав в

функции Main следующий код:

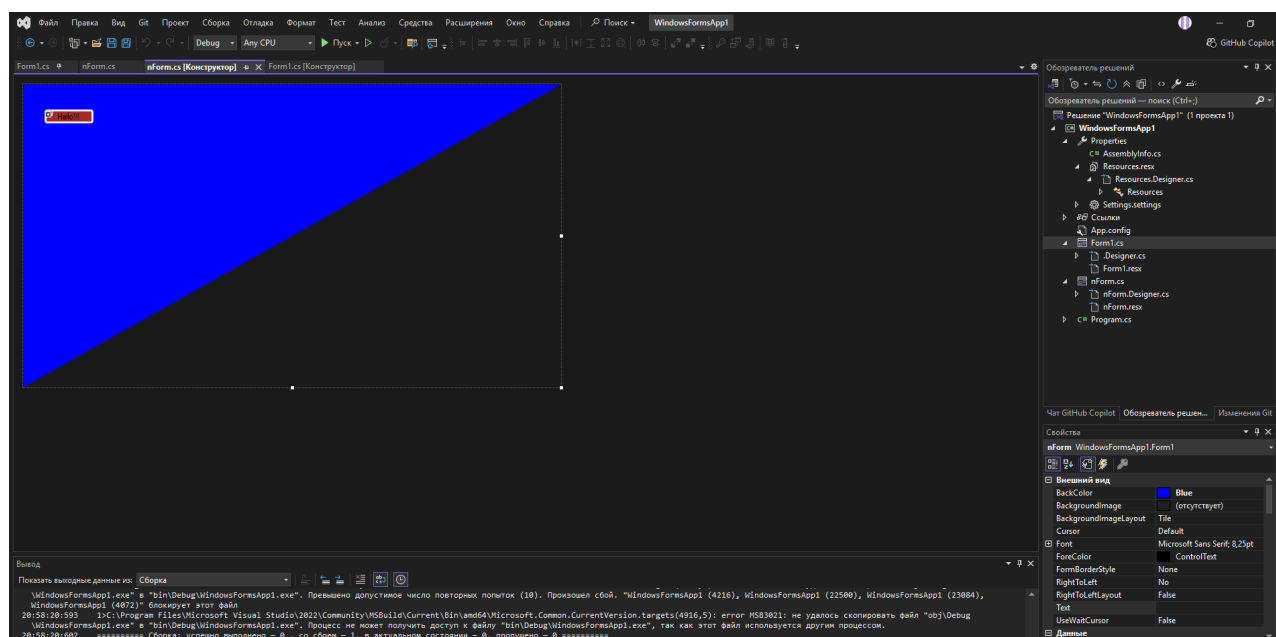
`Application.Run(new nForm());`

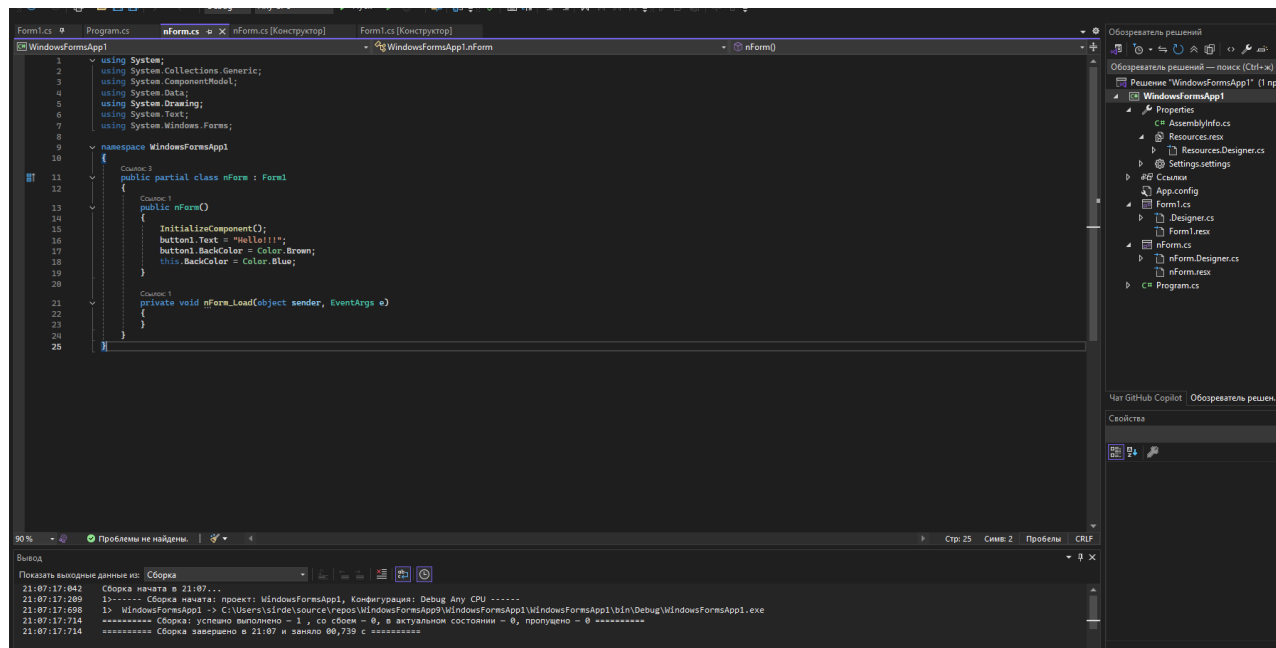
11 Постройте и запустите приложение. Должна открыться

производная форма со своими свойствами.

Проверьте, наследуется ли

заккрытие формы кнопкой.





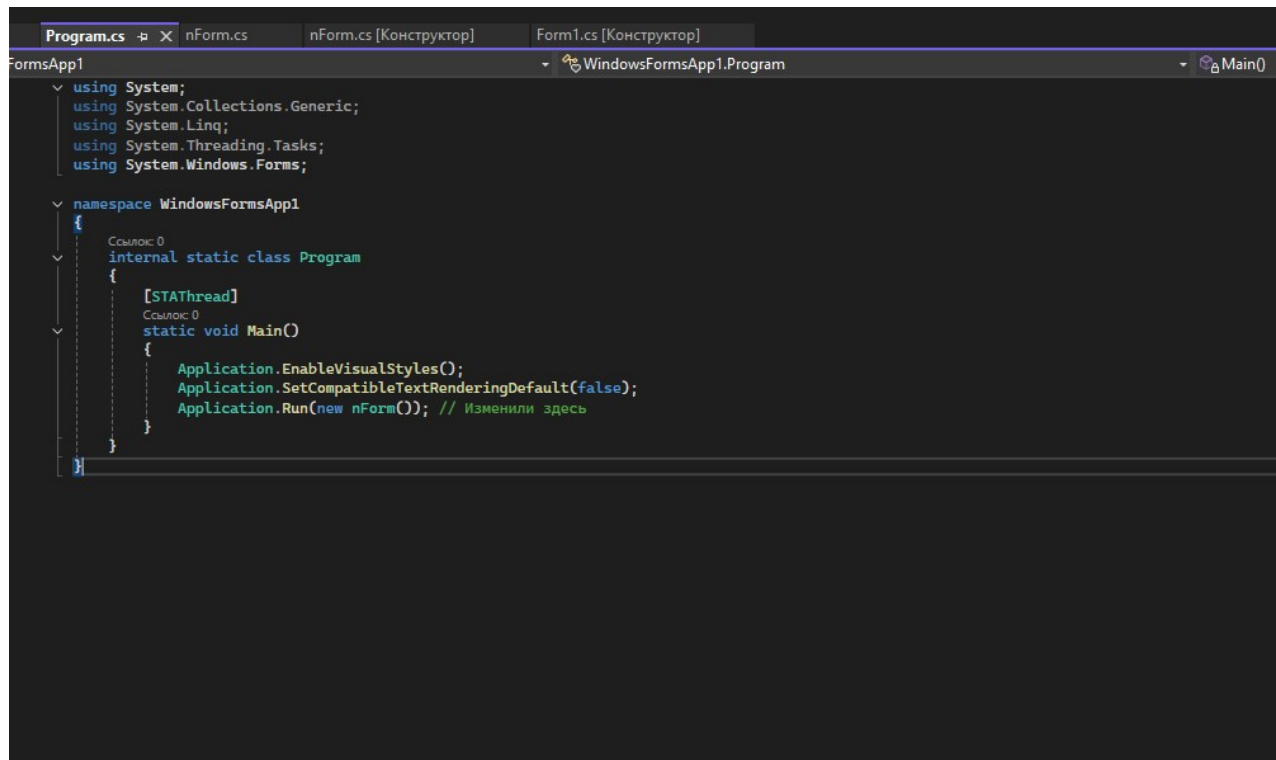
The screenshot shows the Visual Studio IDE with the `nForm.cs` file open. The code defines a partial class `nForm` within the `WindowsFormsApp1` namespace. It includes a constructor `nForm()` and a `Load` event handler `nForm_Load`. The `Load` handler sets the text of a button to "Hello!!!" and its background color to blue. The `Output` window at the bottom shows the successful compilation of the project.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class nForm : Form
    {
        public nForm()
        {
            InitializeComponent();
            button1.Text = "Hello!!!";
            button1.BackColor = Color.Brown;
            this.BackColor = Color.Blue;
        }

