



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

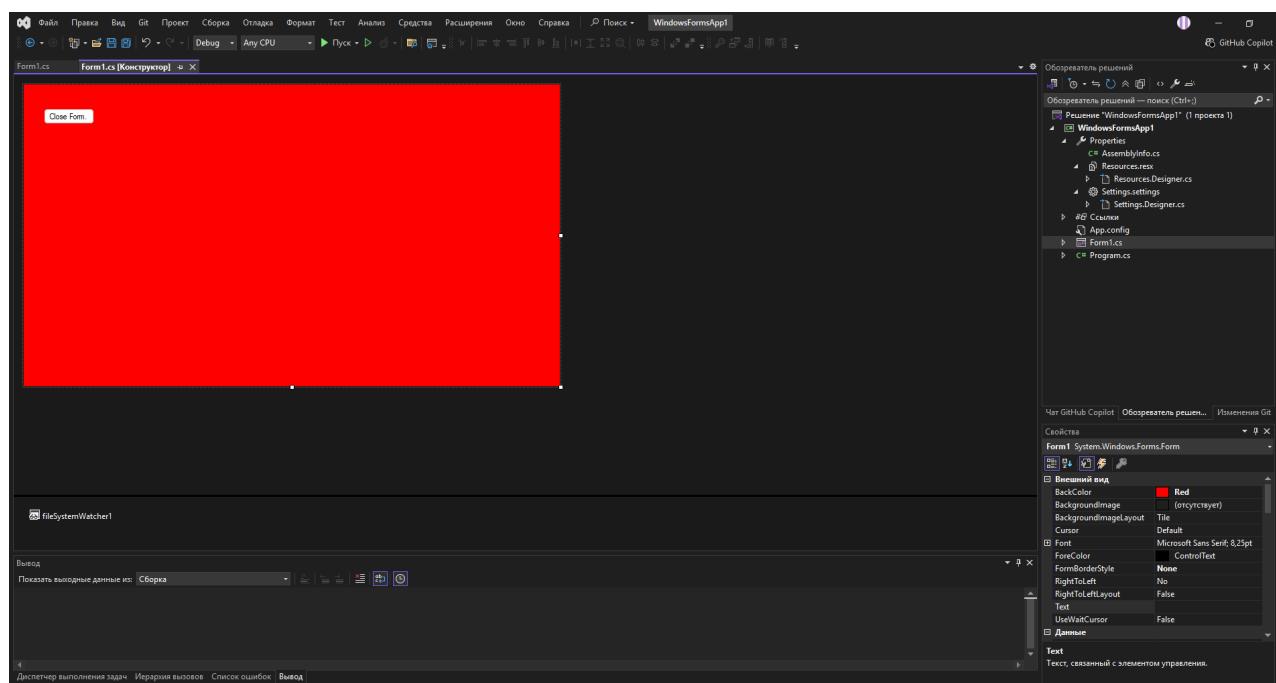
с тремя углами):

```
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 )});
```

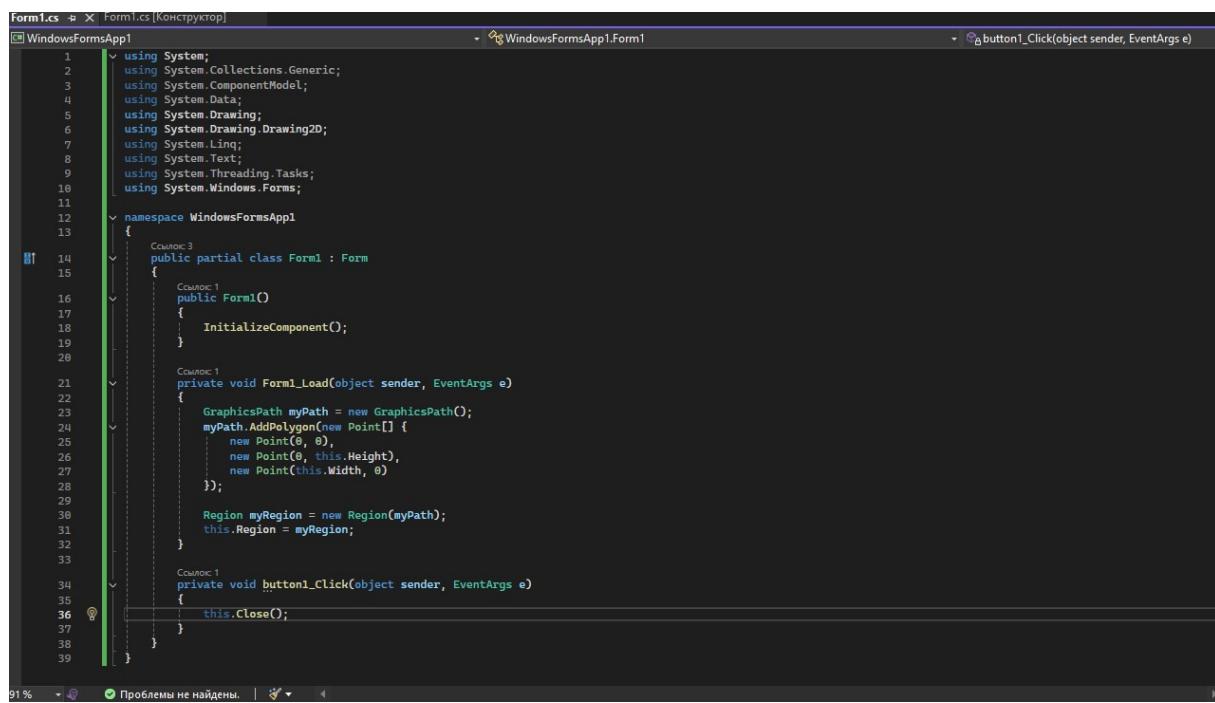
```
Region myRegion = new Region(myPath);
```

```
this.Region = myRegion;
```

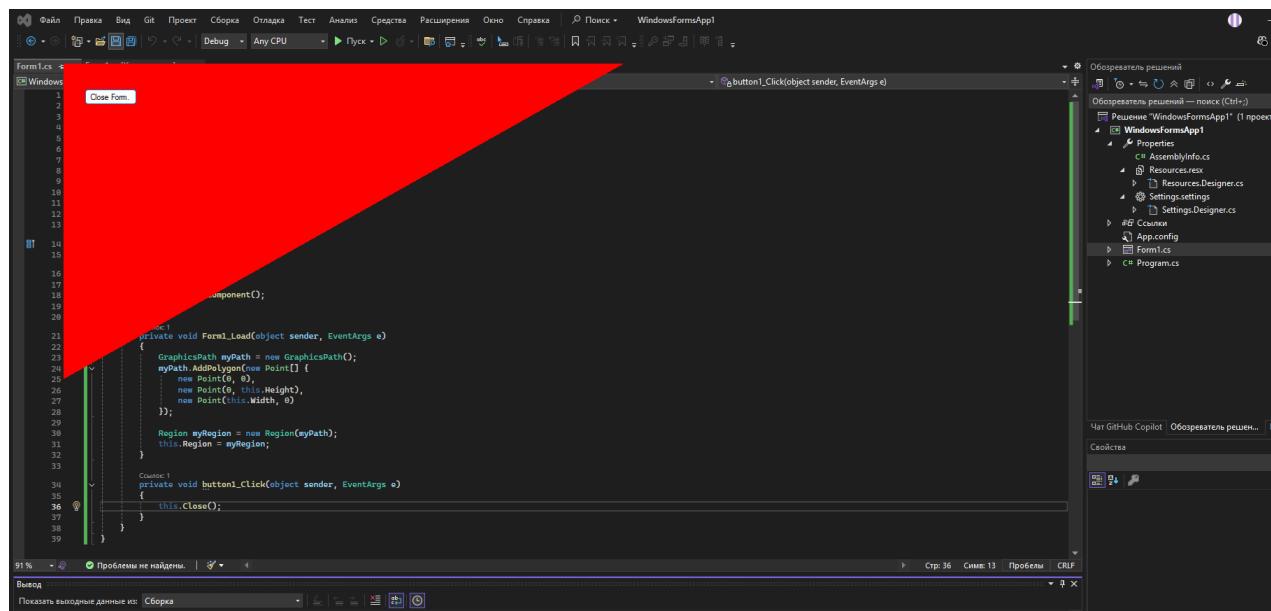
6 Постройте и запустите приложение. Появится треугольная форма.



