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.

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

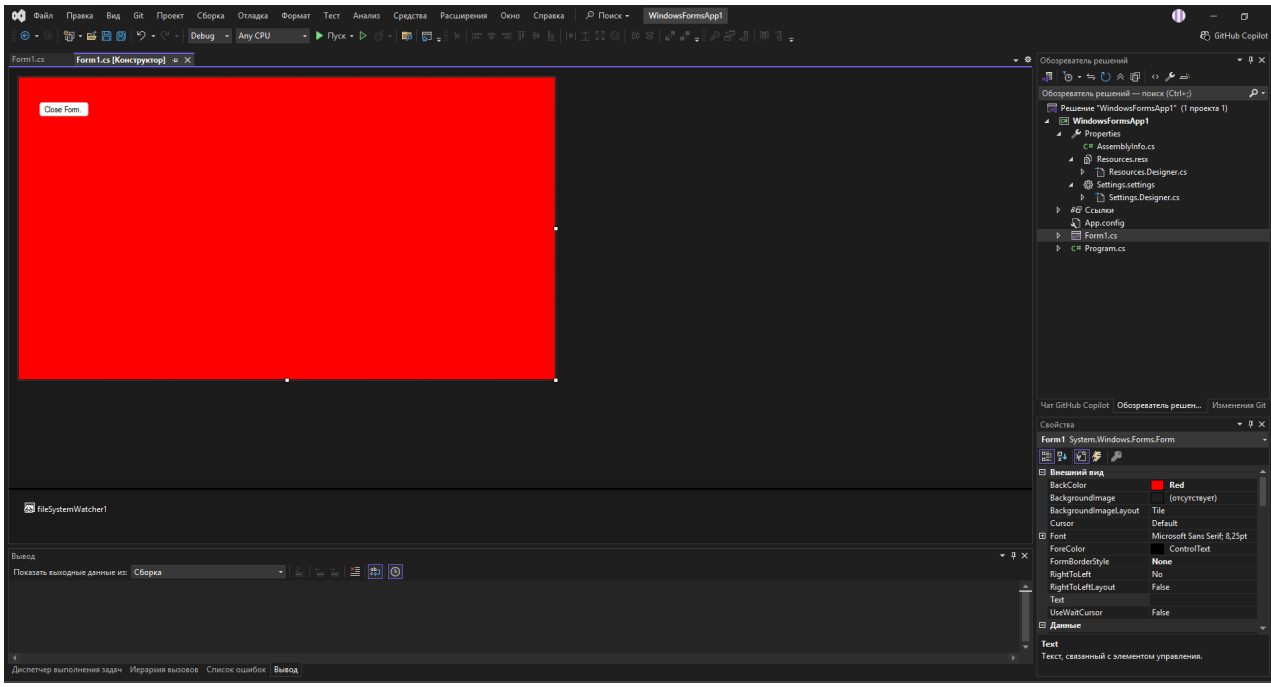
обработчик события Form1 Load. Добавьте в этот метод следующий код

(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.



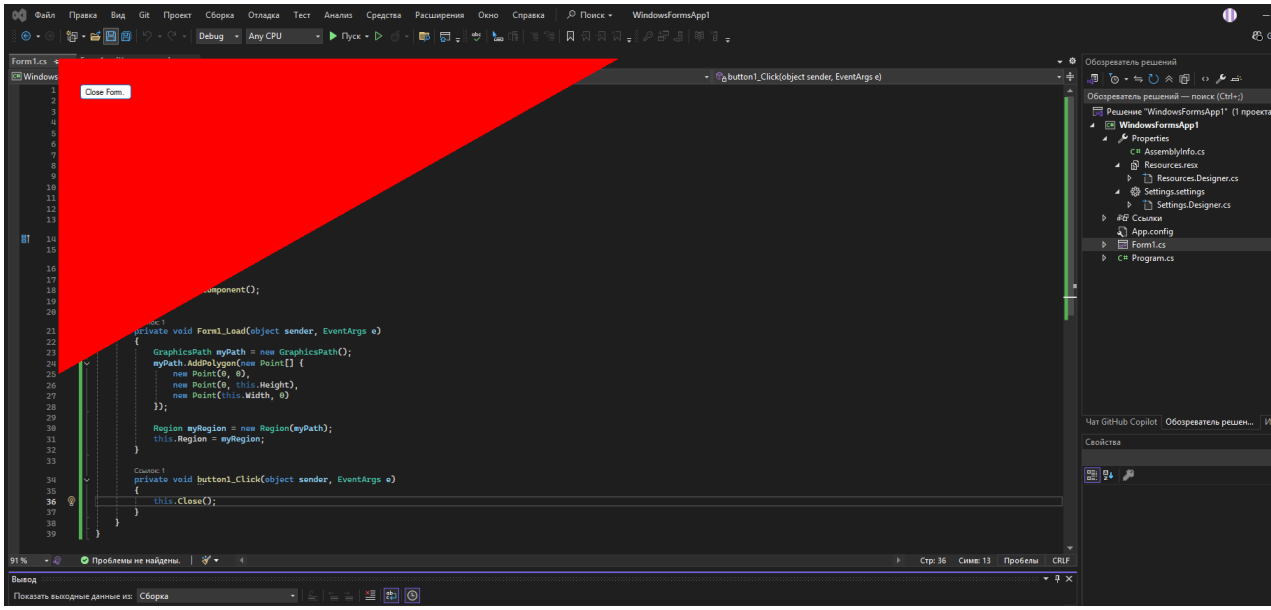

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            GraphicsPath myPath = new 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;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.Close();
        }
    }
}
```



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:

a. 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