        private void nForm_Load(object sender, EventArgs e)
        {
        }
    }
}
```

Сборка началась в 21:07:17...
Сборка завершена в 21:07:17 и заняло 00,739 с



The screenshot shows the `Program.cs` file in the Visual Studio IDE. It contains the `Program` class with a `Main` method that initializes the application and runs the `nForm` instance.

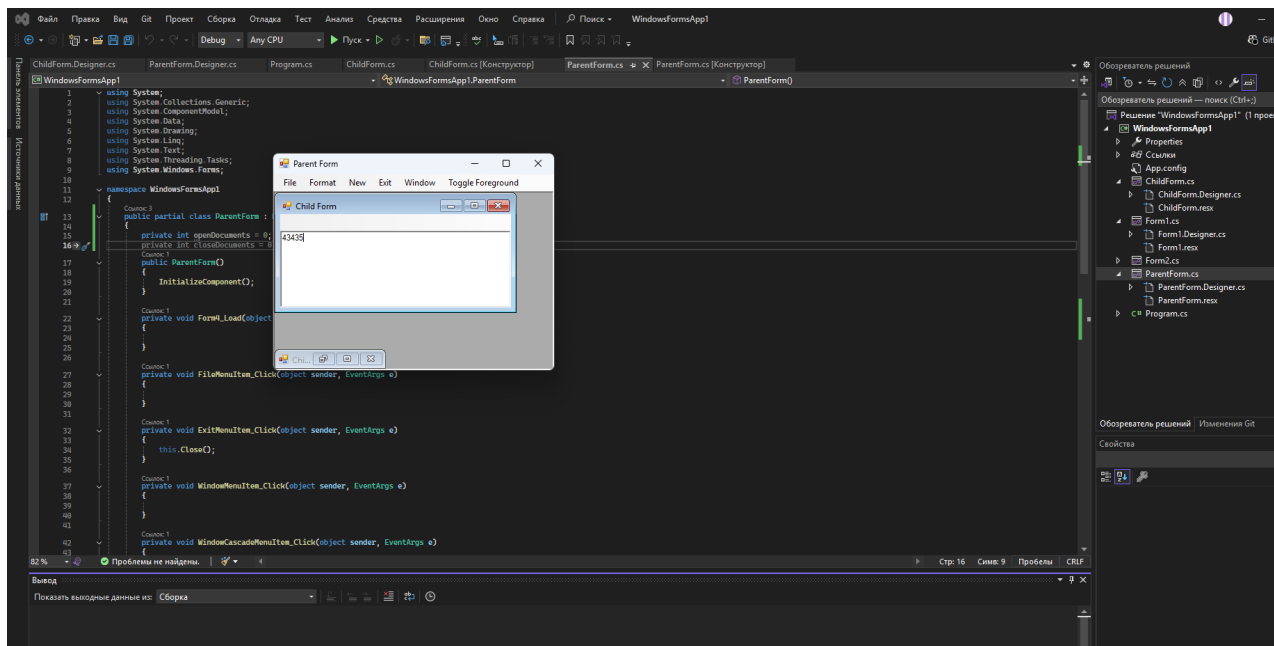
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    internal static class Program
    {
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new nForm()); // Изменили здесь
        }
    }
}
```

Упражнение 4 Создание MDI-приложения

В этом упражнении Вы создадите MDI-приложение с родительской

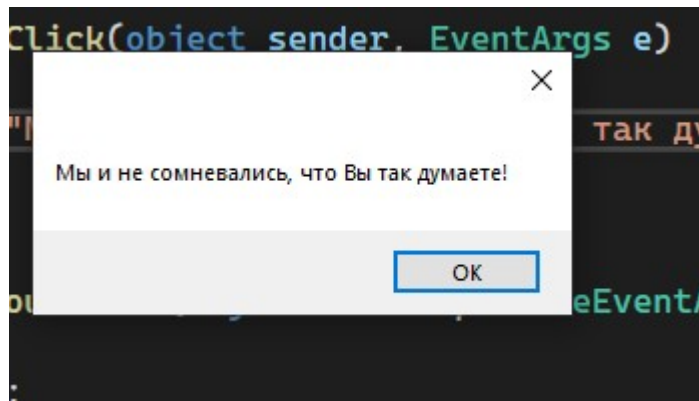
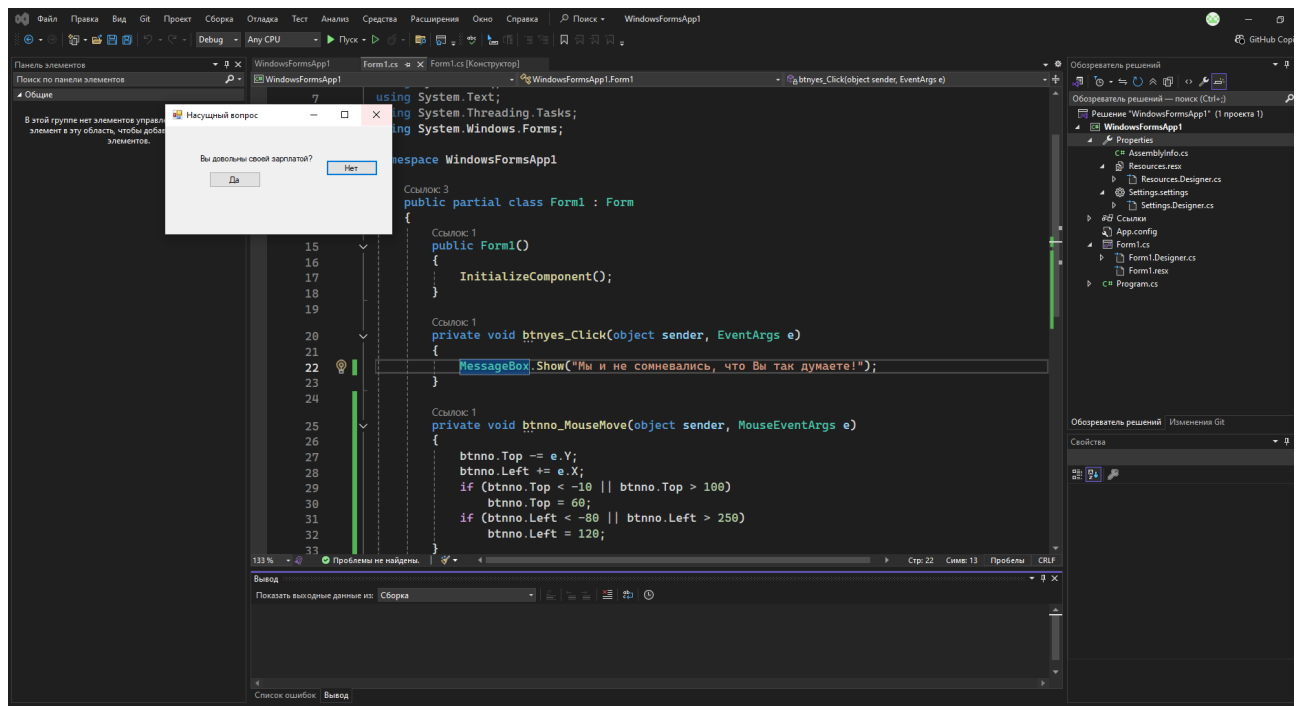
формой, загружающей и организующей дочерние формы. Также Вы познакомитесь с элементом управления MenuStrip, который позволяет создать меню формы.