Отчет по Разработка и тестирование кода



```
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             {
26                 new Point(0, 0),
27                 new Point(0, this.Height),
28                 new Point(this.Width, 0)
29             });
30
31             Region myRegion = new Region(myPath);
32             this.Region = myRegion;
33         }
34
35         private void button1_Click(object sender, EventArgs e)
36         {
37             this.Close();
38         }
39     }
40 }
```



```
1  Close Form.
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21         private void Form1_Load(object sender, EventArgs e)
22         {
23             GraphicsPath myPath = new GraphicsPath();
24             myPath.AddPolygon(new Point[]
25             {
26                 new Point(0, 0),
27                 new Point(0, this.Height),
28                 new Point(this.Width, 0)
29             });
30
31             Region myRegion = new Region(myPath);
32             this.Region = myRegion;
33         }
34
35         private void button1_Click(object sender, EventArgs e)
36         {
37             this.Close();
38         }
39     }
40 }
```

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

OK.

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:

и. 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:

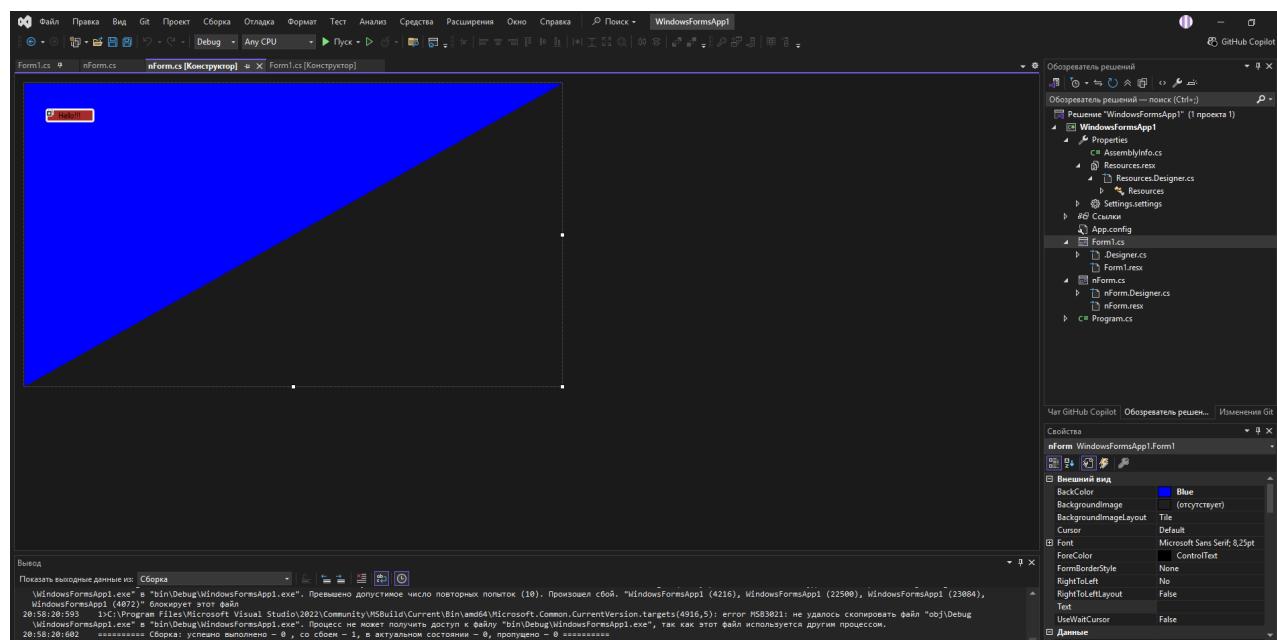
```
Application.Run(new nForm());
```

11 Постройте и запустите приложение. Должна открыться

производная форма со своими свойствами.

Проверьте, наследуется ли

закрытие формы кнопкой.



Отчет по Разработка и тестирование кода

The screenshot shows the Microsoft Visual Studio interface. In the center, the code editor displays `nForm.cs` with the following content:

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Text;
7  using System.Windows.Forms;
8
9  namespace WindowsFormsApp1
10 {
11     public partial class nForm : Form
12     {
13         public nForm()
14         {
15             InitializeComponent();
16             button1.Text = "Hello!!!";
17             button1.BackColor = Color.Brown;
18             this.BackColor = Color.Blue;
19         }
20
21         private void nForm_Load(object sender, EventArgs e)
22         {
23         }
24     }
25 }
```

The Solution Explorer on the right shows the project structure for "WindowsFormsApp1":

- Properties
- Resources.resx
- Settings.settings
- Form1.cs
- Form1.Designer.cs
- nForm.cs
- nForm.Designer.cs
- Program.cs

At the bottom, the Output window shows build logs:

```
21:07:17.042 Помогли в сборке данные из: Сборка
21:07:17.269 1>----- Сборка начата: проект: WindowsFormsApp1, Конфигурация: Debug Any CPU -----
21:07:17.698 1> WindowsFormsApp1 -> C:\Users\slrde\source\repos\WindowsFormsApp1\WindowsFormsApp1\bin\Debug\WindowsFormsApp1.exe
21:07:17.714 ----- Сборка: успешно выполнено - 1 , со ошибкой - 0, в актуальном состоянии - 0, пропущено - 0 -----
21:07:17.714 ===== Сборка завершена в 21:07 и заняло 00,739 с =====
```

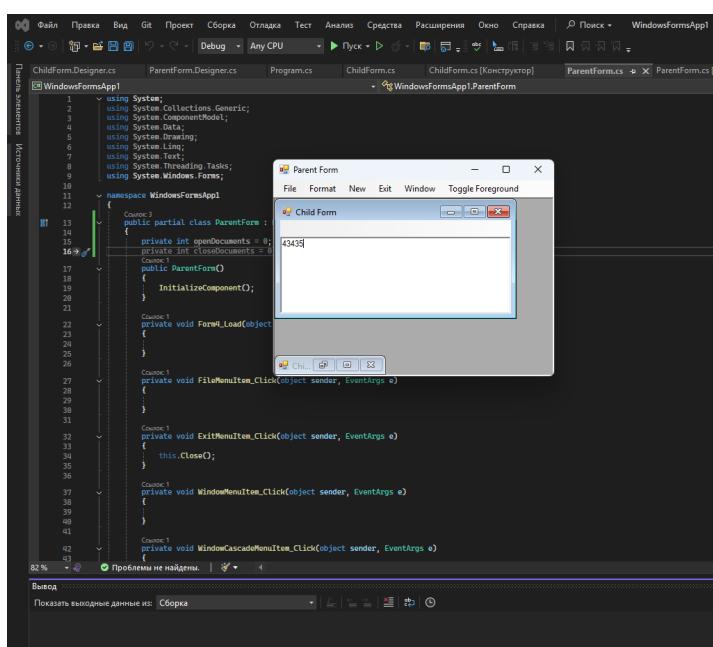
The screenshot shows the Microsoft Visual Studio interface. In the center, the code editor displays `Program.cs` with the following content:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using System.Windows.Forms;
6
7  namespace WindowsFormsApp1
8  {
8      Ссылки 0
9      internal static class Program
10     {
11         [STAThread]
12         Ссылки 0
13         static void Main()
14         {
15             Application.EnableVisualStyles();
16             Application.SetCompatibleTextRenderingDefault(false);
17             Application.Run(new nForm()); // Изменили здесь
18         }
19     }
20 }
```

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.