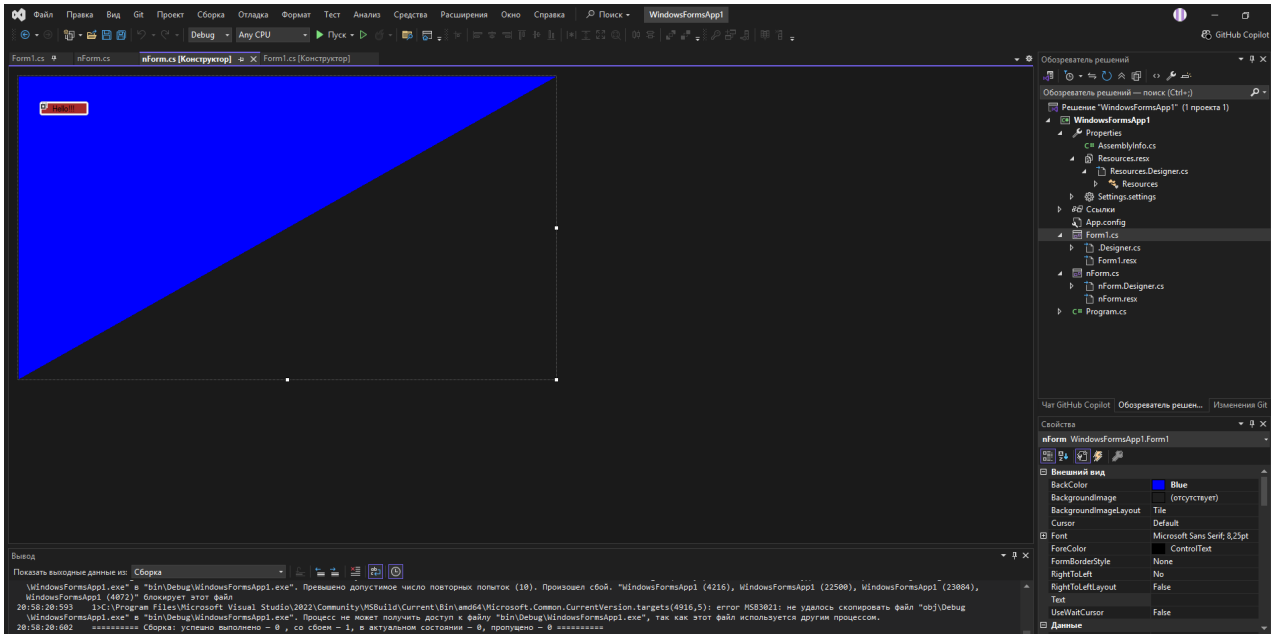
функции Main следующий код:

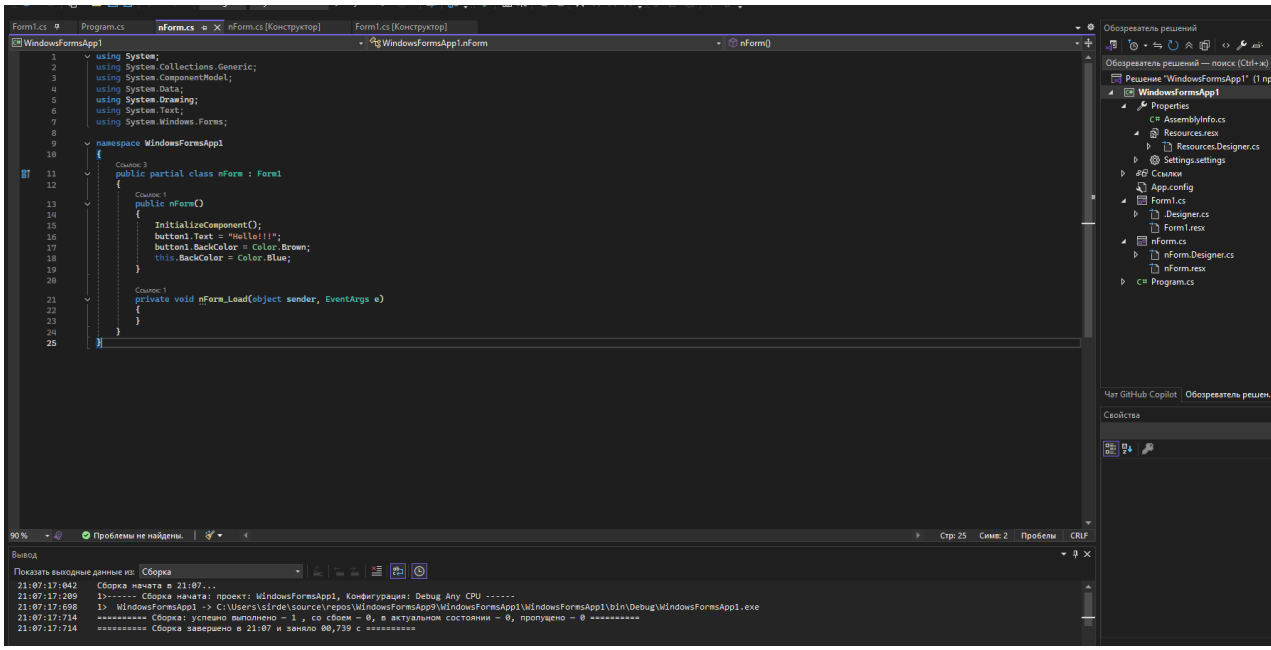
Application.Run(new nForm());

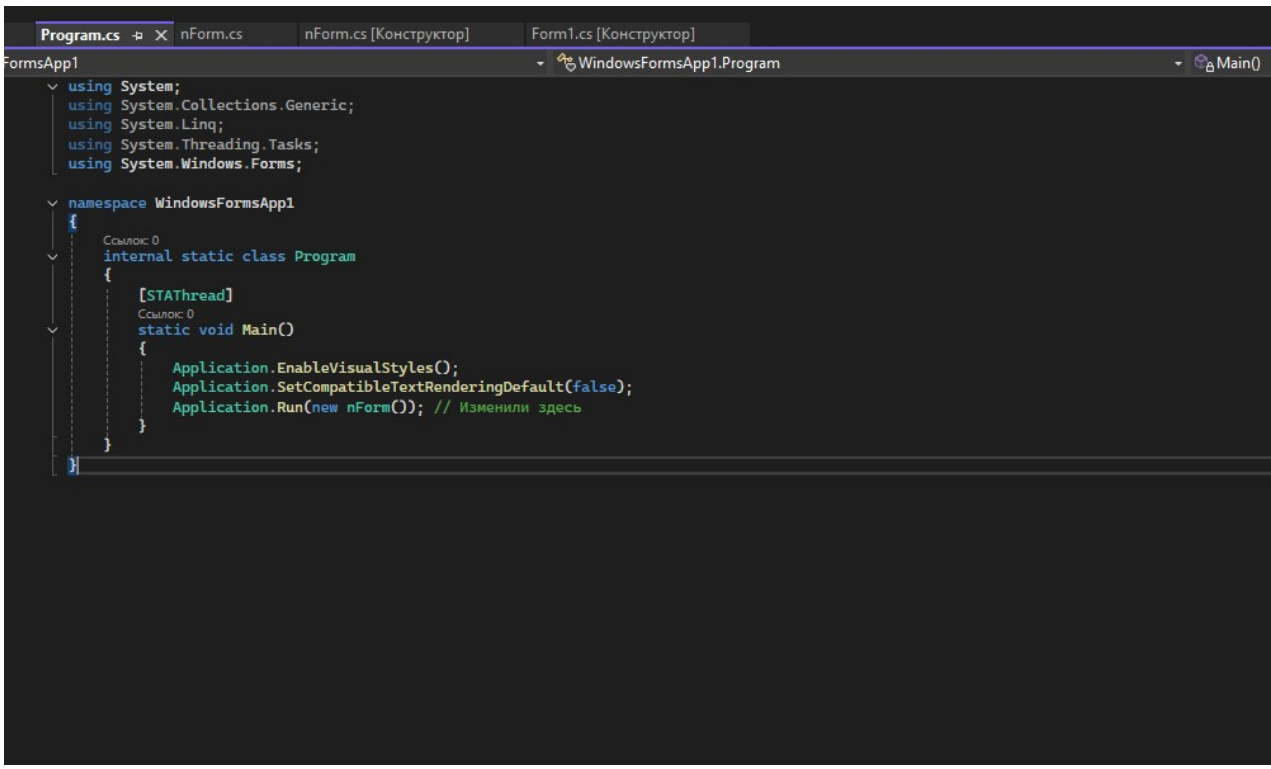
11 Постройте и запустите приложение. Должна открыться

производная форма со своими свойствами. Проверьте, наследуется ли

закрытие формы кнопкой.







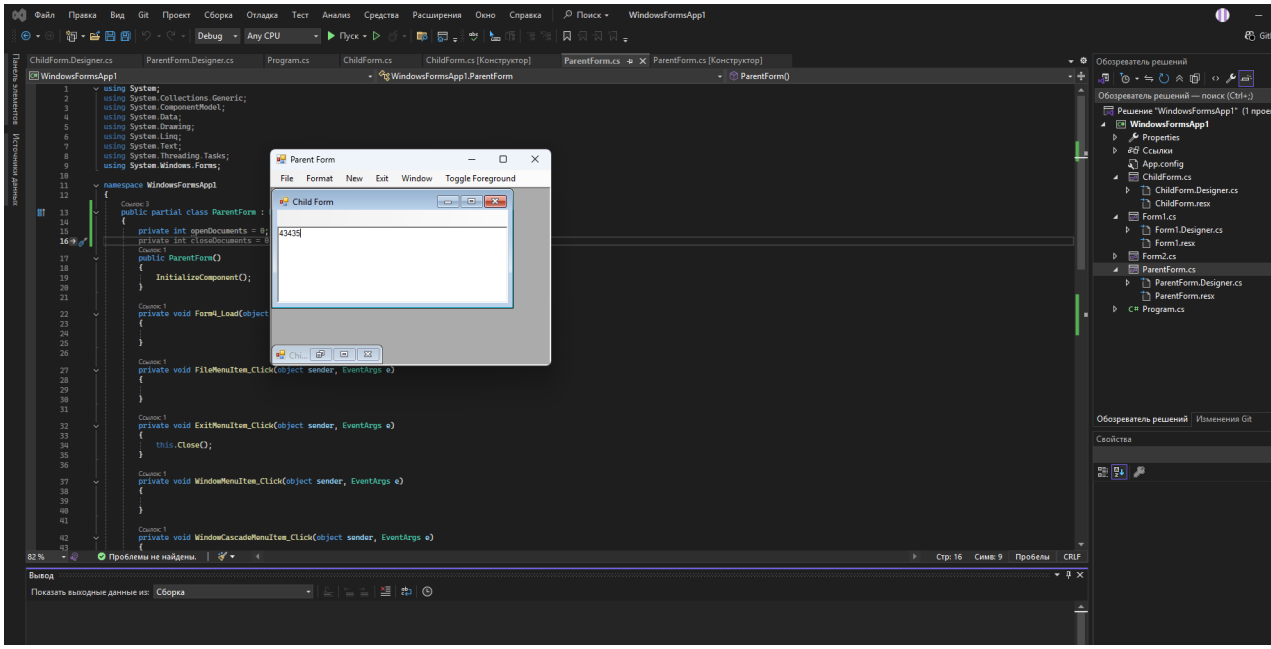
Упражнение 4 Создание MDI-приложения

В этом упражнении Вы создадите MDI-приложение с родительской

формой, загружающей и организующей дочерние формы. Также Вы

познакомитесь с элементом управления MenuStrip, который позволяет

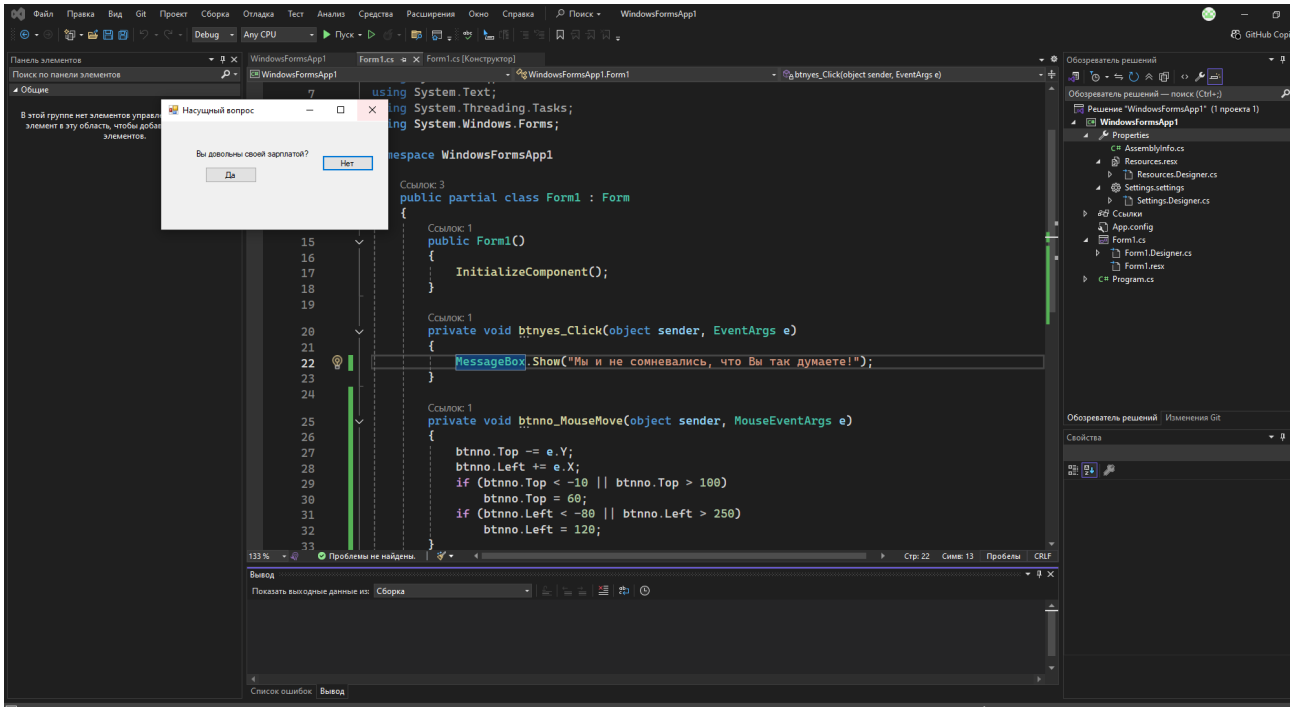
создать меню формы.