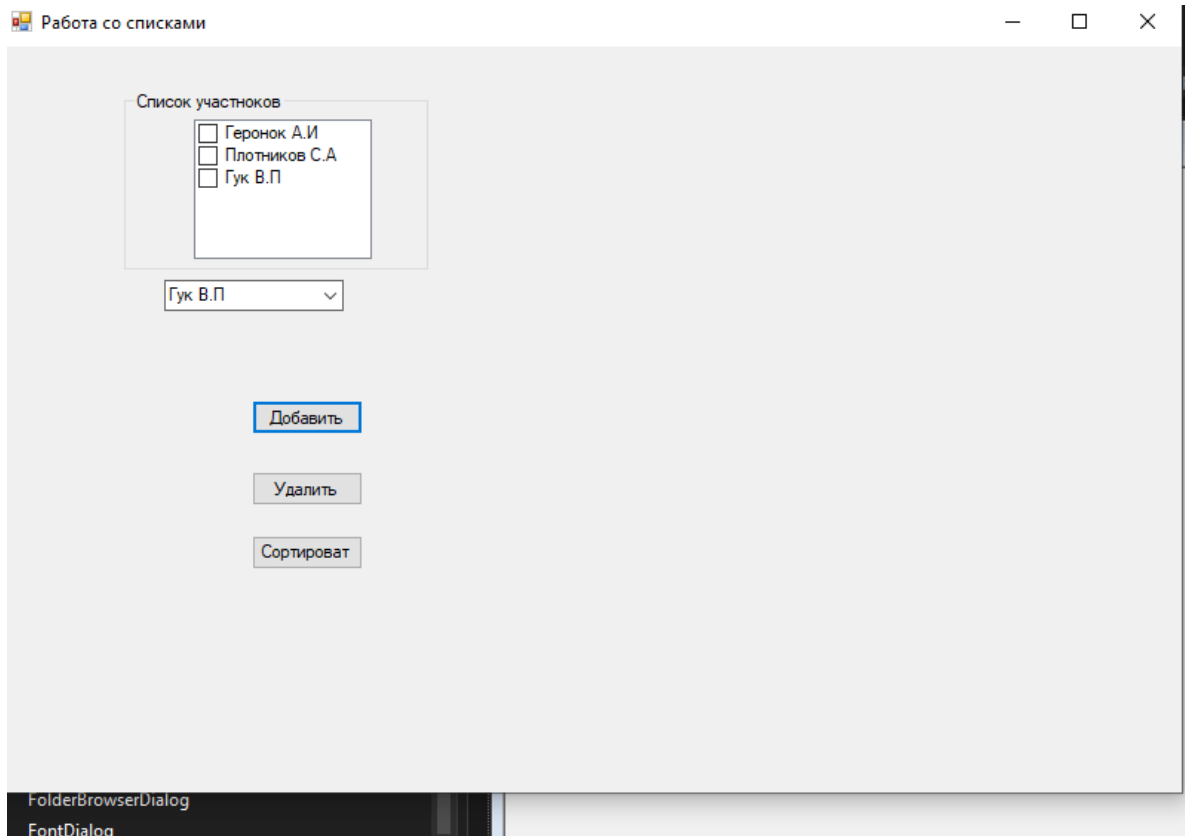
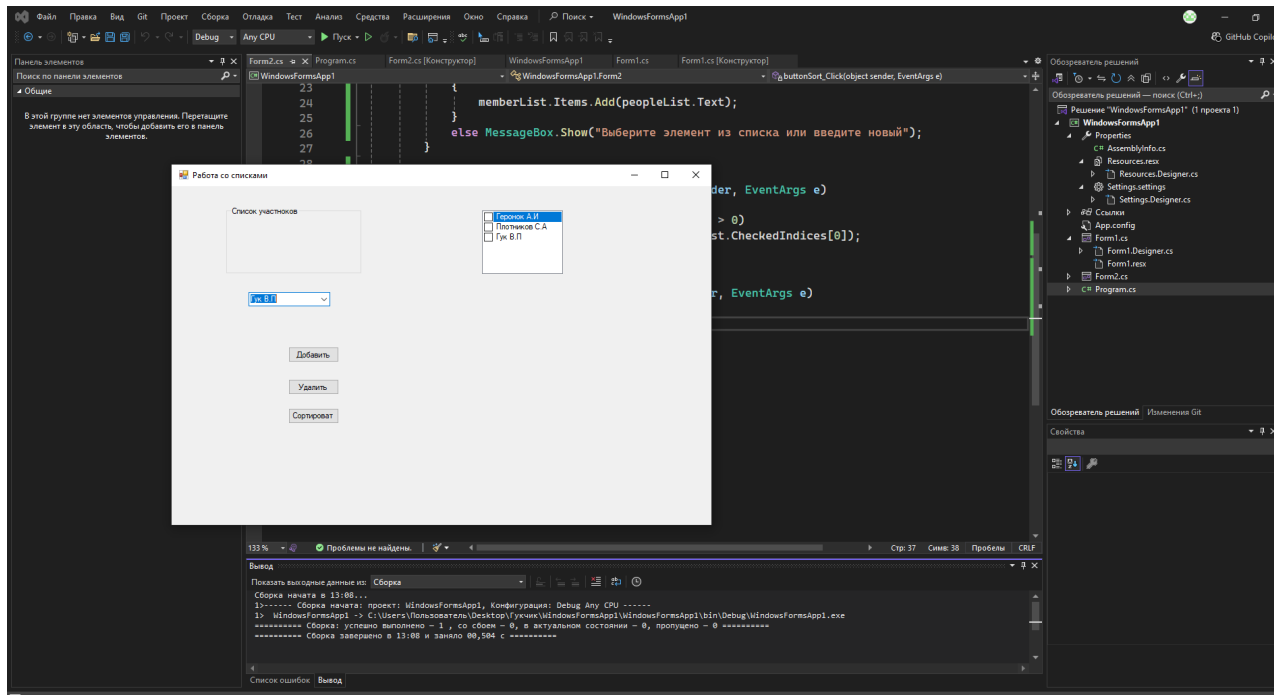
2. Лабораторная работа

Упражнение 1 Обработка событий Click и MouseMove

ф



Упражнение 2 Работа со списками

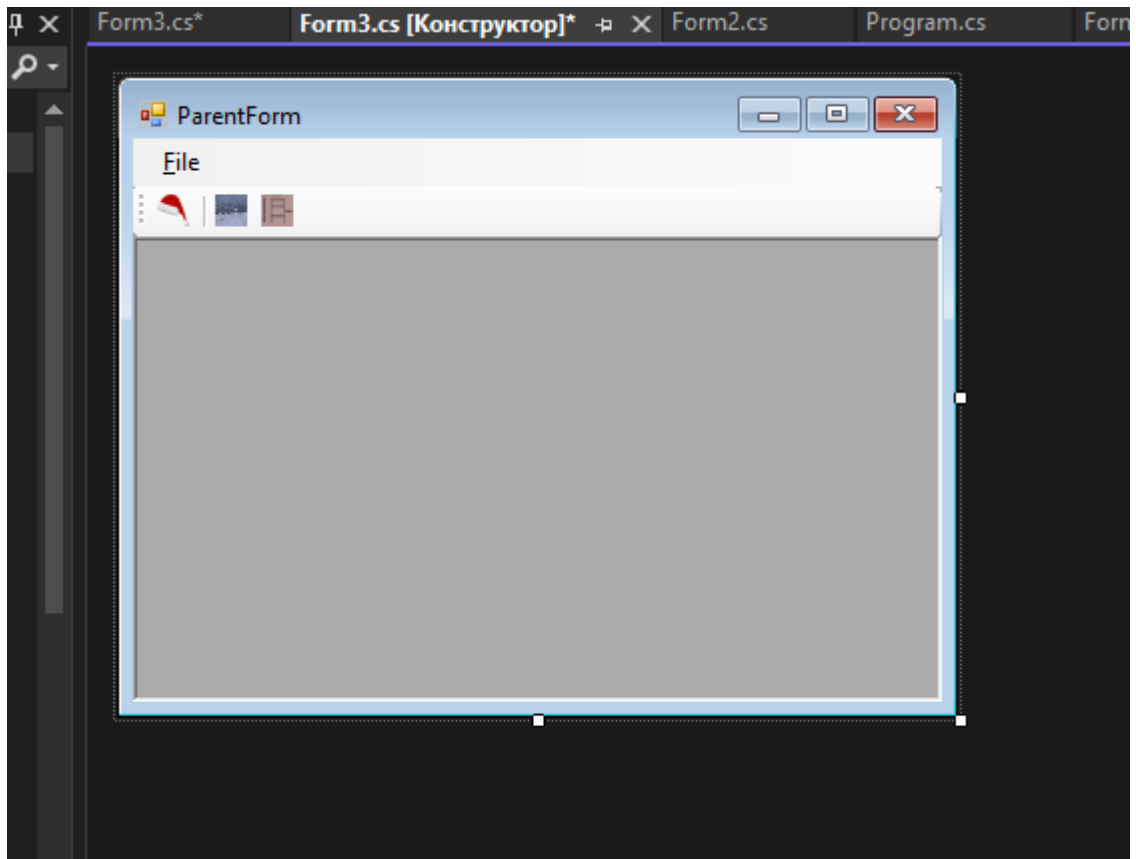


Упражнение 3 Создание и использование элемента управления

ToolStrip

ToolStrip – это элемент управления, разработанный с целью

упрощения создания пользовательских панелей инструментов, которые выглядят и работают, как панели инструментов Microsoft Office и Microsoft Internet Explorer. Используя элемент управления ToolStrip, вы можете быстро разрабатывать легко настраиваемые панели инструментов профессионального вида.



Лабораторная работа 3. Создание элементов управления

Цель работы

Изучение способов разработки элементов управления и получение навыков по их настройке и применению в дальнейшей работе.

Упражнение 1. Создание составного элемента управления

В дополнение к уже существующим элементам управления можно разрабатывать собственные, чтобы обеспечить для своих приложений специализированную функциональность.

Существует три вида разрабатываемых пользователем элементов управления:

- составные (composite), которые создаются при объединении других элементов управления Windows Forms;
- специализированные (custom), создаваемые с нуля и предоставляющие собственный код для прорисовки;
- расширенные (extended), которые добавляют функциональность к уже существующему элементу управления Windows Forms.

Составные элементы управления наследуются от класса UserControl, Он предоставляет базовый уровень функциональности, обеспечивающий добавление других элементов управления, а также свойств, методов и