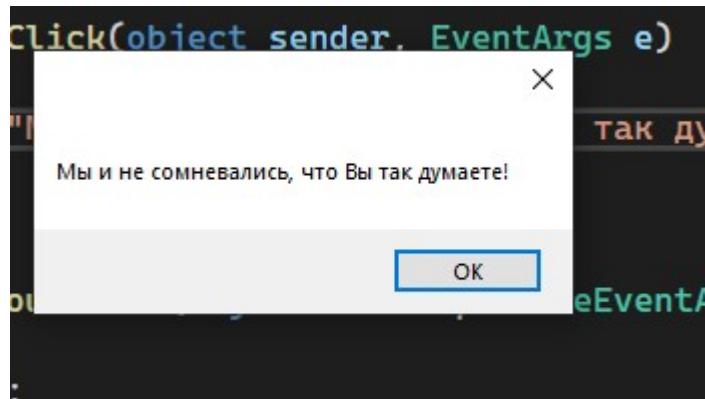
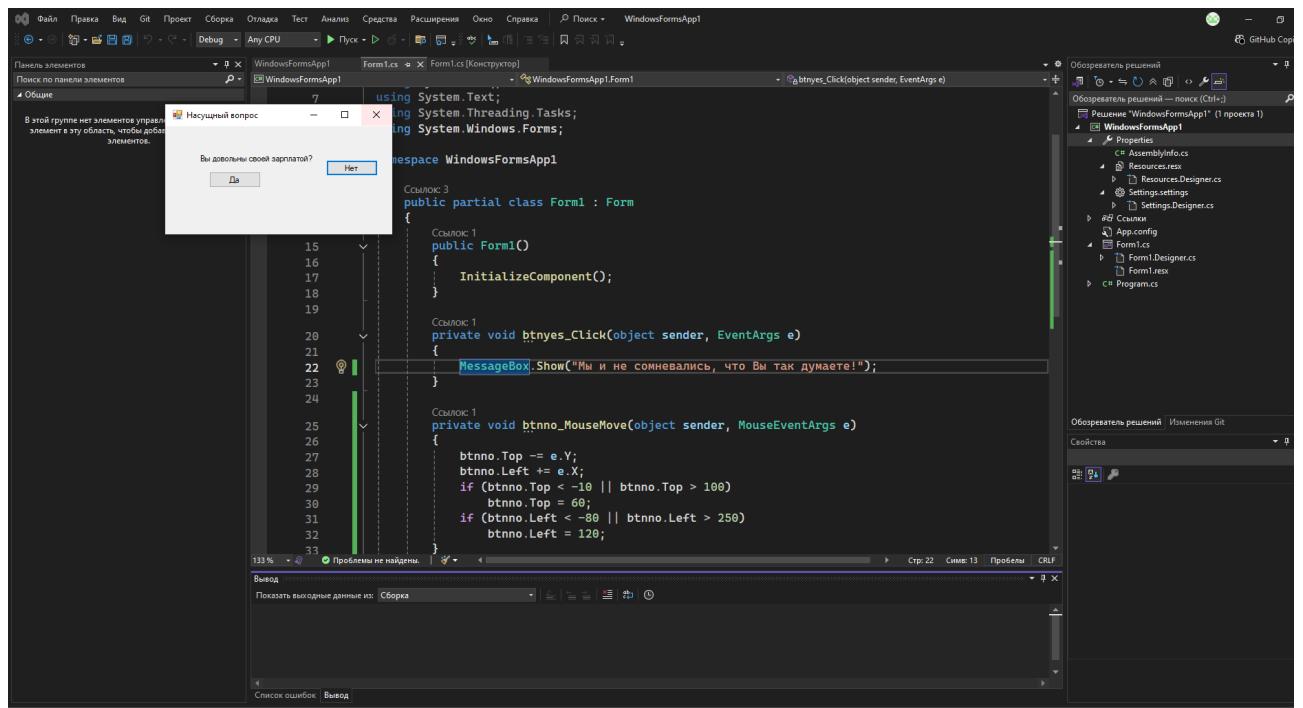


2. Лабораторная работа

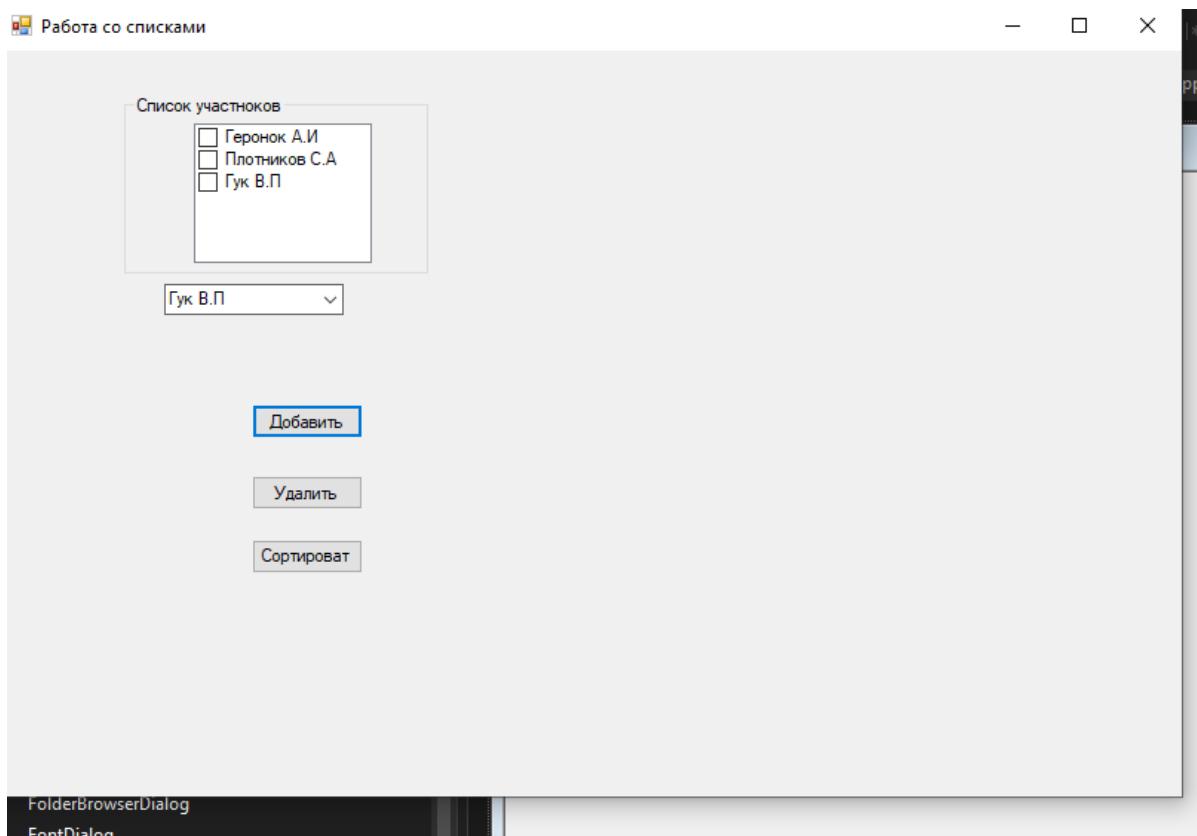
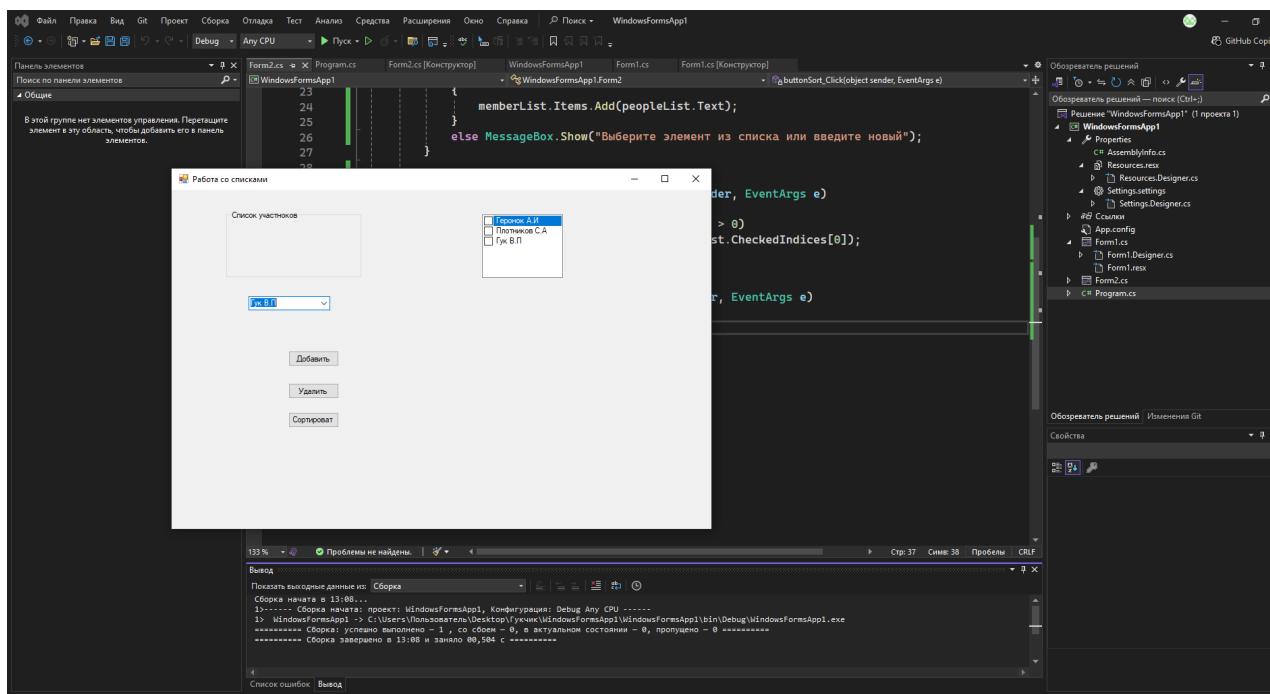
Упражнение 1 Обработка событий Click и MouseMove