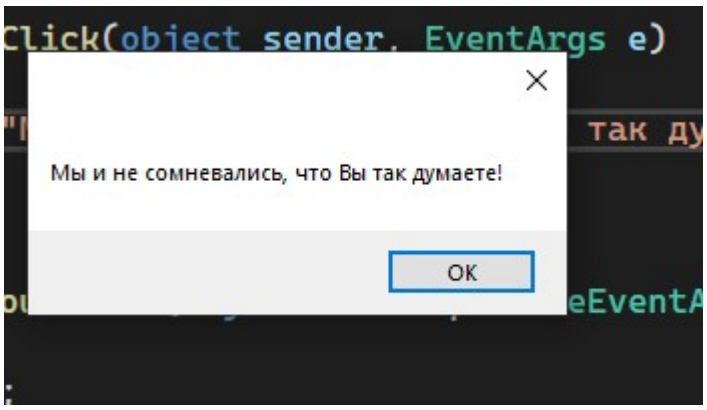


2. Лабораторная работа

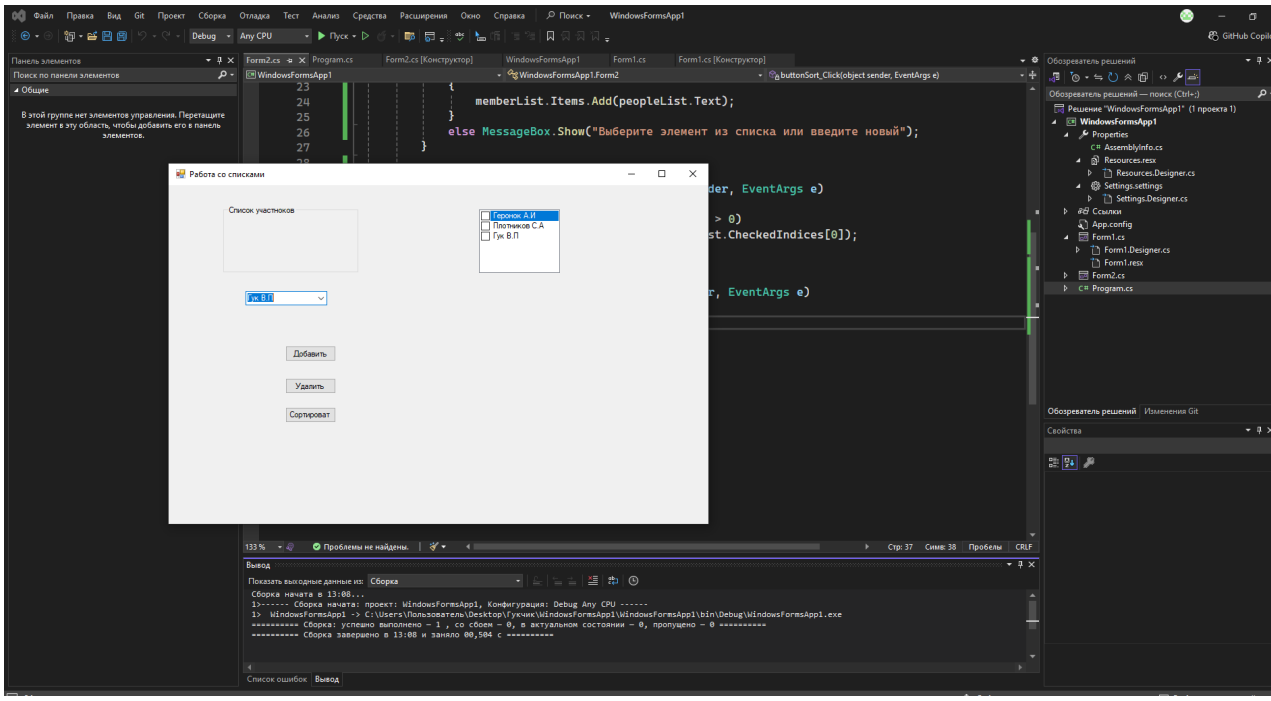
Упражнение 1 Обработка событий Click и MouseMove

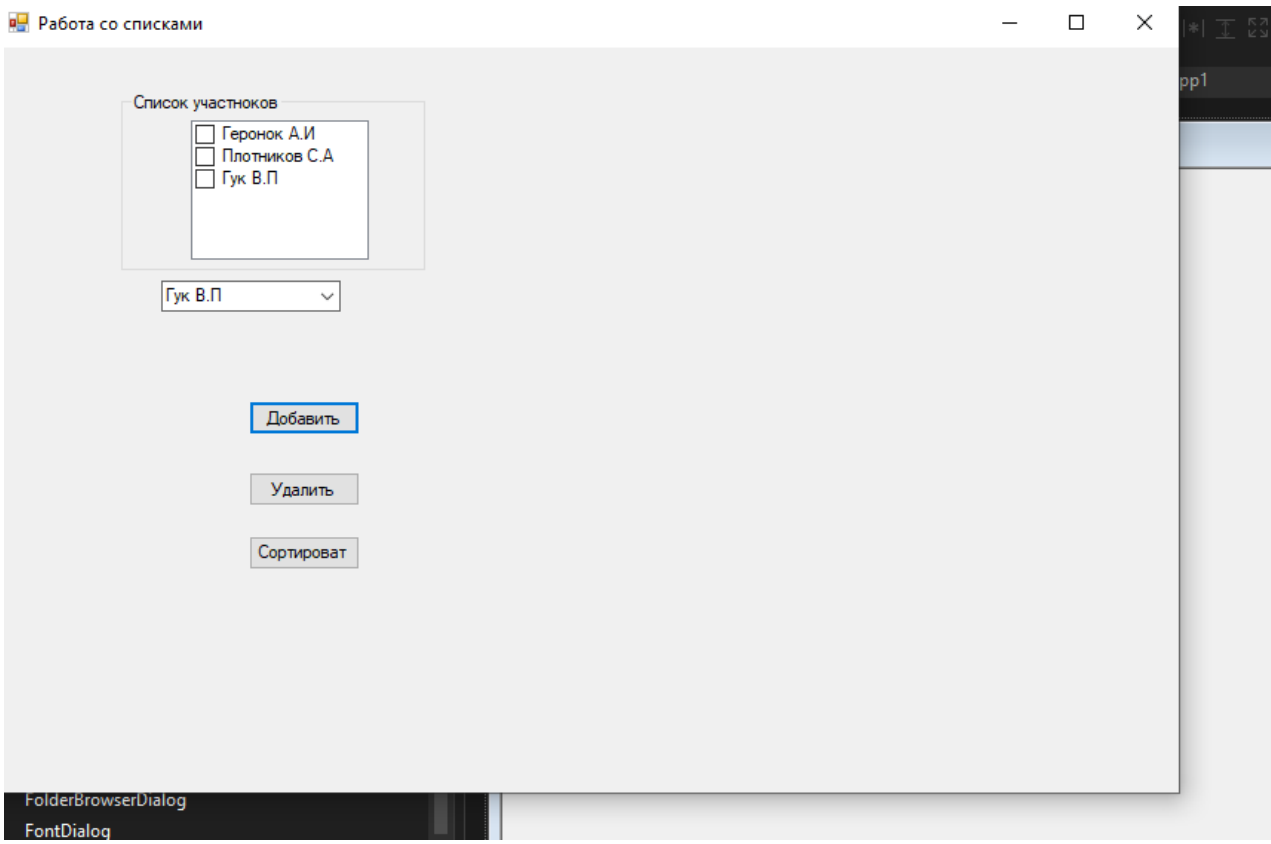
ф





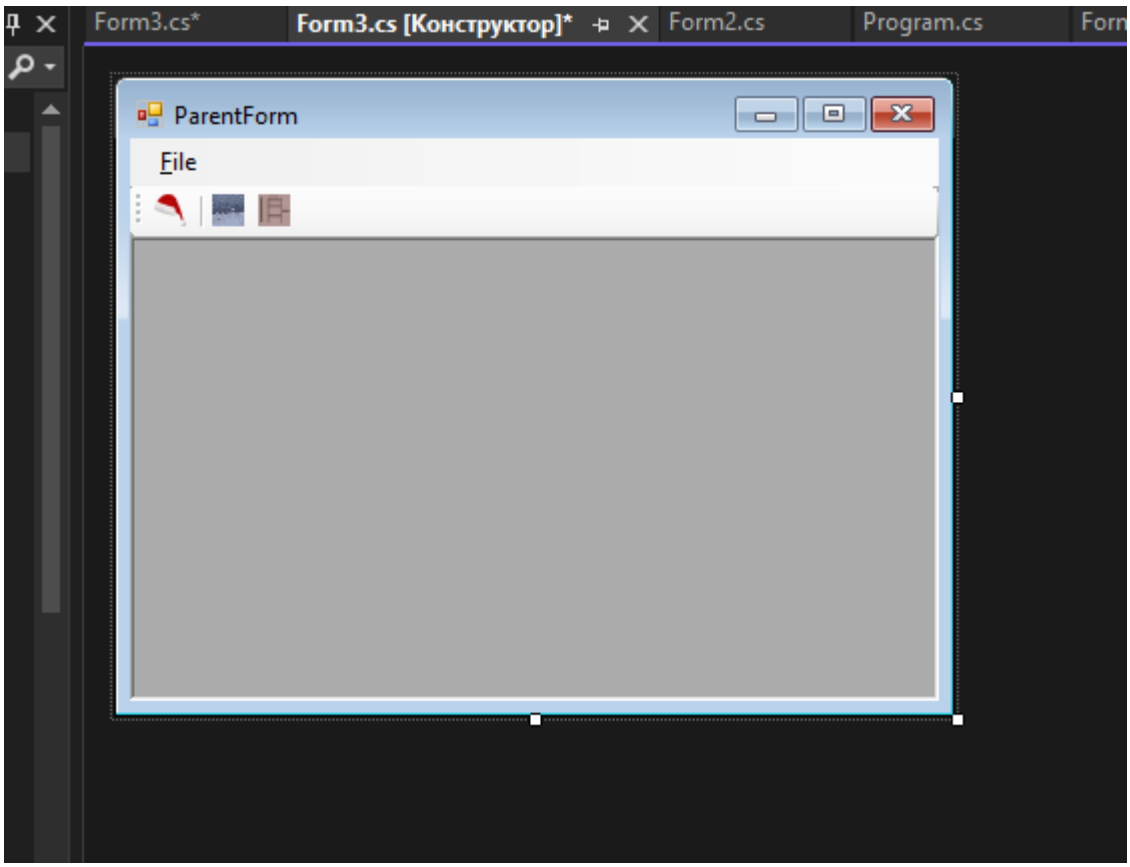
Упражнение 2 Работа со списками





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.


Form3.cs*
Form3.cs [Конструктор]*
Form2.cs
Program.cs
ParentForm
File

Лабораторная работа 3. Создание элементов управления

Цель работы

Изучение способов разработки элементов управления и получение навыков по их настойке 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