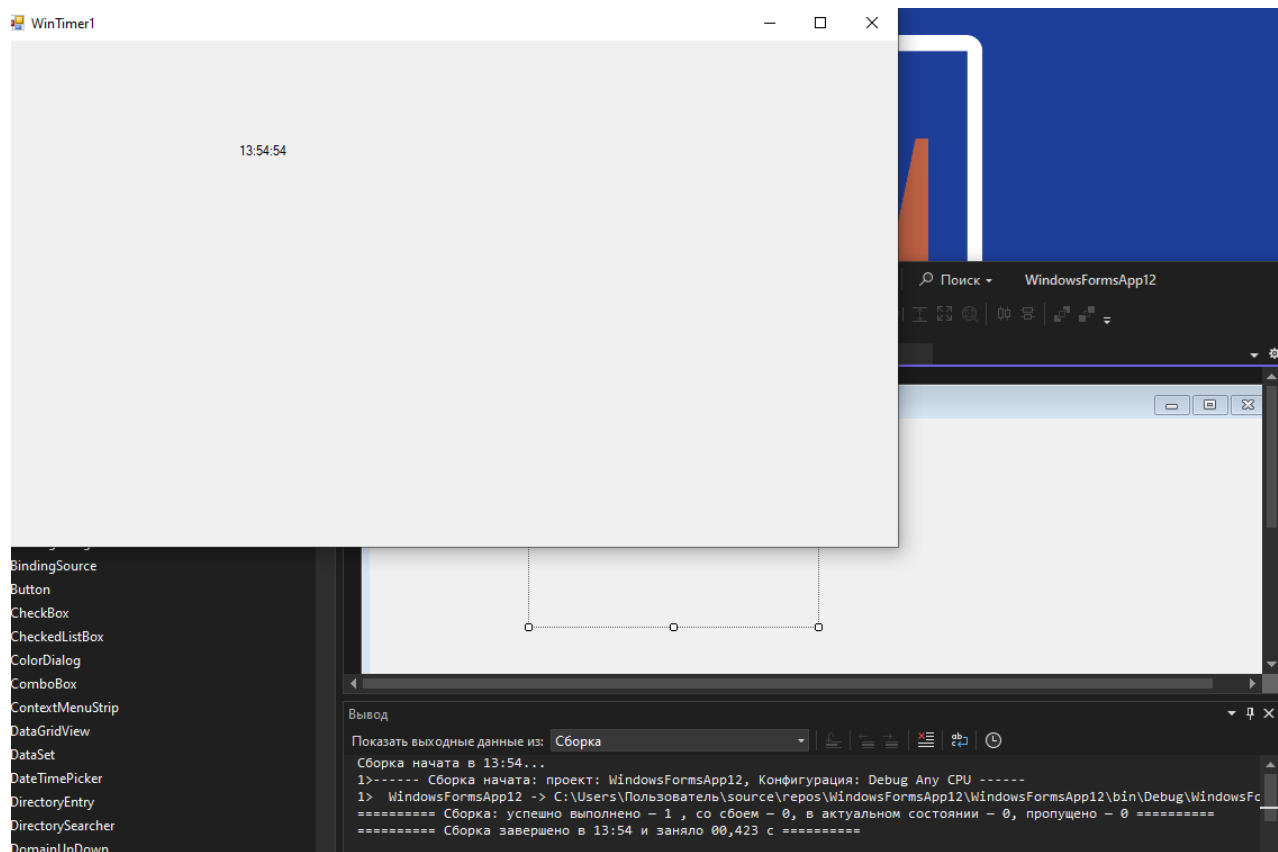
событий. Класс UserControl имеет собственный конструктор, позволяющий использовать в Visual Studio IDE перетаскивание дополнительных

элементов управления из Toolbox на поверхность конструктора и настраивать их.

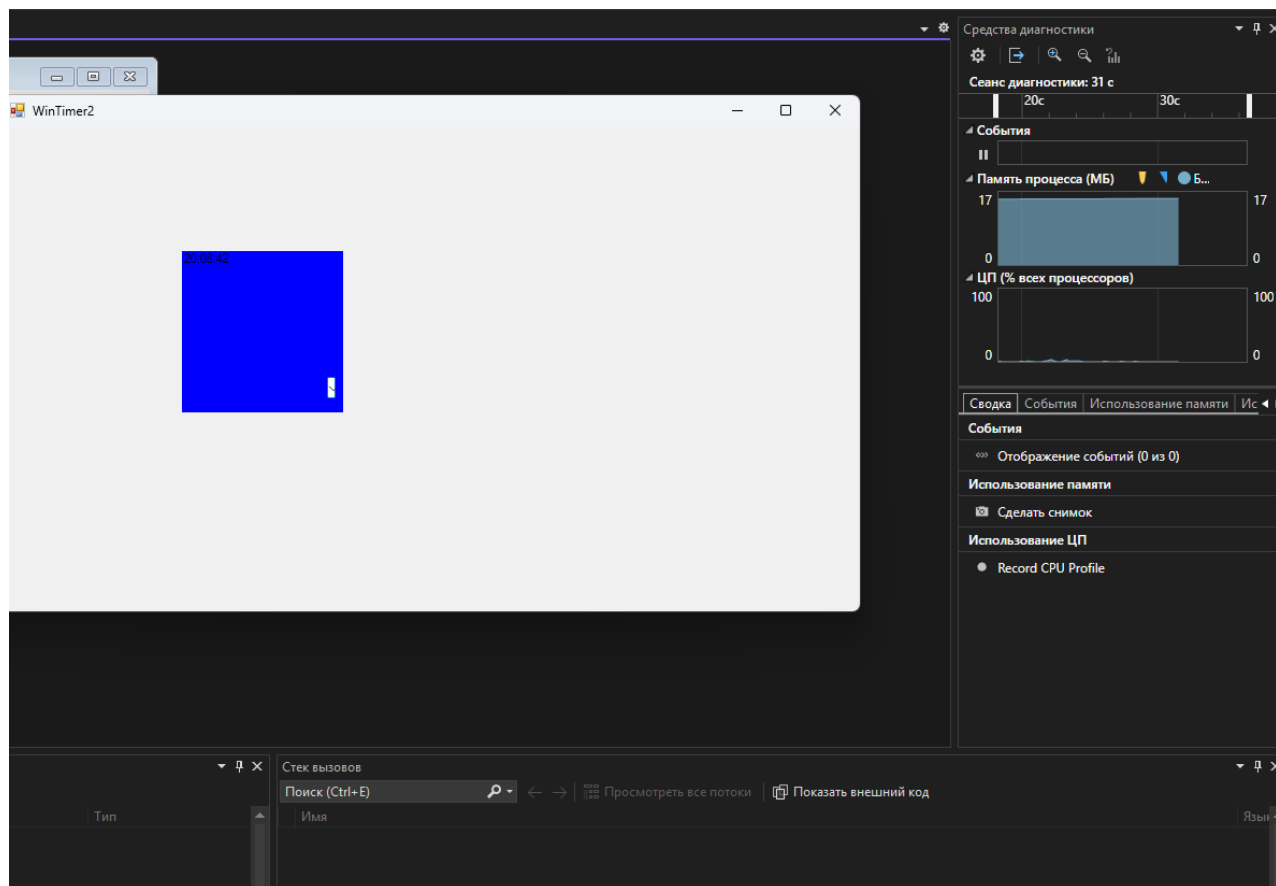
В этом упражнении вы создадите составной элемент управления, действующий как цифровые часы. В него вы добавите элемент управления

Label, отображающий правильное время, и компонент Timer, каждую секунду обновляющий Label. Предоставив свойство Enabled элемента управления Timer, вы дадите пользователям возможность включать и отключать часы.

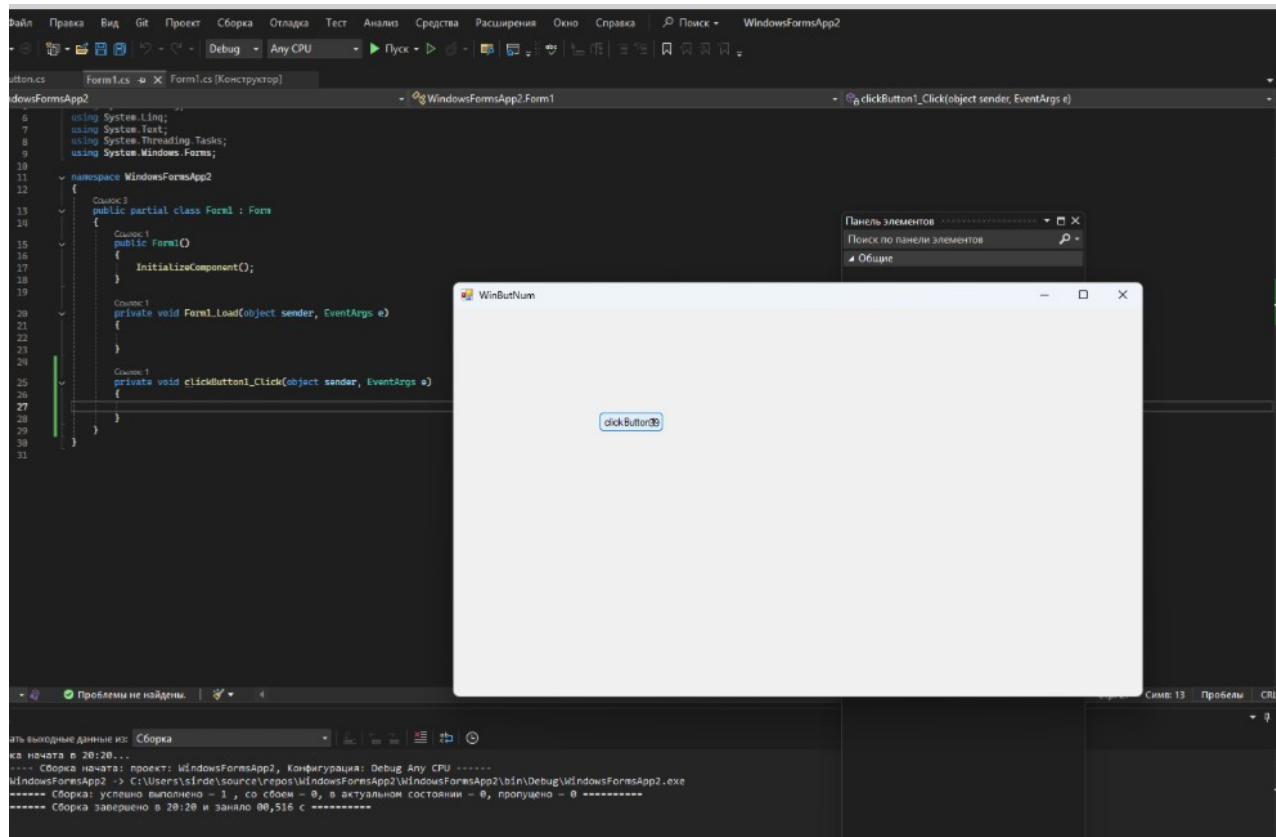
Разработка составного элемента управления