ϕ

Отчет по Разработка и тестирование кода



Упражнение 2 Работа со списками

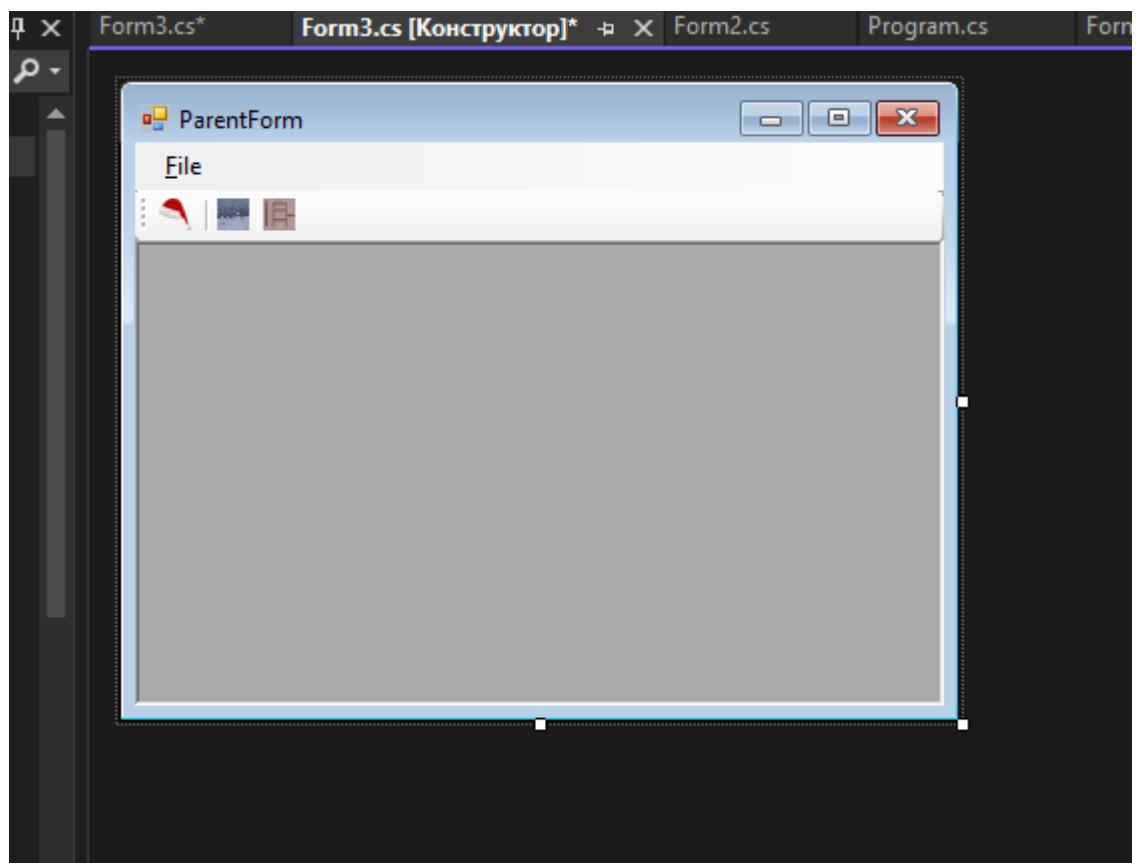


Упражнение 3 Создание и использование элемента управления

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

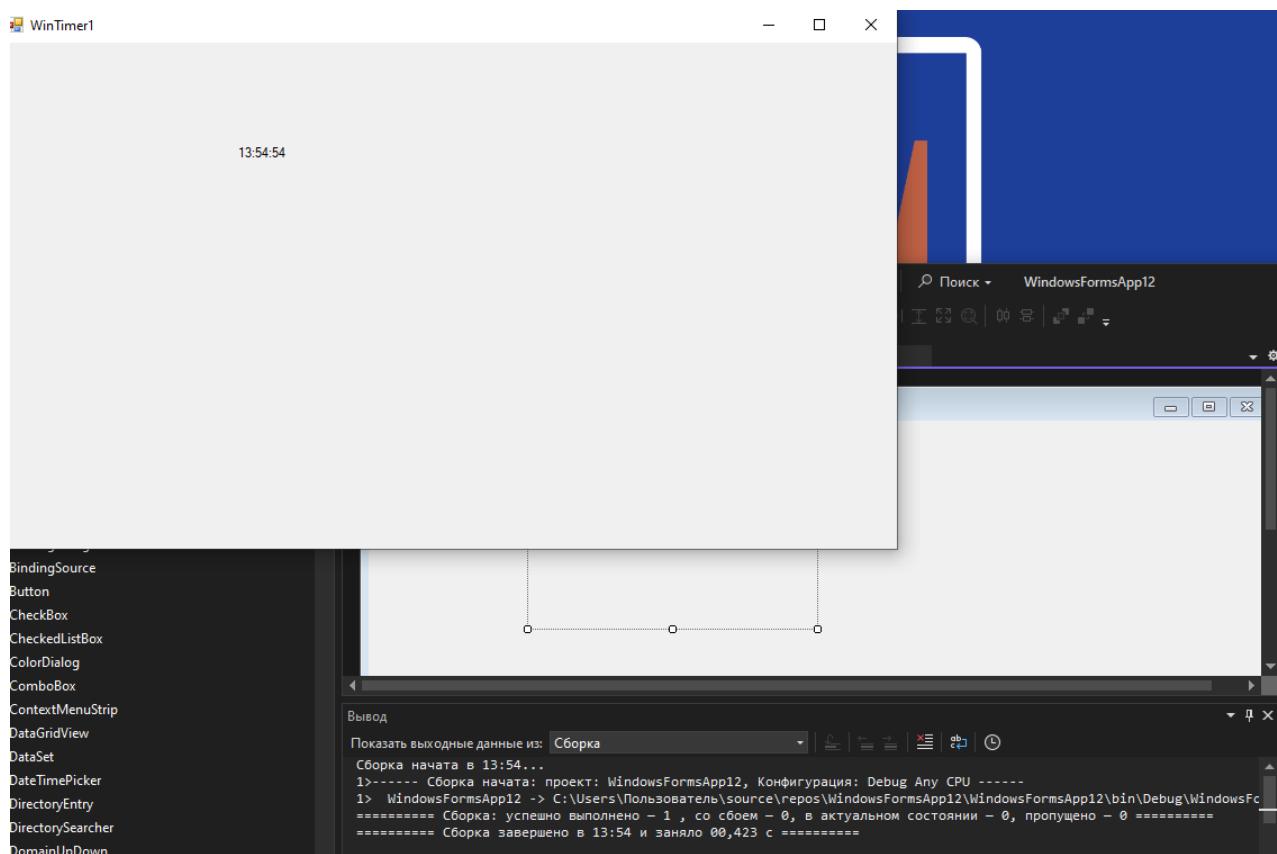
событий. Класс UserControl имеет собственный конструктор, позволяющий использовать в Visual Studio IDE перетаскивание дополнительных

элементов управления из Toolbox на поверхность конструктора и настраивать их.