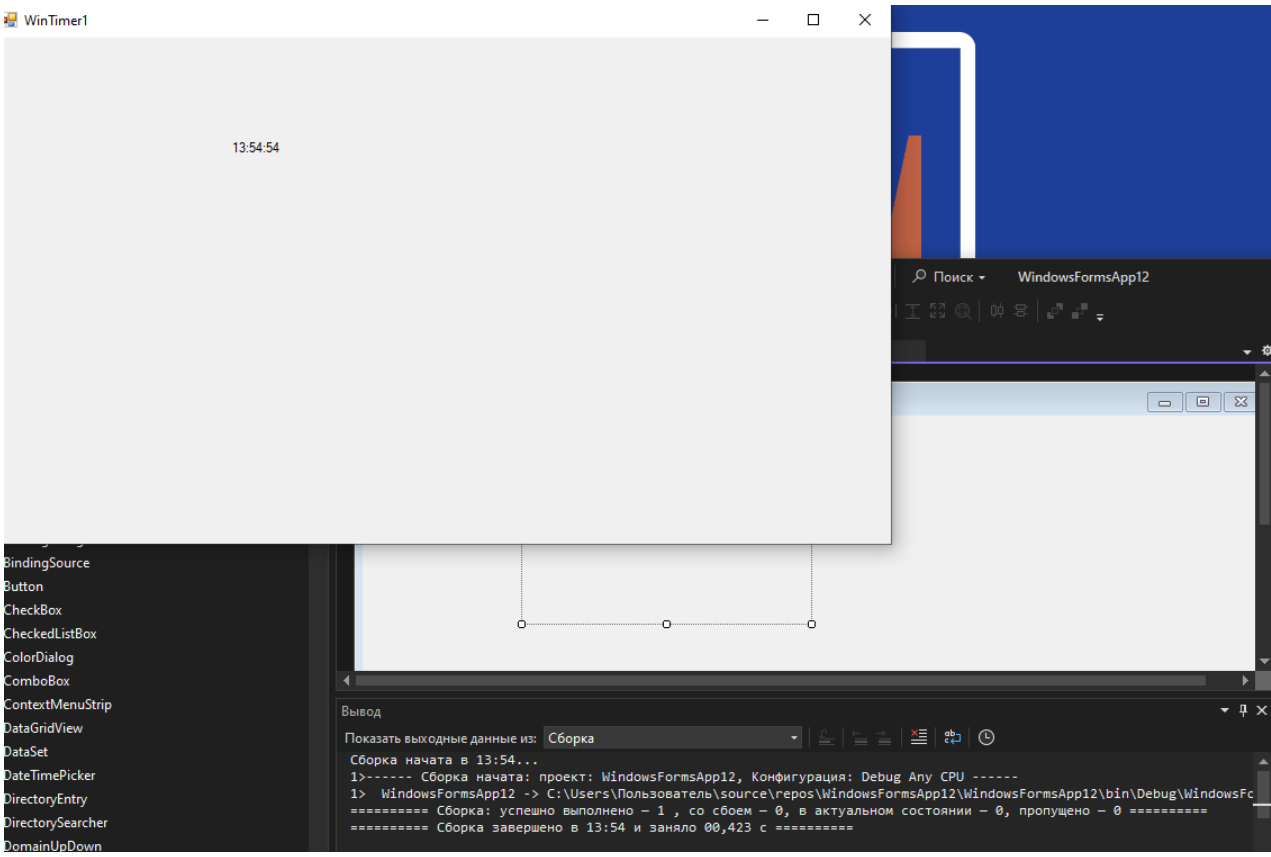
событий. Класс UserControl имеет собственный конструктор, позволяю-щий использовать в Visual Studio IDE перетаскивание дополнительных

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

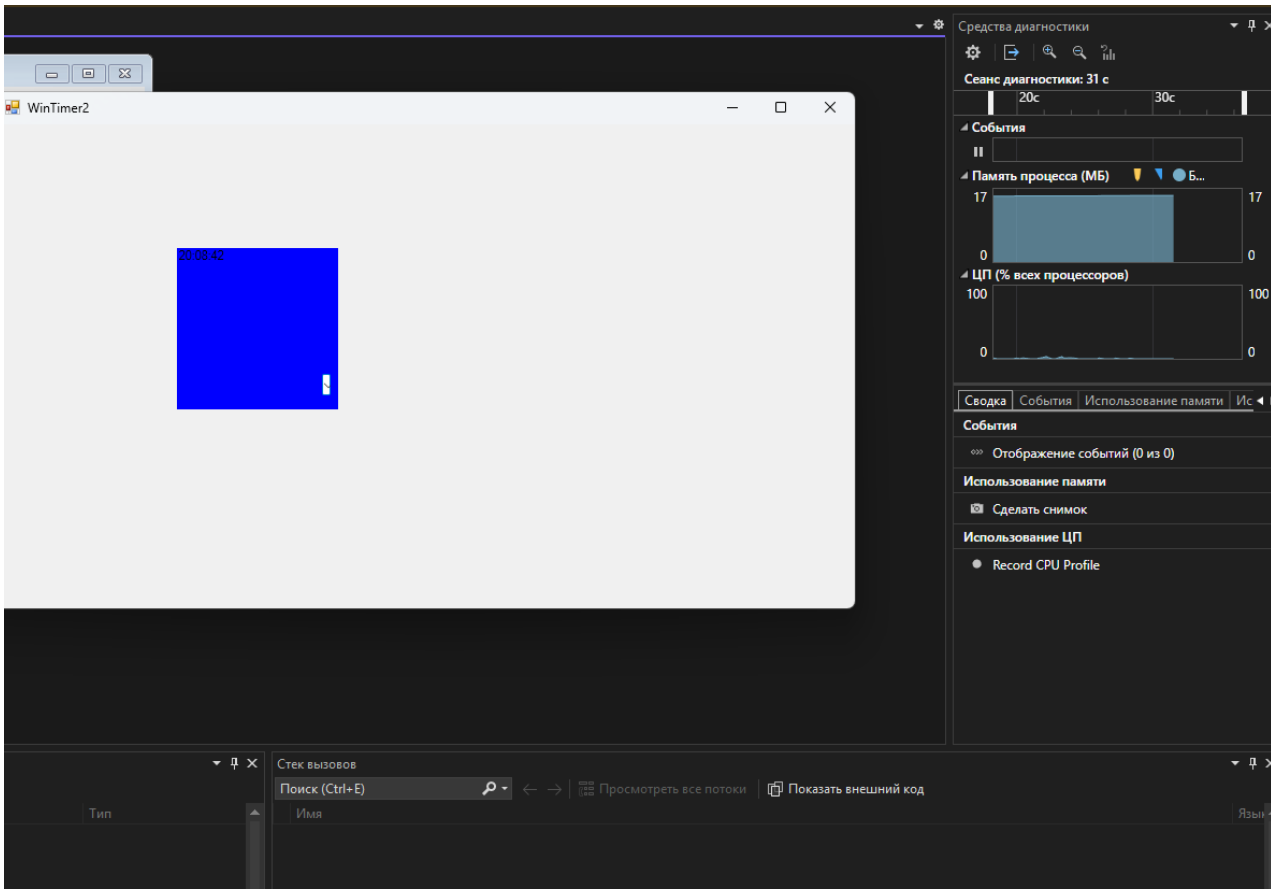
Label, отображающий правильное время, и компонент Timer, каждую секунду обновляющий Label. Предоставив свойство Enabled элемента управления Timer, вы дадите пользователям возможность включать и отключать часы.