Создание специализированного элемента управления



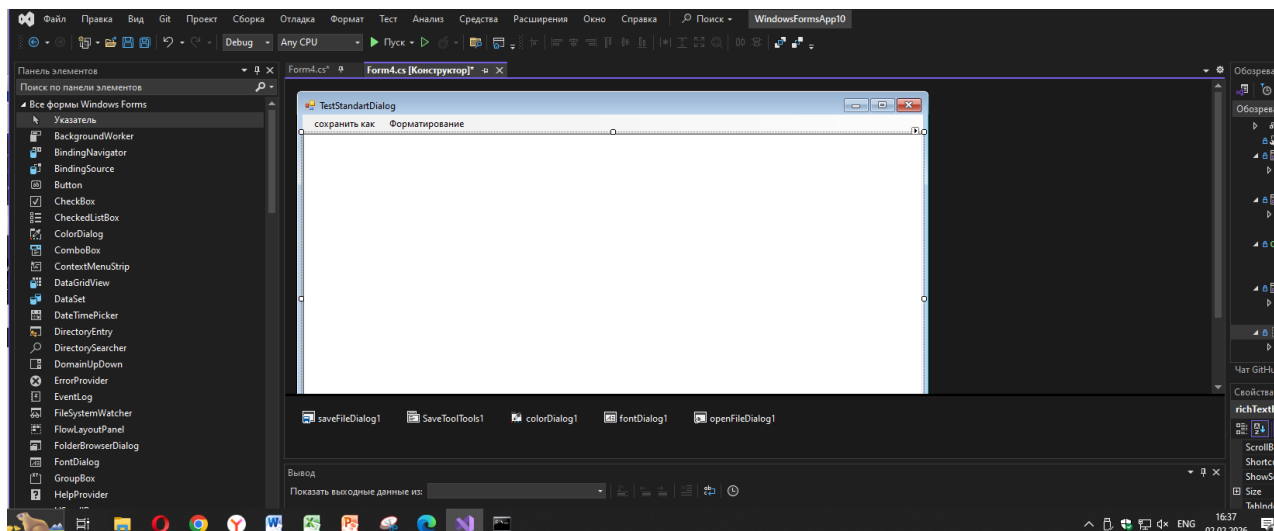
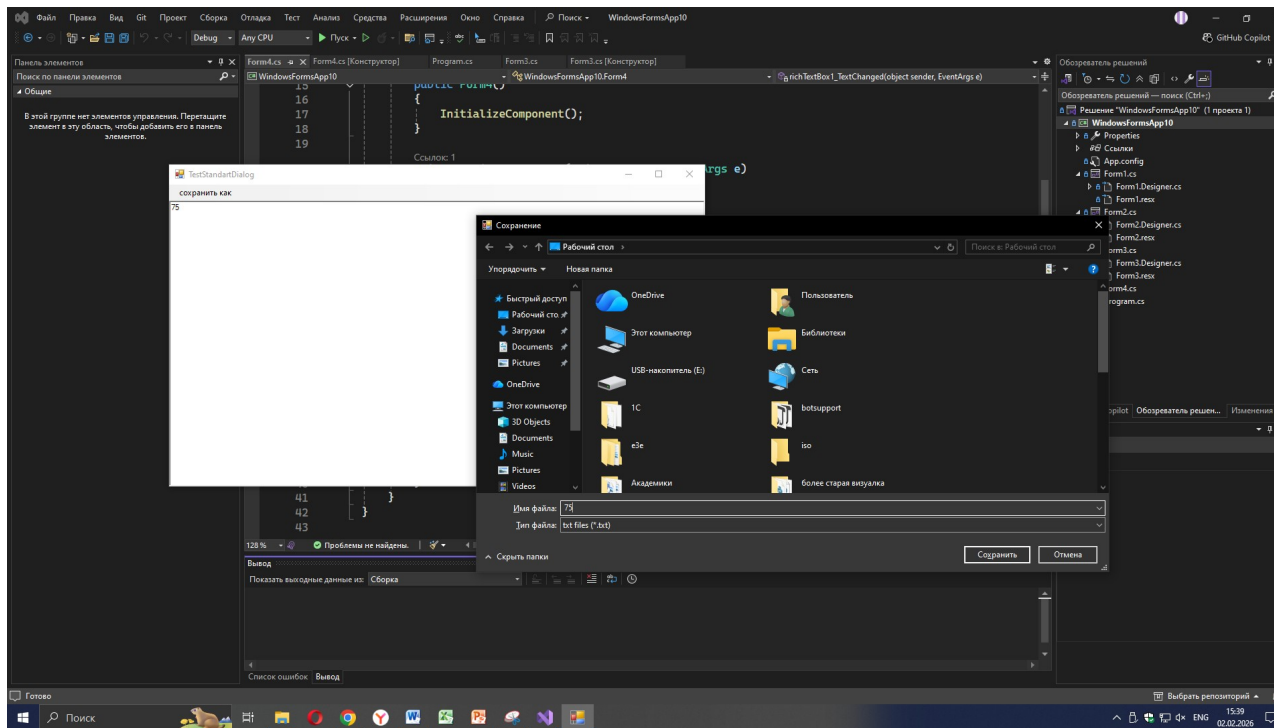
Создание расширенных элементов управления