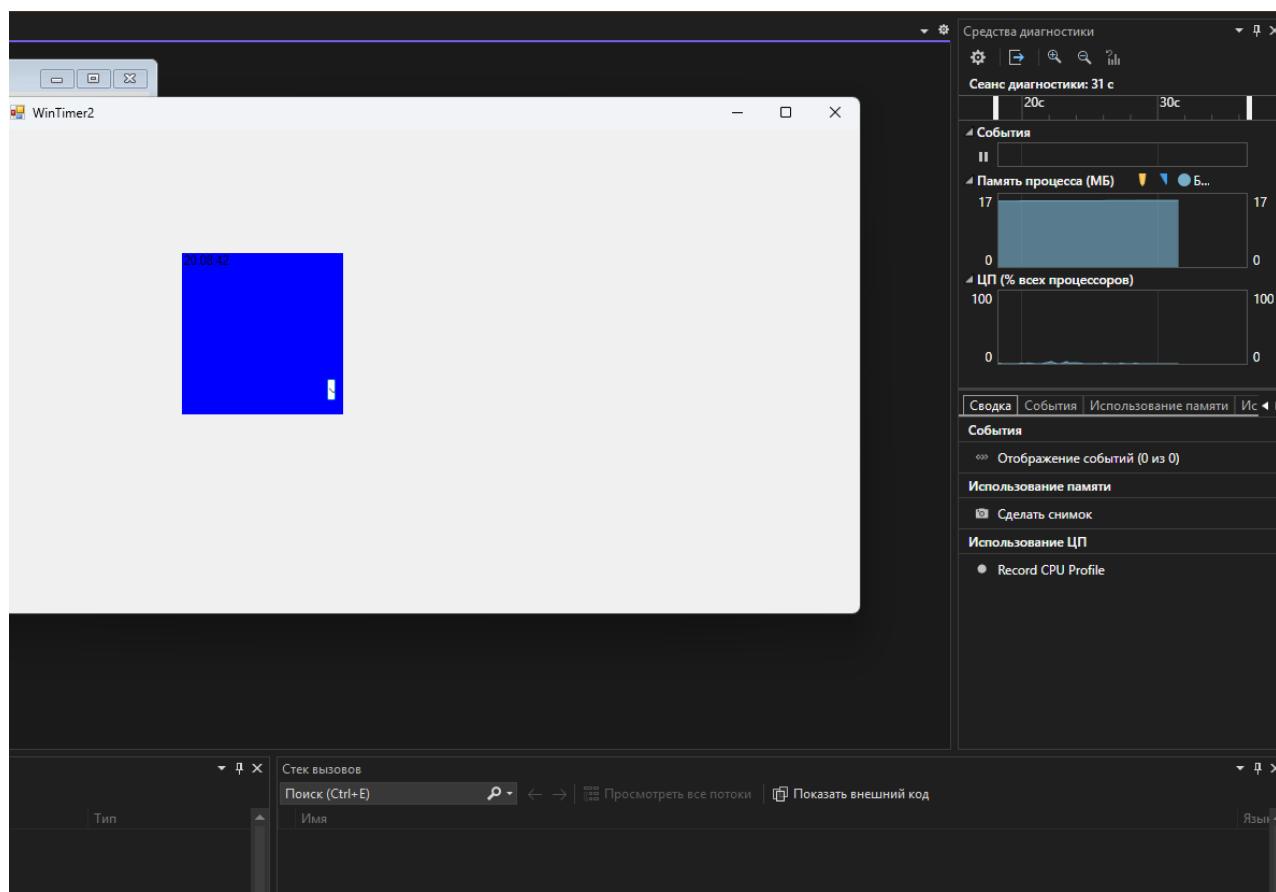
В этом упражнении вы создадите составной элемент управления, действующий как цифровые часы. В него вы добавите элемент управления Label, отображающий правильное время, и компонент Timer, каждую секунду обновляющий Label. Предоставив свойство Enabled элемента управления Timer, вы дадите пользователям возможность включать и отключать часы.