Разработка составного элемента управления

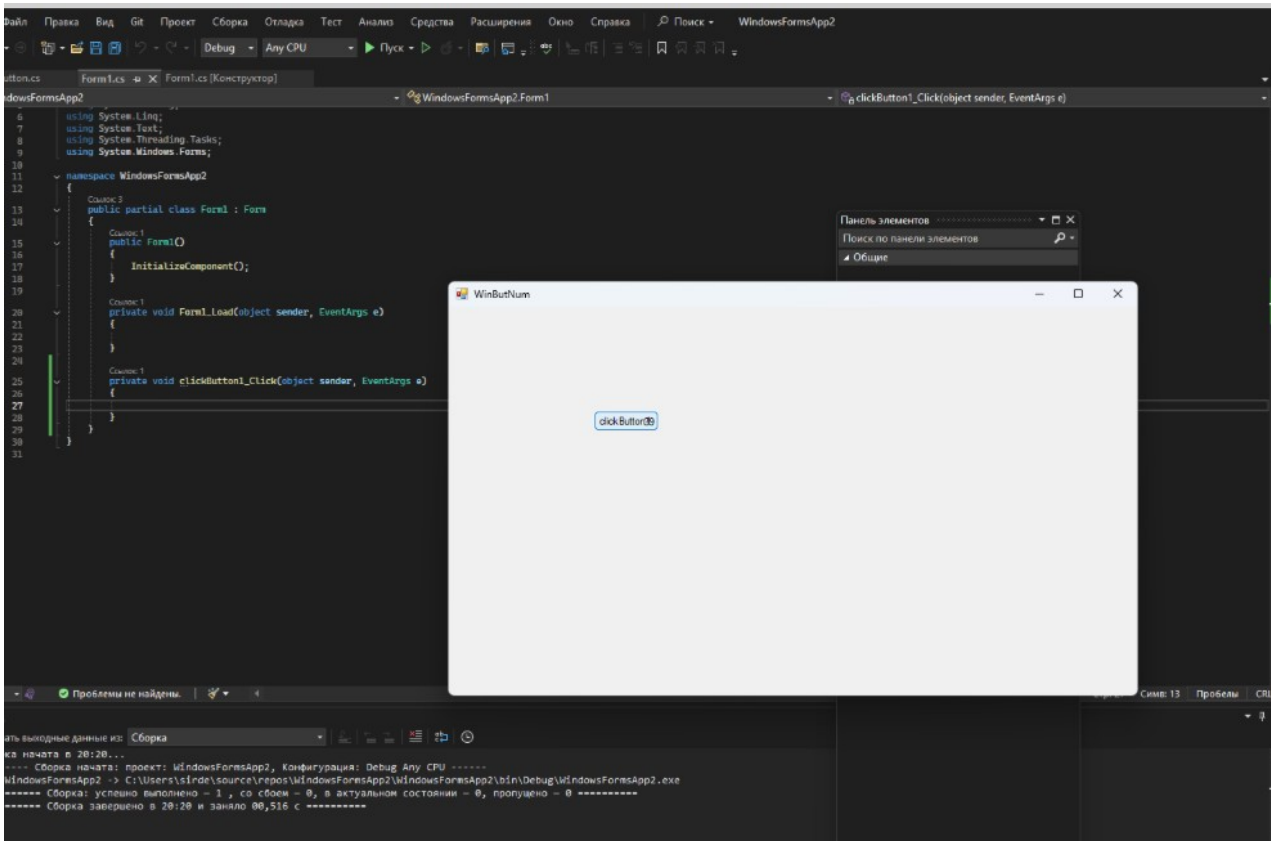


Создание специализированного элемента

управления



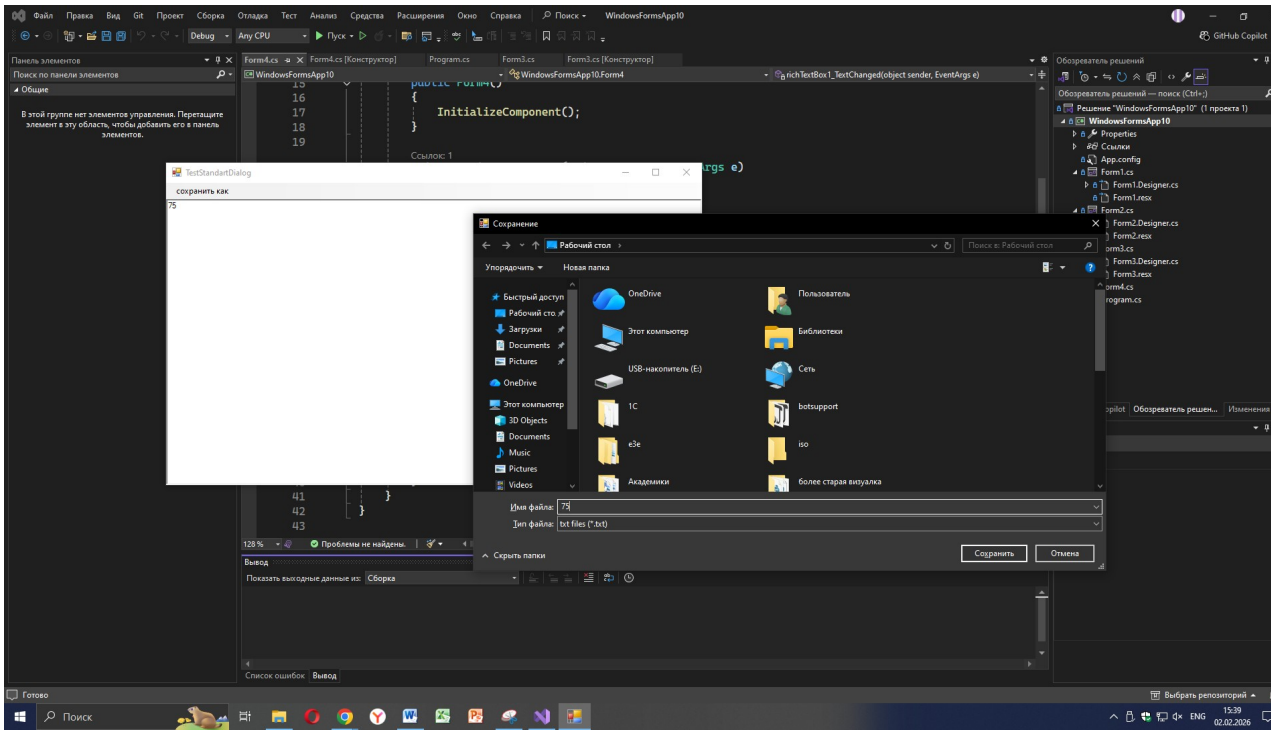
Создание расширенных элементов управления

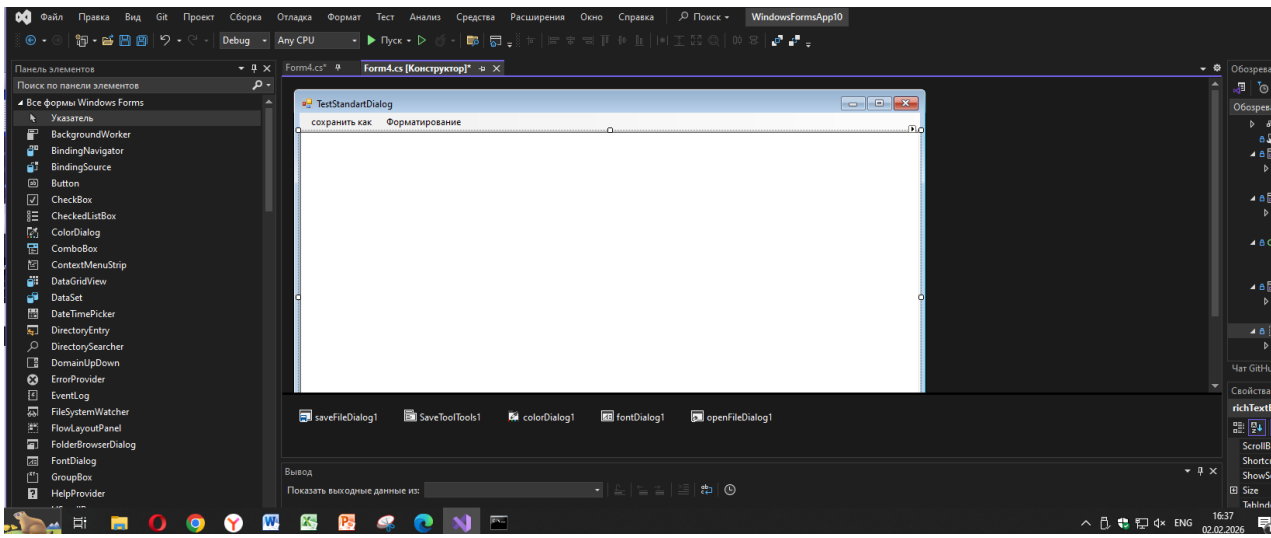

Файл Правка Вид Git Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
WindowsFormsApp2
Debug
Any CPU
Пуск
Form1.cs
Form1.cs [Конструктор]
WindowsFormsApp2.Form1
clickButton1_Click(object sender, EventArgs e)
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace WindowsFormsApp2
public partial class Form1 : Form
public Form1()
InitializeComponent();
private void Form1_Load(object sender, EventArgs e)
private void clickButton1_Click(object sender, EventArgs e)
Панель элементов
Поиск по панели элементов
Общие
WinButNum
clickButton39
Проблемы не найдены.
Сборка
Сборка начата: проект: WindowsFormsApp2, Конфигурация: Debug Any CPU
WindowsFormsApp2 -> C:\Users\sirde\source\repos\WindowsFormsApp2\WindowsFormsApp2\bin\Debug\WindowsFormsApp2.exe