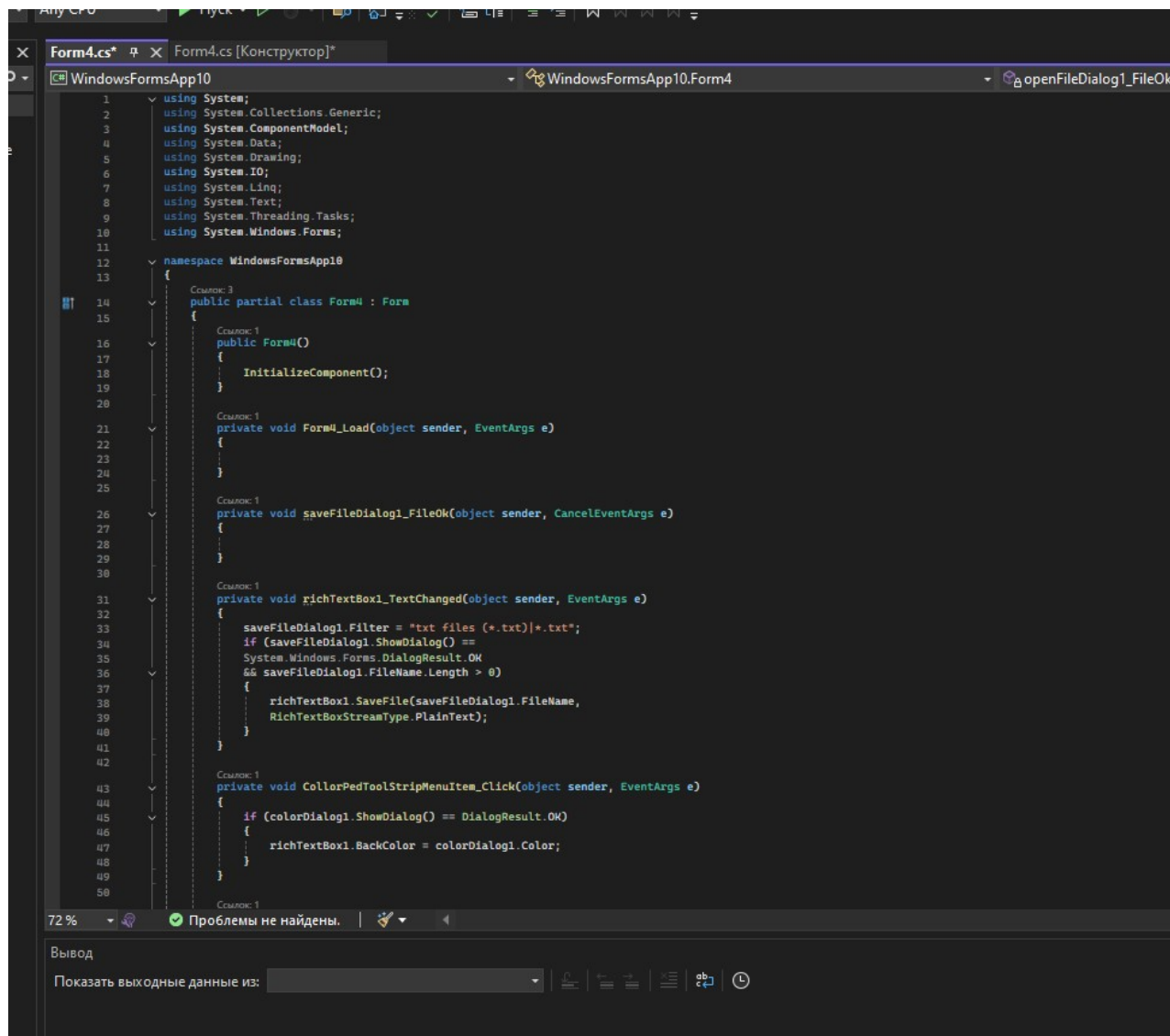


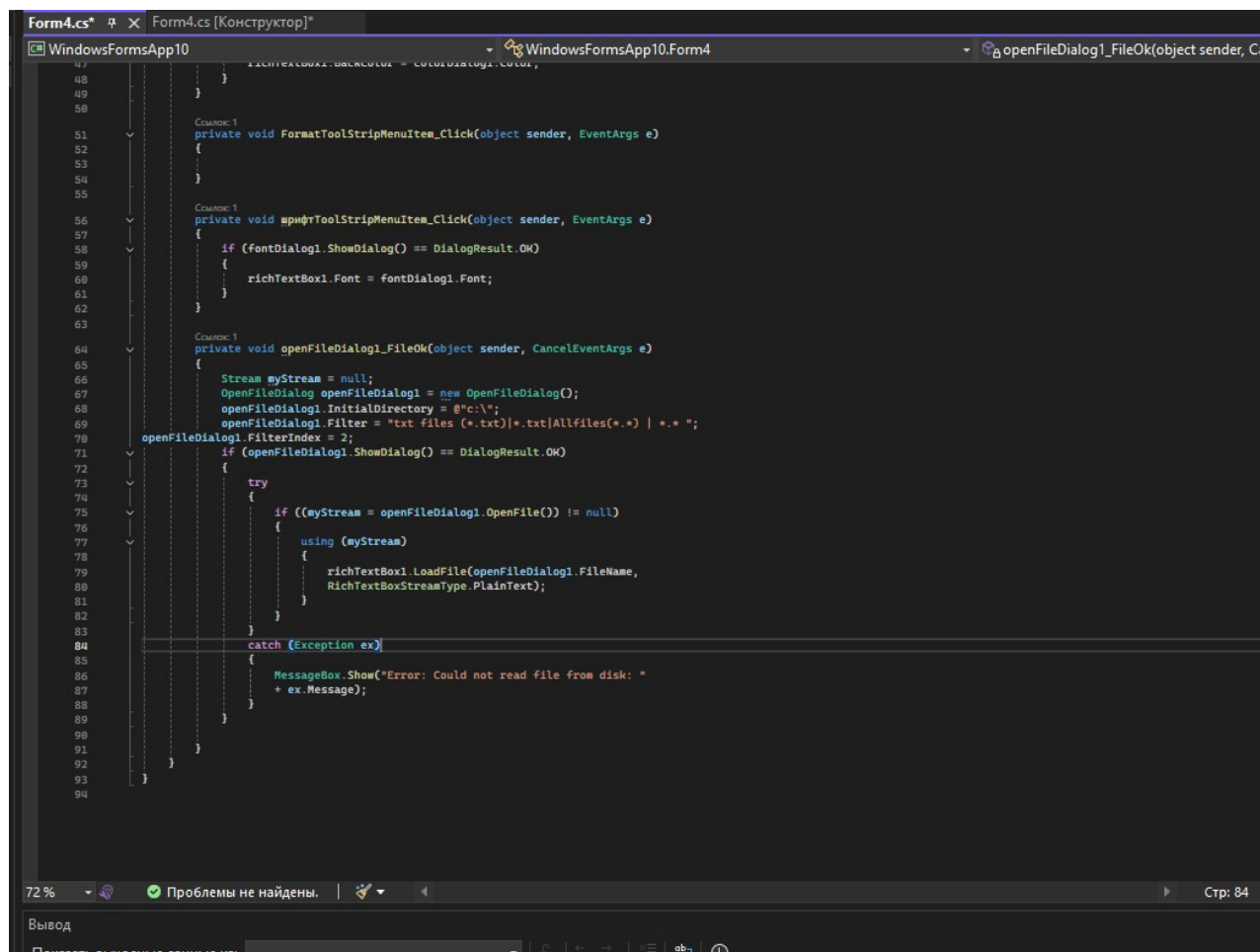
Лабораторная работа 4 Использование окон диалога в формах

Цель работы

Изучение способов использования компонентов, представляющие диалоговые окна и получение навыков по работе с окнами диалога.







```
Form4.cs*  Form4.cs [Конструктор]*
WindowsFormsApp10
Form4
openFileDialog1_FileOk(object sender, C

47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94

    }
    }
    }

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

    private void FormatToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (fontDialog1.ShowDialog() == DialogResult.OK)
        {
            richTextBox1.Font = fontDialog1.Font;
        }
    }

    private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
    {
        Stream myStream = null;
        OpenFileDialog openFileDialog1 = new OpenFileDialog();
        openFileDialog1.InitialDirectory = @"c:\\";
        openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.* ";
        openFileDialog1.FilterIndex = 2;
        if (openFileDialog1.ShowDialog() == DialogResult.OK)
        {
            try
            {
                if ((myStream = openFileDialog1.OpenFile()) != null)
                {
                    using (myStream)
                    {
                        richTextBox1.LoadFile(openFileDialog1.FileName,
                            RichTextBoxStreamType.PlainText);
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error: Could not read file from disk: "
                    + ex.Message);
            }
        }
    }
}
```

72 % 72 % Проблемы не найдены. Стр: 84

Вывод

Показать выходные данные из:

Лабораторная работа 6 Организация печати в формах

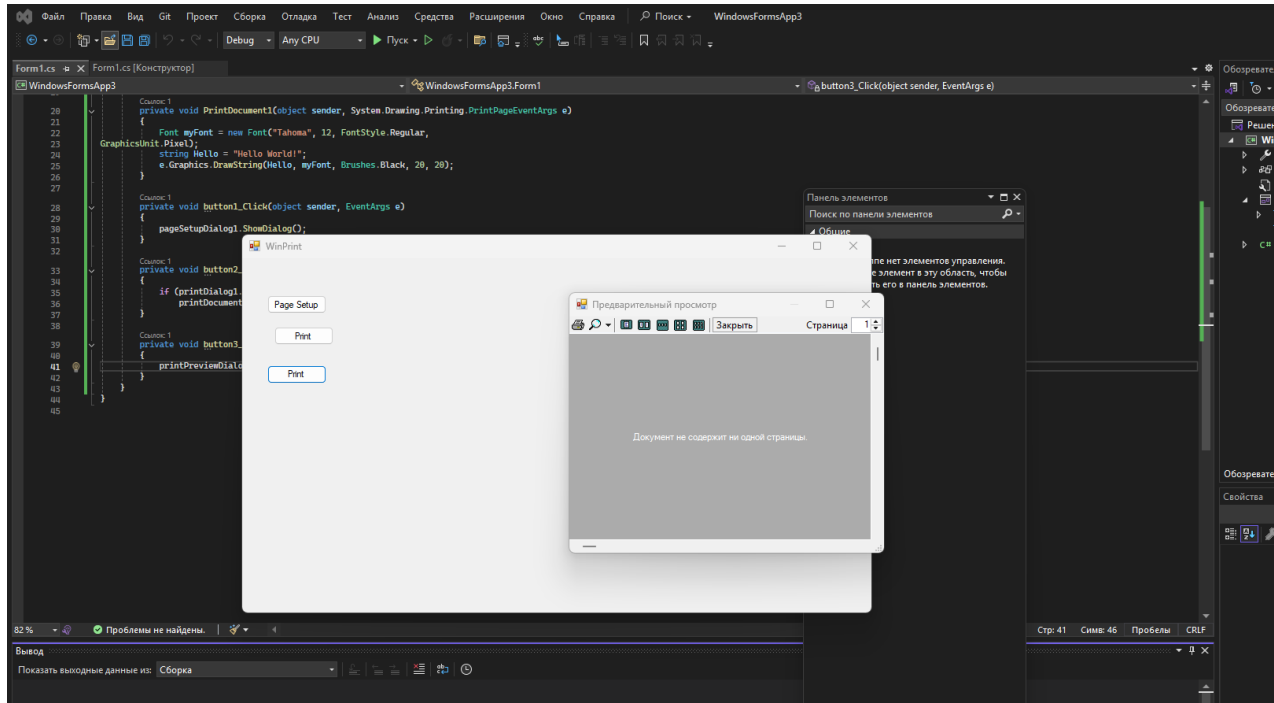
windows

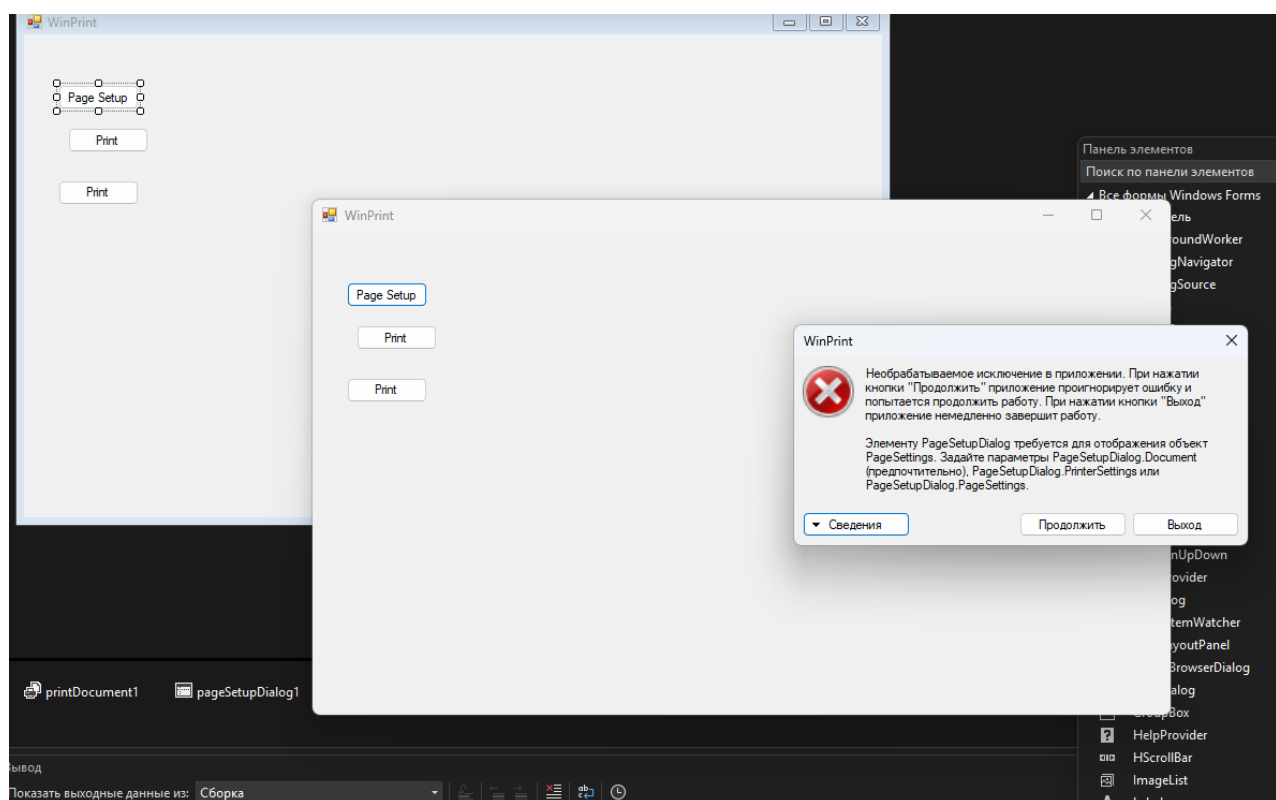
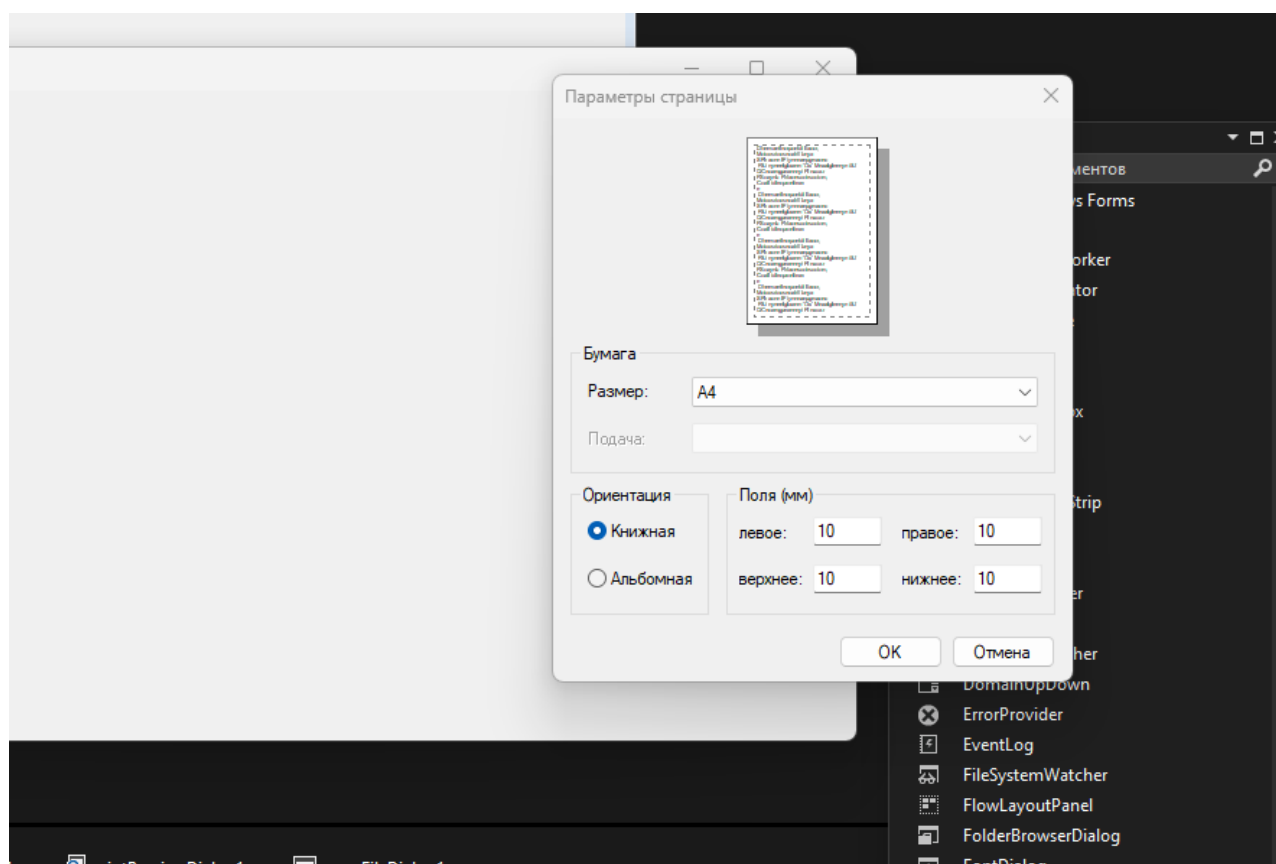
Цель работы

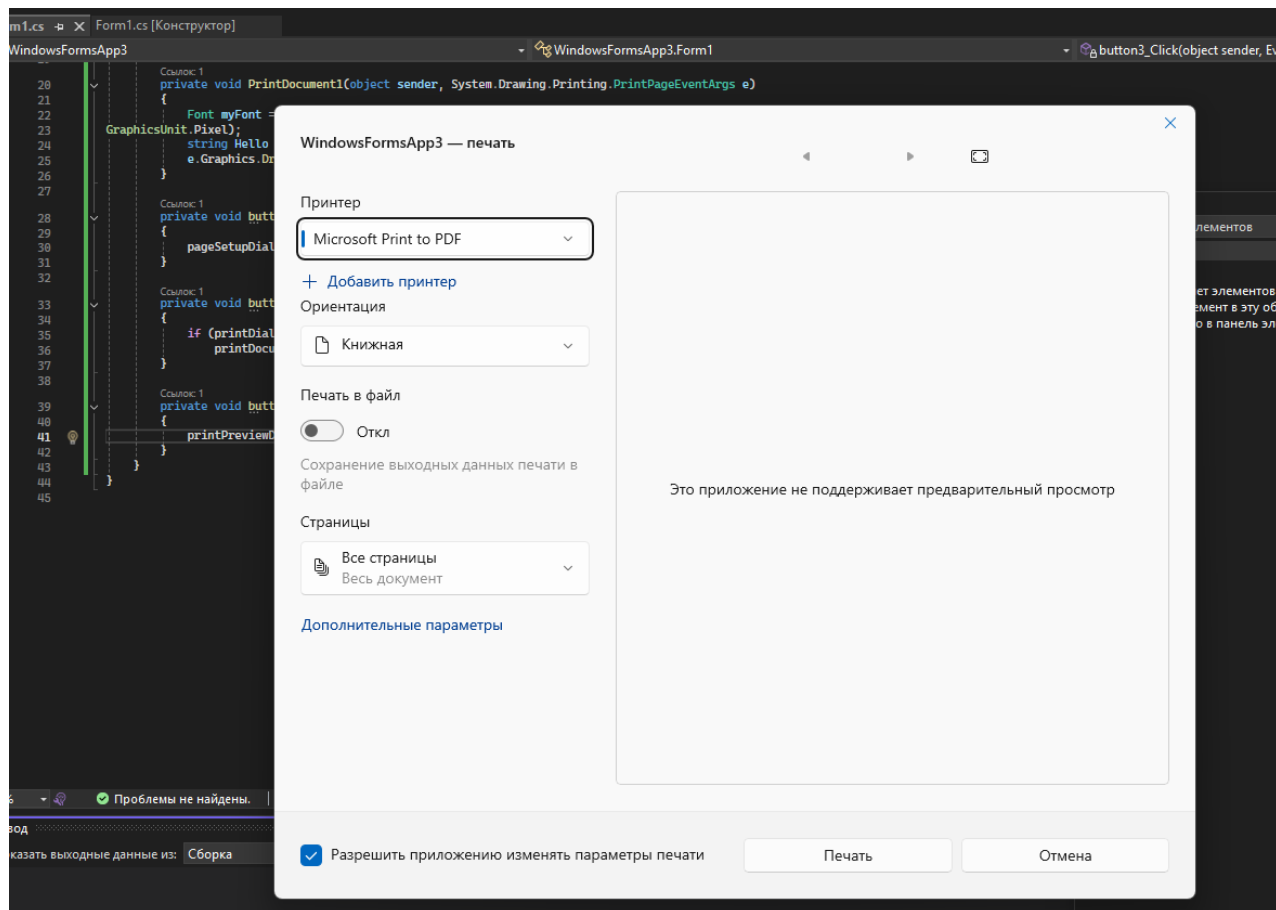
Изучение классов, реализующих задачу программирования печати и

получение навыков по работе в программе с диалоговыми окнами.