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

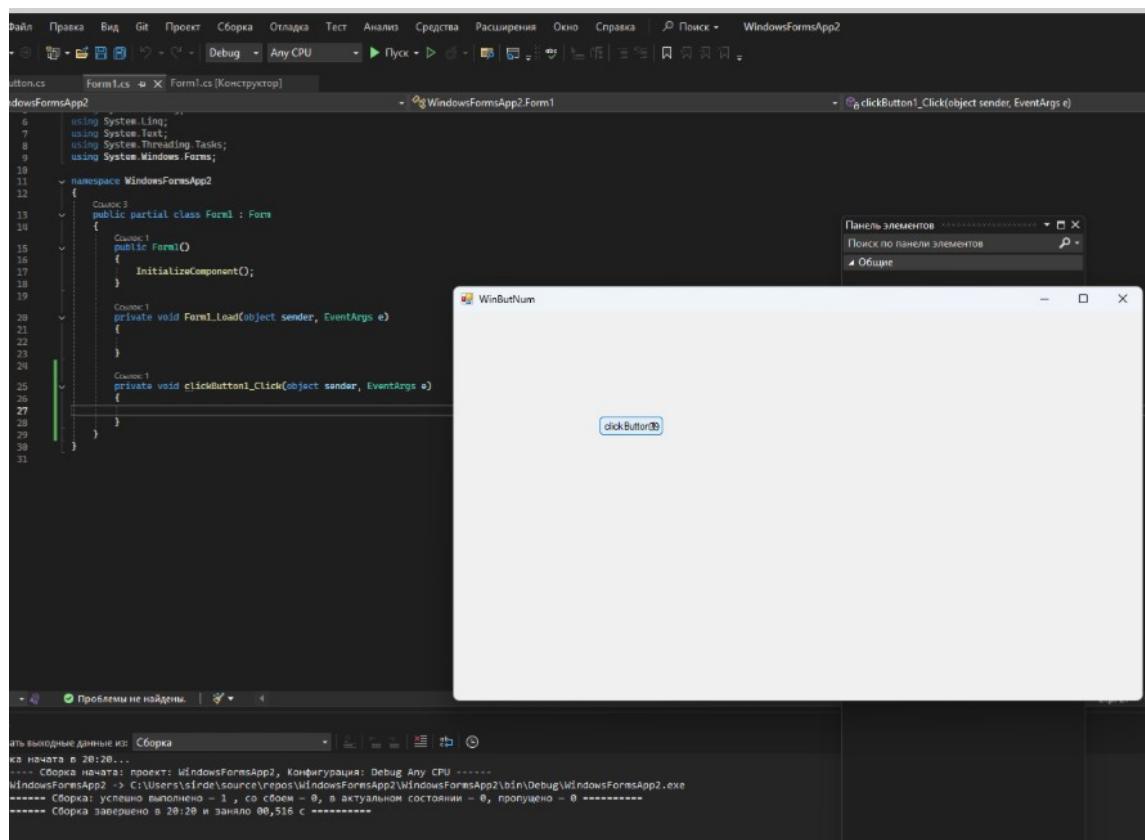


Создание специализированного элемента управления



Создание расширенных элементов управления

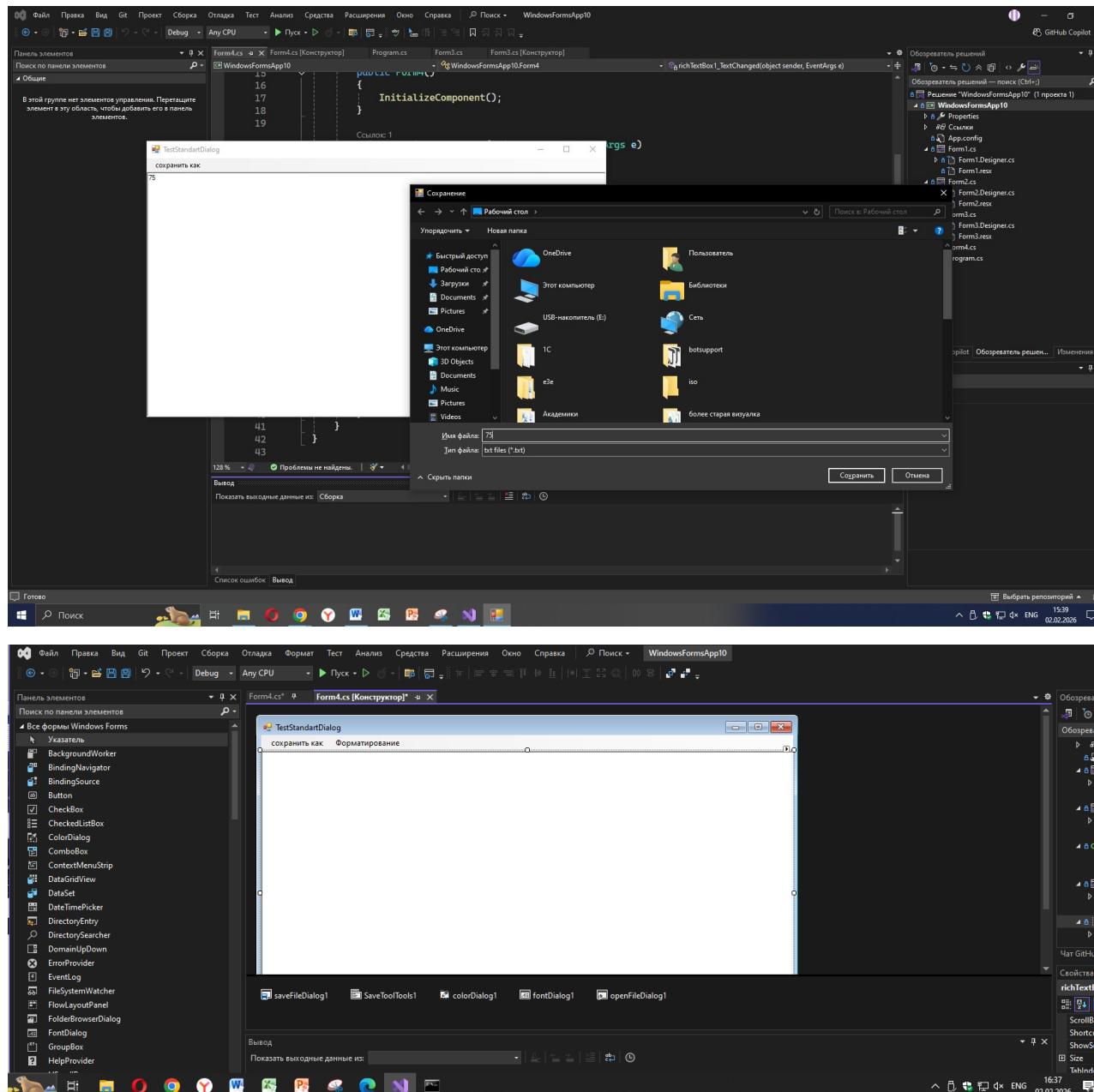
Отчет по Разработка и тестирование кода



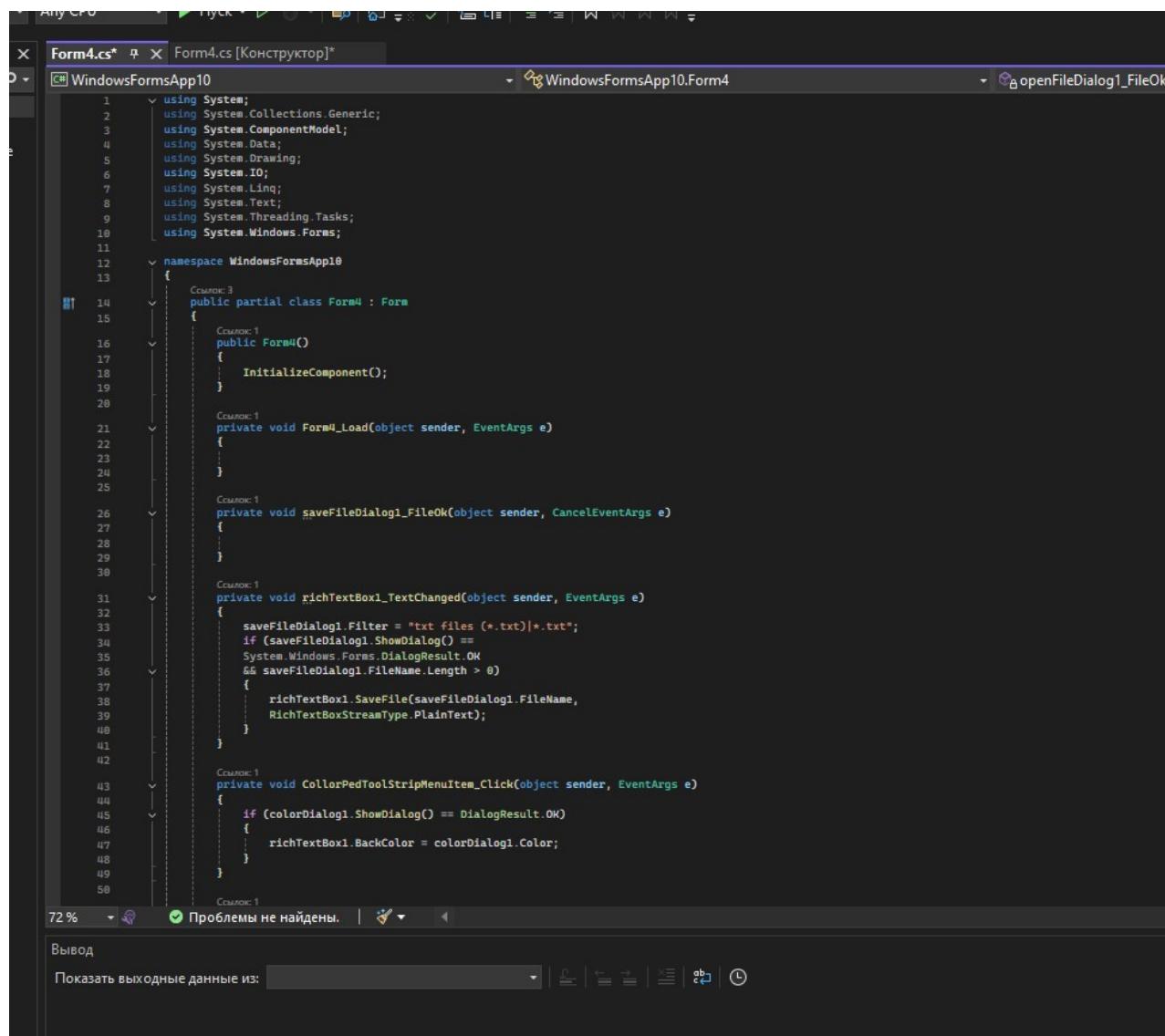
Лабораторная работа 4 Использование окон диалога в формах

Цель работы

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



Отчет по Разработка и тестирование кода



The screenshot shows the Microsoft Visual Studio IDE interface. The title bar displays "Any CPU" and "Form4.cs [Конструктор]*". The main window shows the code for Form4.cs:

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.IO;
7  using System.Linq;
8  using System.Text;
9  using System.Threading.Tasks;
10 using System.Windows.Forms;
11
12 namespace WindowsFormsApp10
13 {
14     public partial class Form4 : Form
15     {
16         public Form4()
17         {
18             InitializeComponent();
19         }
20
21         private void Form4_Load(object sender, EventArgs e)
22         {
23         }
24
25         private void saveFileDialog1_FileOk(object sender, CancelEventArgs e)
26         {
27         }
28
29         private void richTextBox1_TextChanged(object sender, EventArgs e)
30         {
31             saveFileDialog1.Filter = "txt files (*.txt)|*.txt";
32             if (saveFileDialog1.ShowDialog() ==
33                 System.Windows.Forms.DialogResult.OK
34                 && saveFileDialog1.FileName.Length > 0)
35             {
36                 richTextBox1.SaveFile(saveFileDialog1.FileName,
37                     RichTextBoxStreamType.PlainText);
38             }
39
40         }
41
42         private void ColorPsdToolStripMenuItem_Click(object sender, EventArgs e)
43         {
44             if (colorDialog1.ShowDialog() == DialogResult.OK)
45             {
46                 richTextBox1.BackColor = colorDialog1.Color;
47             }
48         }
49
50     }
51 }
```

The status bar at the bottom indicates "72%" and "Проблемы не найдены." (No problems found).