Сборка: успешно выполнено – 1 , со сбоем – 0, в актуальном состоянии – 0, пропущено – 0
Сборка завершено в 20:20 и заняло 00,516 с

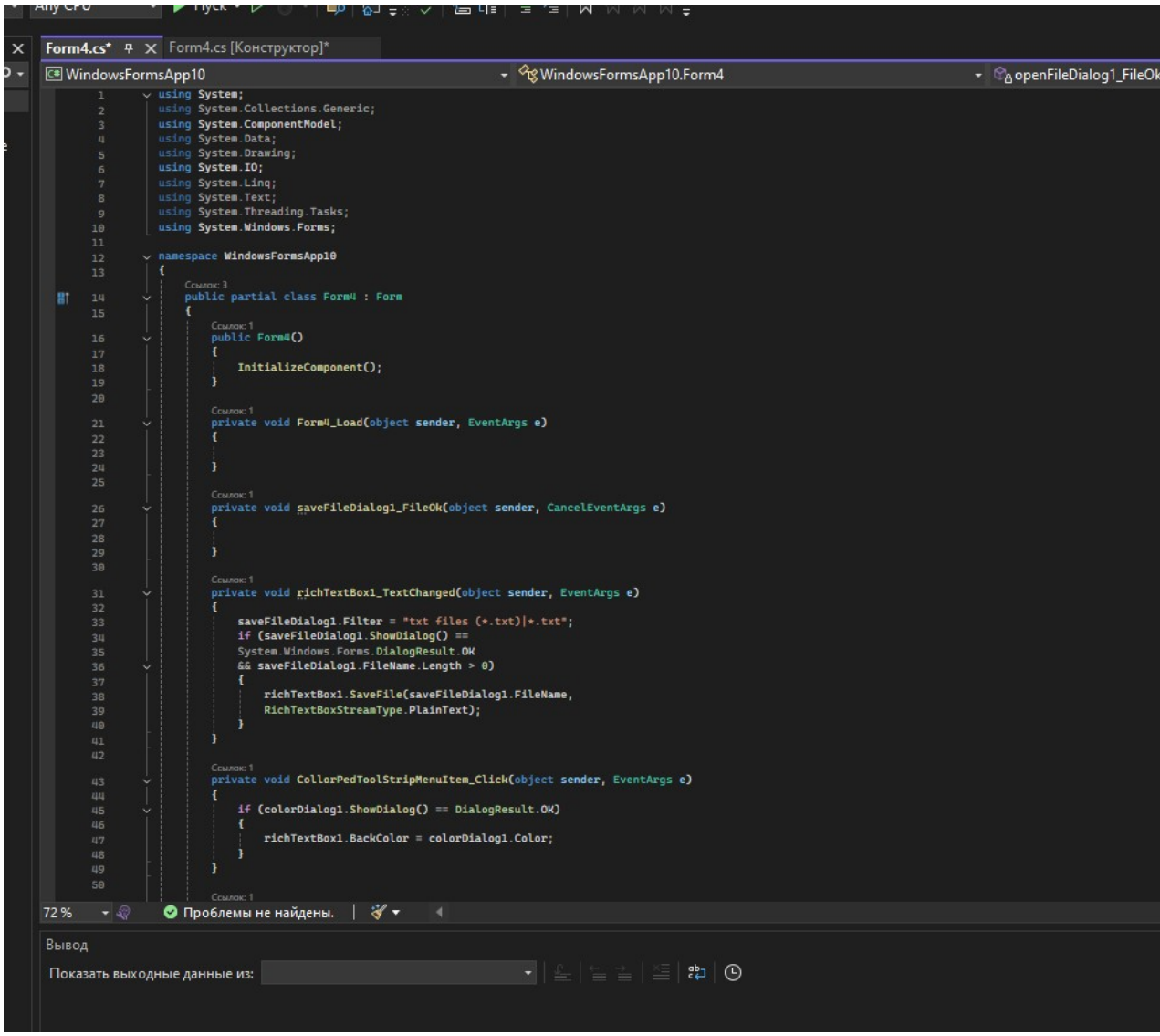
Лабораторная работа 4 Использование окон диалога в формах

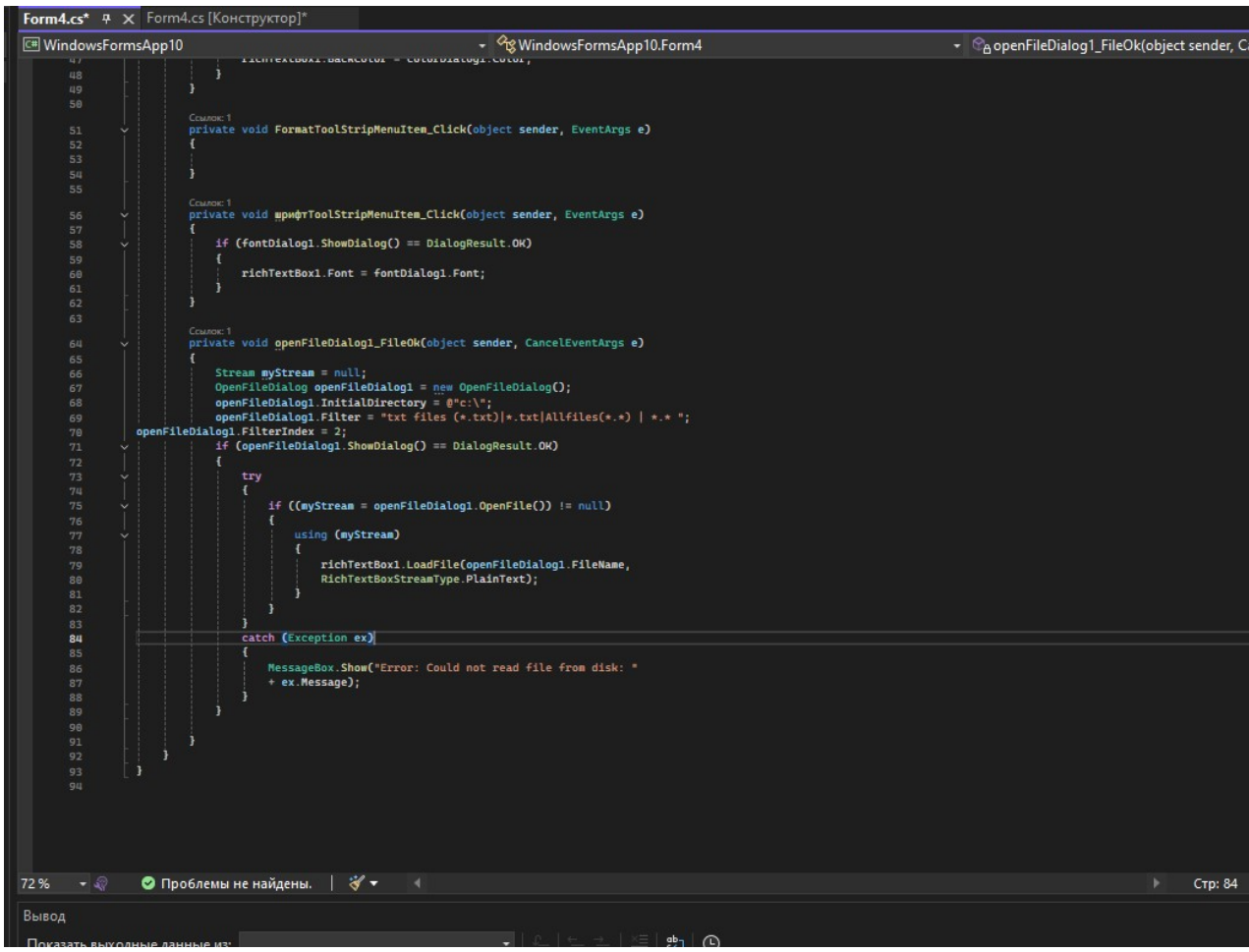
Цель работы

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






Form4.cs*
Form4.cs [Конструктор]*
WindowsFormsApp10
WindowsFormsApp10.Form4
openFileDialog1_FileOk
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace WindowsFormsApp10
{
Ссылок: 3
public partial class Form4 : Form
{
Ссылок: 1
public Form4()
{
InitializeComponent();
}
Ссылок: 1
private void Form4_Load(object sender, EventArgs e)
{
}
Ссылок: 1
private void saveFileDialog1_FileOk(object sender, CancelEventArgs e)
{
}
Ссылок: 1
private void richTextBox1_TextChanged(object sender, EventArgs e)
{
saveFileDialog1.Filter = "txt files (*.txt)|*.txt";
if (saveFileDialog1.ShowDialog() ==
System.Windows.Forms.DialogResult.OK
&& saveFileDialog1.FileName.Length > 0)
{
richTextBox1.SaveFile(saveFileDialog1.FileName,