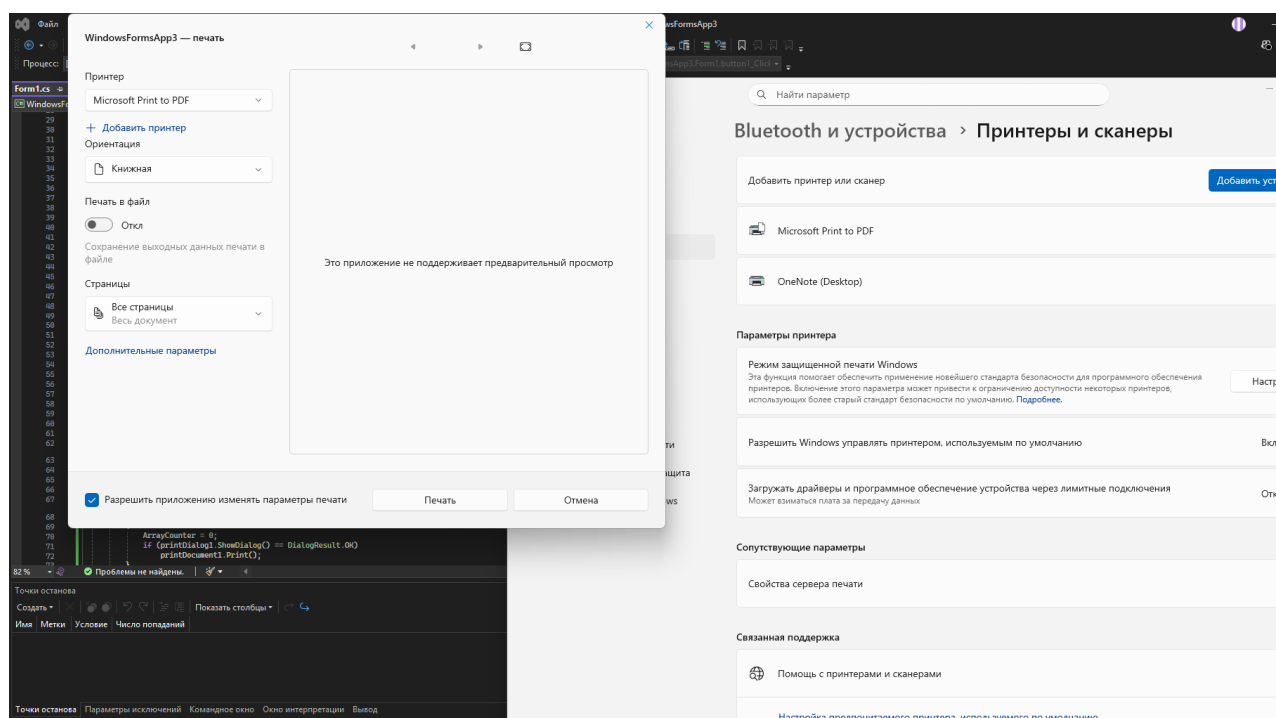
Упражнение 1 Использование диалоговых окон для печати







Упражнение 2 Создание документа печати




```

private int ArrayCounter = 0;

class Form1
{
    public Form1()
    {
        InitializeComponent();
    }
}

class Program
{
    private void PrintDocument1(object sender, System.Drawing.Printing.PrintPageEventArgs e)
    {
        if (strings != null && ArrayCounter < strings.Length)
        {
            float LeftMargin = e.MarginBounds.Left;
            float TopMargin = e.MarginBounds.Top;
            float MyLines = 0;
            float YPosition = 0;
            int Counter = 0;
            string CurrentLine;

            MyLines = e.MarginBounds.Height / (this.Font.GetHeight(e.Graphics));

            while (Counter < MyLines && ArrayCounter <= strings.Length - 1)
            {
                CurrentLine = strings[ArrayCounter];
                YPosition = TopMargin + Counter * this.Font.GetHeight(e.Graphics);
                e.Graphics.DrawString(CurrentLine, this.Font, Brushes.Black, LeftMargin, YPosition, new StringFormat());
                Counter++;
                ArrayCounter++;
            }

            if (ArrayCounter < strings.Length)
            {
                e.HasMorePages = true;
            }
            else
            {
                e.HasMorePages = false;
                ArrayCounter = 0;
            }
        }
        else
        {
            Font myFont = new Font("Tahoma", 12, FontStyle.Regular, GraphicsUnit.Pixel);
            string Hello = "Hello World!";
            e.Graphics.DrawString(Hello, myFont, Brushes.Black, 20, 20);
        }
    }

    class Form1
    {
        private void button1_Click(object sender, EventArgs e)
        {
            pageSetupDialog1.ShowDialog();
        }

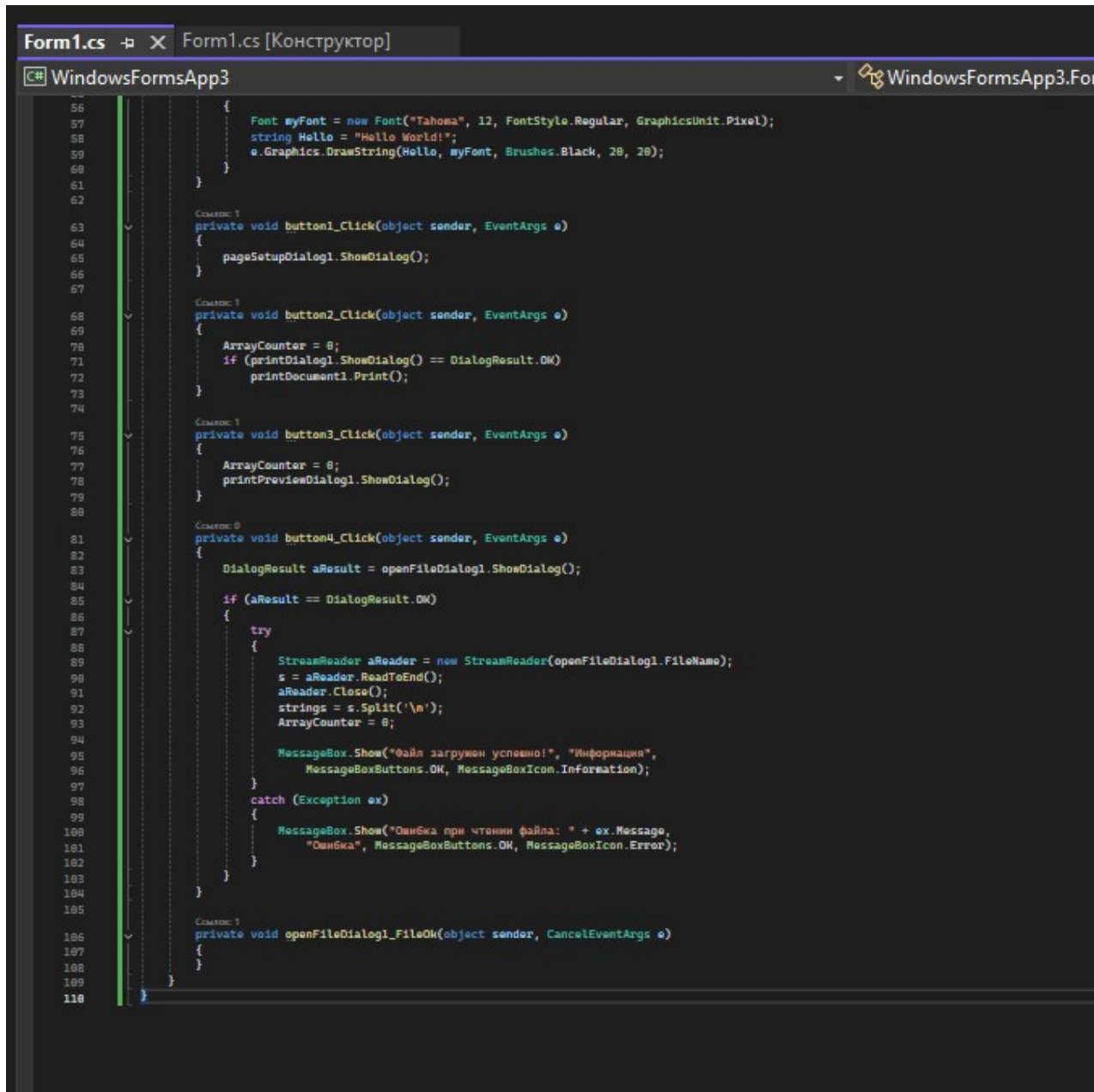
        class Form1
        {
            private void button2_Click(object sender, EventArgs e)
            {
                ArrayCounter = 0;
                if (printDialog1.ShowDialog() == DialogResult.OK)
                {
                    printDocument1.Print();
                }
            }

            class Form1
            {
                private void button3_Click(object sender, EventArgs e)
            }
        }
    }
}

```

Проблема не найдена.

Показать столбцы



```
Form1.cs Form1.cs [Конструктор]
C# WindowsFormsApp3
56 {
57     Font myFont = new Font("Tahoma", 12, FontStyle.Regular, GraphicsUnit.Pixel);
58     string Hello = "Hello World!";
59     e.Graphics.DrawString(Hello, myFont, Brushes.Black, 20, 20);
60 }
61
62
63 // Событие: 1
64 private void button1_Click(object sender, EventArgs e)
65 {
66     pageSetupDialog1.ShowDialog();
67 }
68
69 // Событие: 1
70 private void button2_Click(object sender, EventArgs e)
71 {
72     ArrayCounter = 0;
73     if (printDialog1.ShowDialog() == DialogResult.OK)
74         printDocument1.Print();
75 }
76
77 // Событие: 1
78 private void button3_Click(object sender, EventArgs e)
79 {
80     ArrayCounter = 0;
81     printPreviewDialog1.ShowDialog();
82 }
83
84 // Событие: 0
85 private void button4_Click(object sender, EventArgs e)
86 {
87     DialogResult aResult = openFileDialog1.ShowDialog();
88
89     if (aResult == DialogResult.OK)
90     {
91         try
92         {
93             StreamReader aReader = new StreamReader(openFileDialog1.FileName);
94             s = aReader.ReadToEnd();
95             aReader.Close();
96             strings = s.Split('\n');
97             ArrayCounter = 0;
98
99             MessageBox.Show("Файл загружен успешно!", "Информация",
100                             MessageBoxButtons.OK, MessageBoxIcon.Information);
101         }
102         catch (Exception ex)
103         {
104             MessageBox.Show("Ошибка при чтении файла: " + ex.Message,
105                             "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
106         }
107     }
108 }
109
110 // Событие: 1
111 private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
112 {
113 }
```

Упражнение 3

Создание специализированной формы предварительного