```
Form4.cs*  X  Form4.cs [Конструктор]*  WindowsFormsApp10  WindowsFormsApp10.Form4  openFileDialog1_FileOk(object sender, CancelEventArgs e)
```

```
    47    richTextBox1.BackColor = colorDialog1.Color;
    48
    49
    50
    51    }
    52
    53
    54
    55
    56    }
    57
    58    }
    59
    60    }
    61
    62
    63
    64    }
    65
    66
    67    Stream myStream = null;
    68    OpenFileDialog openFileDialog1 = new OpenFileDialog();
    69    openFileDialog1.InitialDirectory = @"c:\";
    70    openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*";
    71    openFileDialog1.FilterIndex = 2;
    72    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    73    {
    74        try
    75        {
    76            if ((myStream = openFileDialog1.OpenFile()) != null)
    77            {
    78                using (myStream)
    79                {
    80                    richTextBox1.LoadFile(openFileDialog1.FileName,
    81                    RichTextBoxStreamType.PlainText);
    82                }
    83            }
    84        catch (Exception ex)
    85        {
    86            MessageBox.Show("Error: Could not read file from disk: "
    87            + ex.Message);
    88        }
    89
    90
    91
    92    }
    93
    94}
```

72% ✅ Проблемы не найдены. | Стр: 84

Вывод

Показать выполненные линии из:

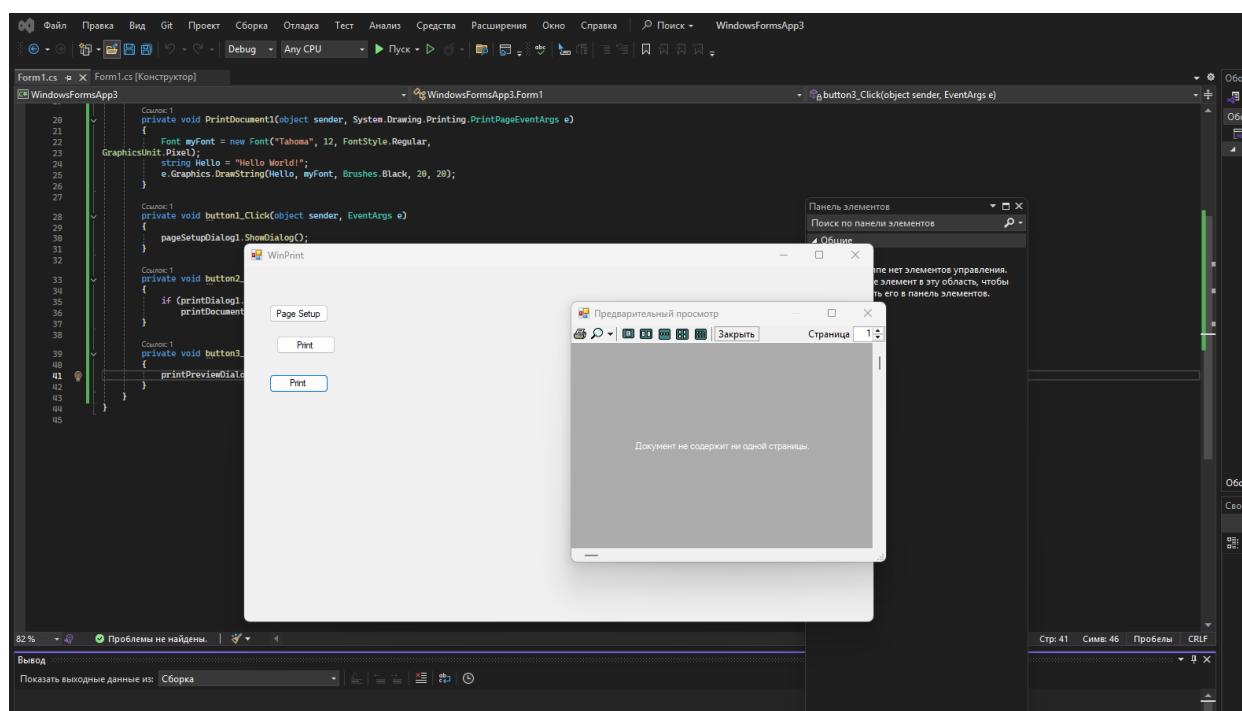
Лабораторная работа 6 Организация печати в формах

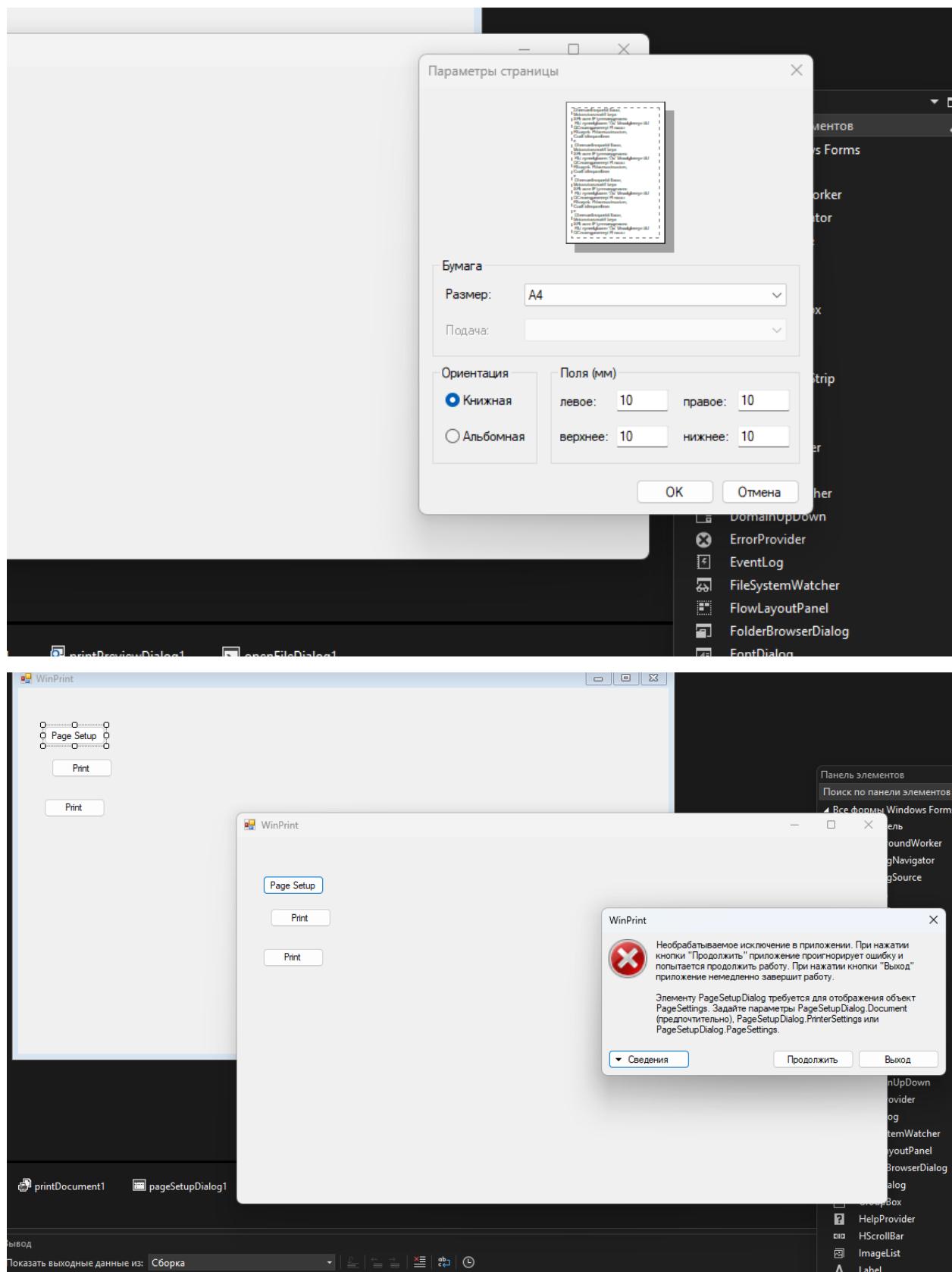
windows

Цель работы

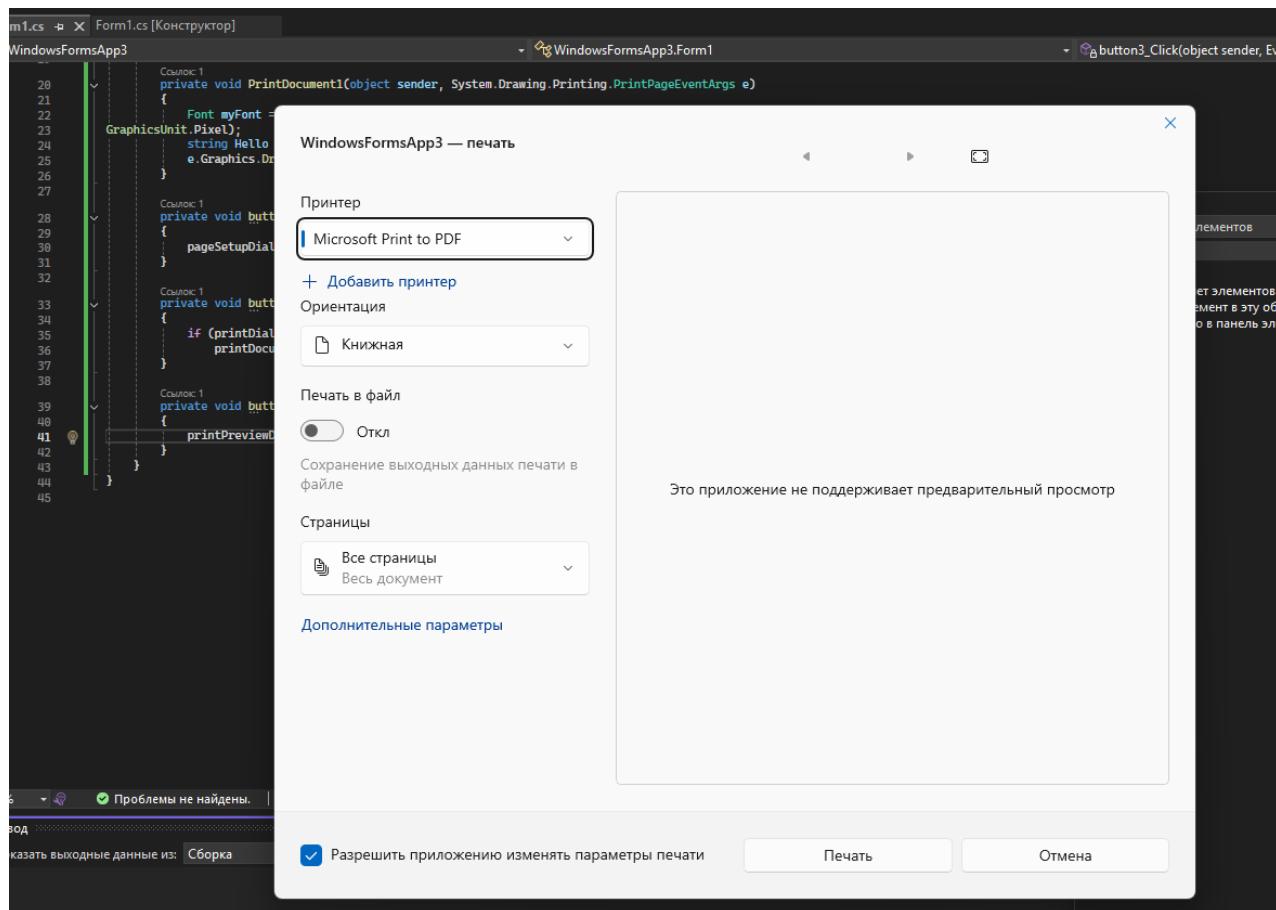
Изучение классов, реализующих задачу
программирования печати и
получение навыков по работе в программе с
диалоговыми окнами.

Упражнение 1 Использование диалоговых окон для печати

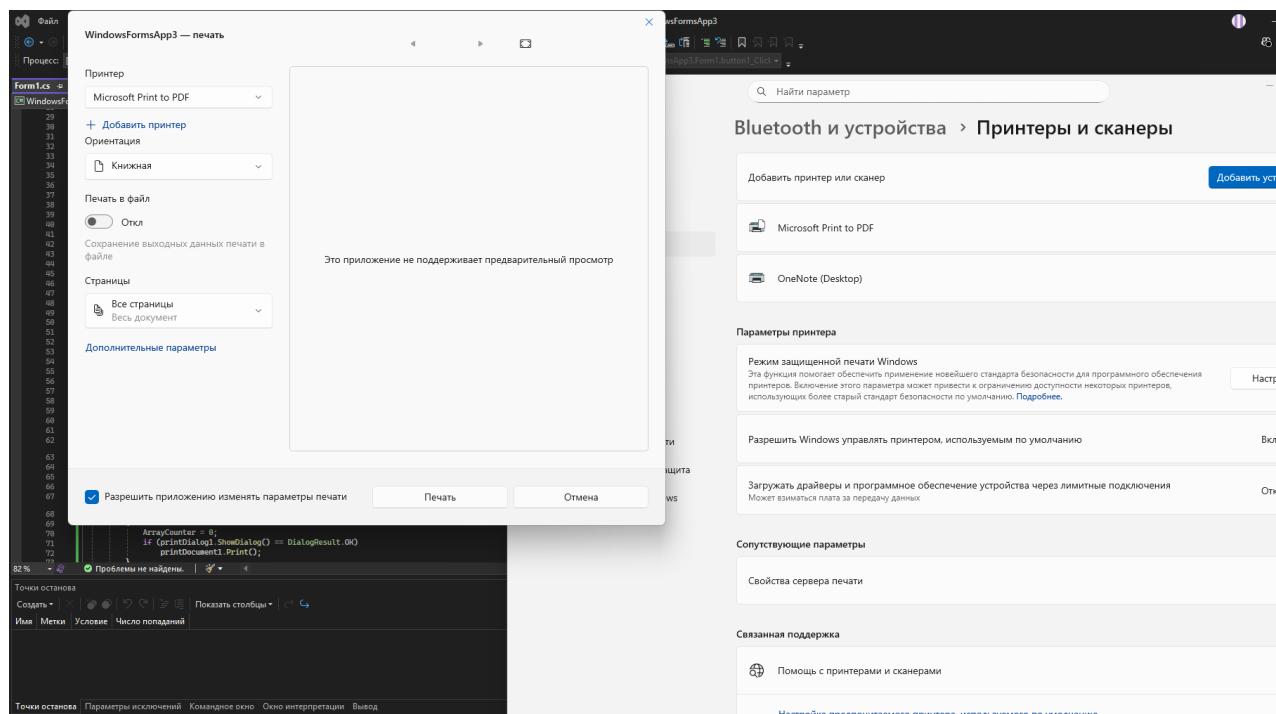




Отчет по Разработка и тестирование кода



Упражнение 2 Создание документа печати