RichTextBoxStreamType.PlainText);
}
}
Ссылок: 1
private void CollorPedToolStripMenuItem_Click(object sender, EventArgs e)
{
if (colorDialog1.ShowDialog() == DialogResult.OK)
{
richTextBox1.BackColor = colorDialog1.Color;
}
}
Ссылок: 1
72 %
Проблемы не найдены.
Вывод
Показать выходные данные из:

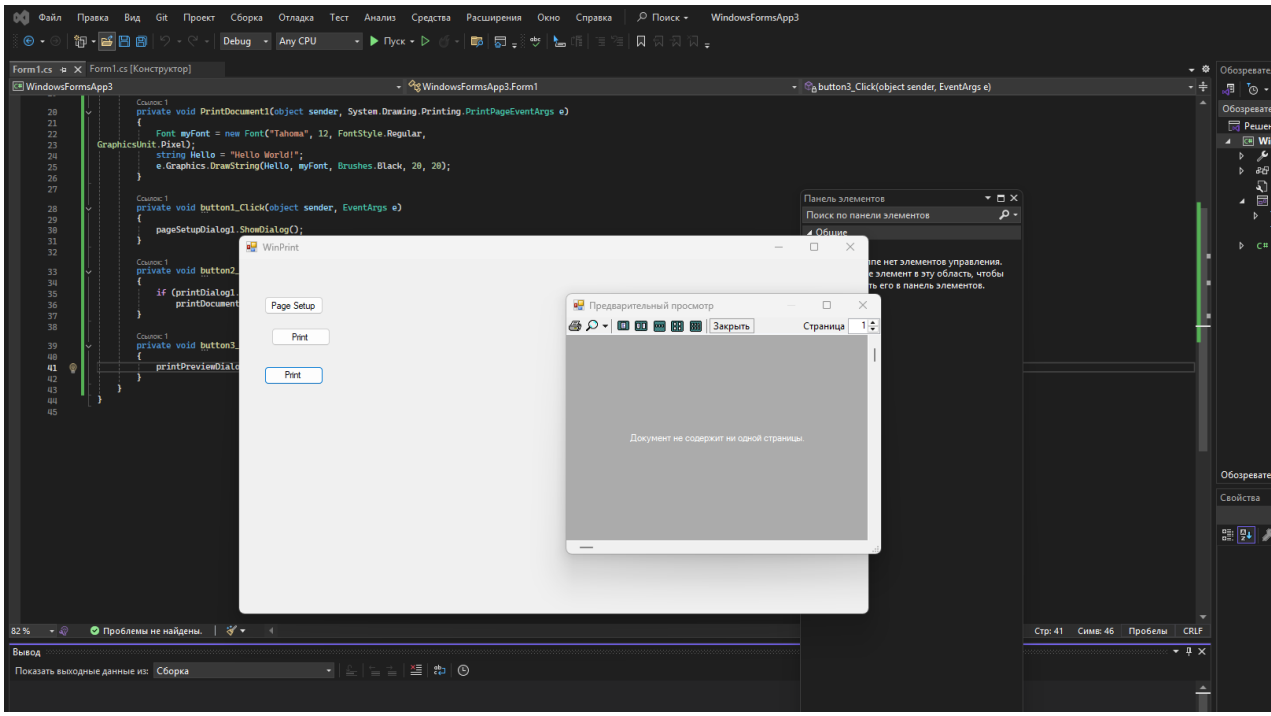

Form4.cs*
Form4.cs [Конструктор]*
WindowsFormsApp10
WindowsFormsApp10.Form4
openFileDialog1_FileOk(object sender, C
}
}
Ссылок: 1
private void FormatToolStripMenuItem_Click(object sender, EventArgs e)
{
}
Ссылок: 1
private void шрифтToolStripMenuItem_Click(object sender, EventArgs e)
{
if (fontDialog1.ShowDialog() == DialogResult.OK)
{
richTextBox1.Font = fontDialog1.Font;
}
}
Ссылок: 1
private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
{
Stream myStream = null;
OpenFileDialog openFileDialog1 = new OpenFileDialog();
openFileDialog1.InitialDirectory = @"c:\";
openFileDialog1.Filter = "txt files (*.txt)|*.txt|Allfiles(*.*) | *.* ";
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 %
Проблемы не найдены.