```
private int ArrayCounter = 0;

class1
public Form1()
{
    InitializeComponent();
}

class1
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 VPosition = 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];
            VPosition = TopMargin + Counter * this.Font.GetHeight(e.Graphics);
            e.Graphics.DrawString(CurrentLine, this.Font, Brushes.Black, LeftMargin, VPosition, ref 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, 10);
    }
}

class1
private void button1_Click(object sender, EventArgs e)
{
    pageSetupDialog1.ShowDialog();
}

class1
private void button2_Click(object sender, EventArgs e)
{
    ArrayCounter = 0;
    if (printDialog1.ShowDialog() == DialogResult.OK)
        printDocument1.Print();
}

class1
private void button3_Click(object sender, EventArgs e)
{
    // Проблемы не найдены.
}
```

НОВА



Показать столбцы | ⌂ ⌂ ↻

```
Form1.cs  Form1.cs [Конструктор]
WindowsFormsApp3  WindowsFormsApp3.Form1.cs

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  private void button1_Click(object sender, EventArgs e)
64  {
65      pageSetupDialog1.ShowDialog();
66  }
67
68  private void button2_Click(object sender, EventArgs e)
69  {
70      ArrayCounter = 0;
71      if (printDialog1.ShowDialog() == DialogResult.OK)
72          printDocument1.Print();
73  }
74
75  private void button3_Click(object sender, EventArgs e)
76  {
77      ArrayCounter = 0;
78      printPreviewDialog1.ShowDialog();
79  }
80
81  private void button4_Click(object sender, EventArgs e)
82  {
83      DialogResult aResult = openFileDialog1.ShowDialog();
84
85      if (aResult == DialogResult.OK)
86      {
87          try
88          {
89              StreamReader aReader = new StreamReader(openFileDialog1.FileName);
90              s = aReader.ReadToEnd();
91              aReader.Close();
92              strings = s.Split('\n');
93              ArrayCounter = 0;
94
95              MessageBox.Show("Файл загружен успешно!", "Информация",
96                             MessageBoxButtons.OK, MessageBoxIcon.Information);
97          }
98          catch (Exception ex)
99          {
100             MessageBox.Show("Ошибка при чтении файла: " + ex.Message,
101                         "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
102         }
103     }
104
105  private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
106  {
107  }
108
109 }
```

Упражнение 3

Создание специализированной формы предварительного