Стр: 84
Вывод

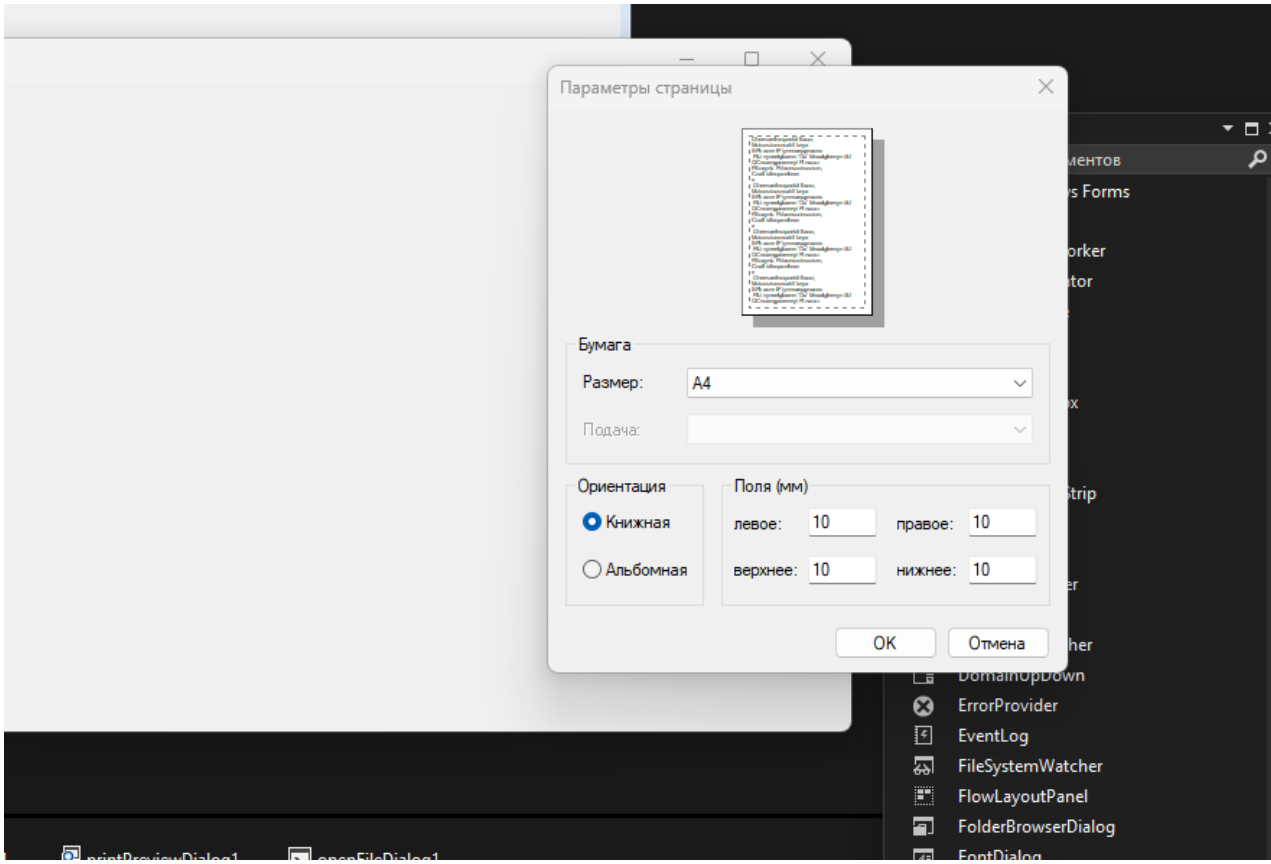
Лабораторная работа 6 Организация печати в формах windows

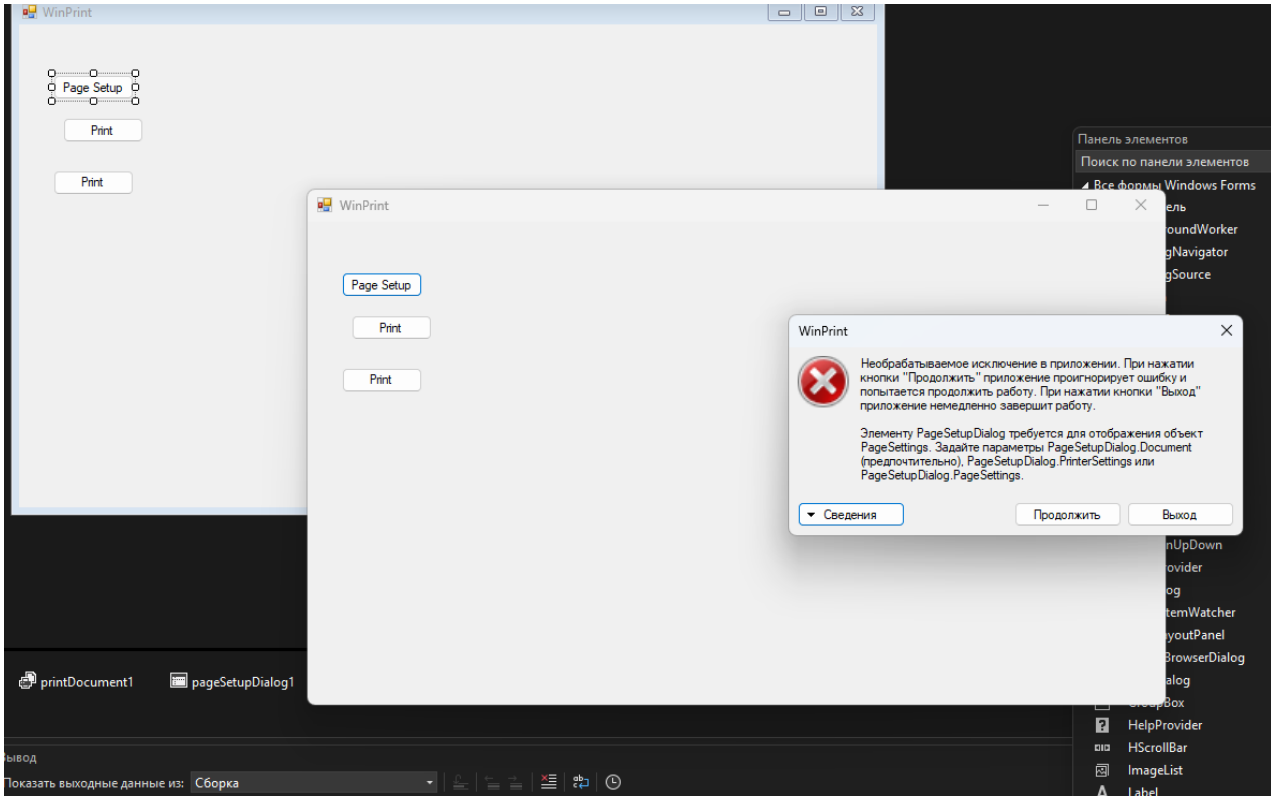
Цель работы

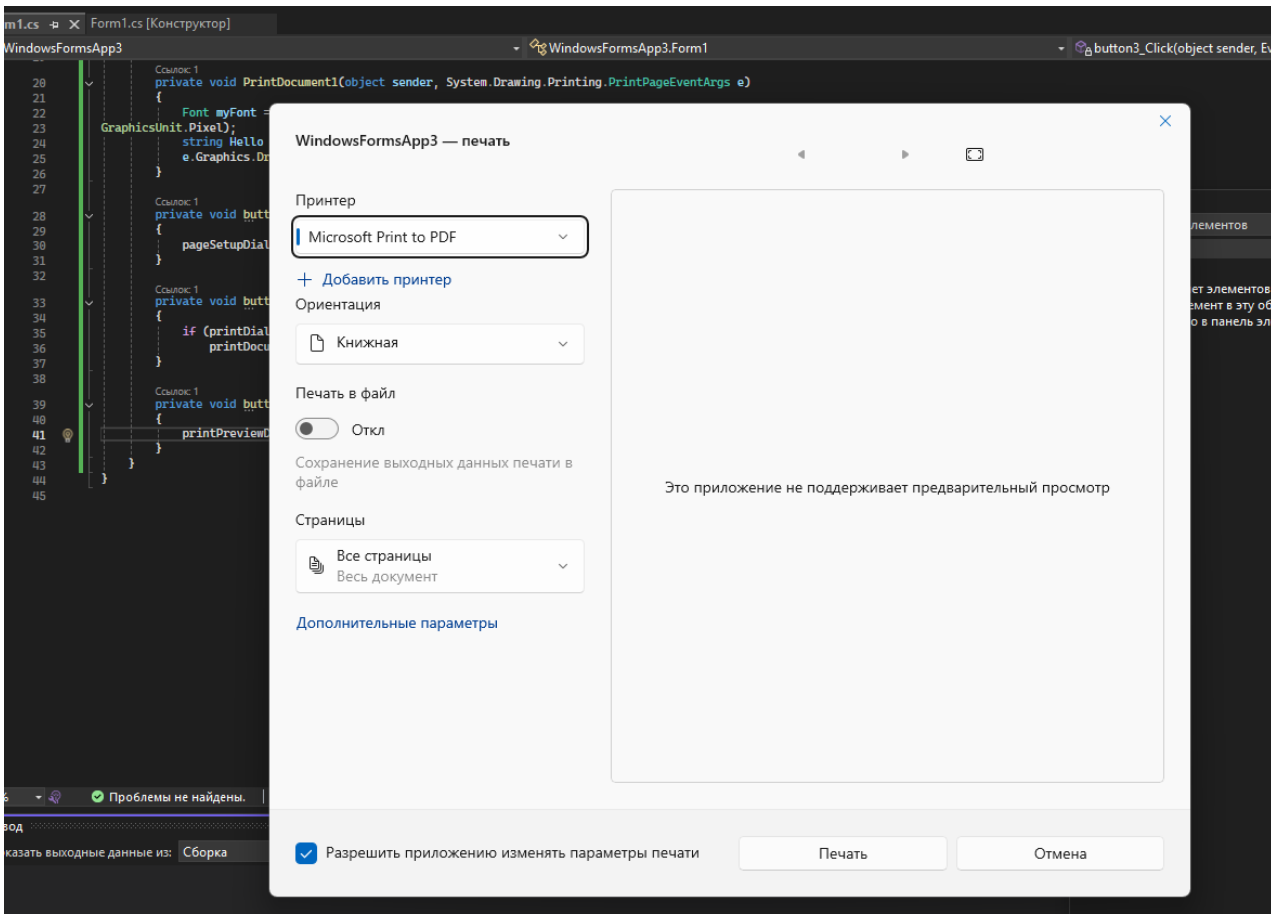
Изучение классов, реализующих задачу программирования печати и получение навыков по работе в программе с диалоговыми окнами.

Упражнение 1 Использование диалоговых окон для печати

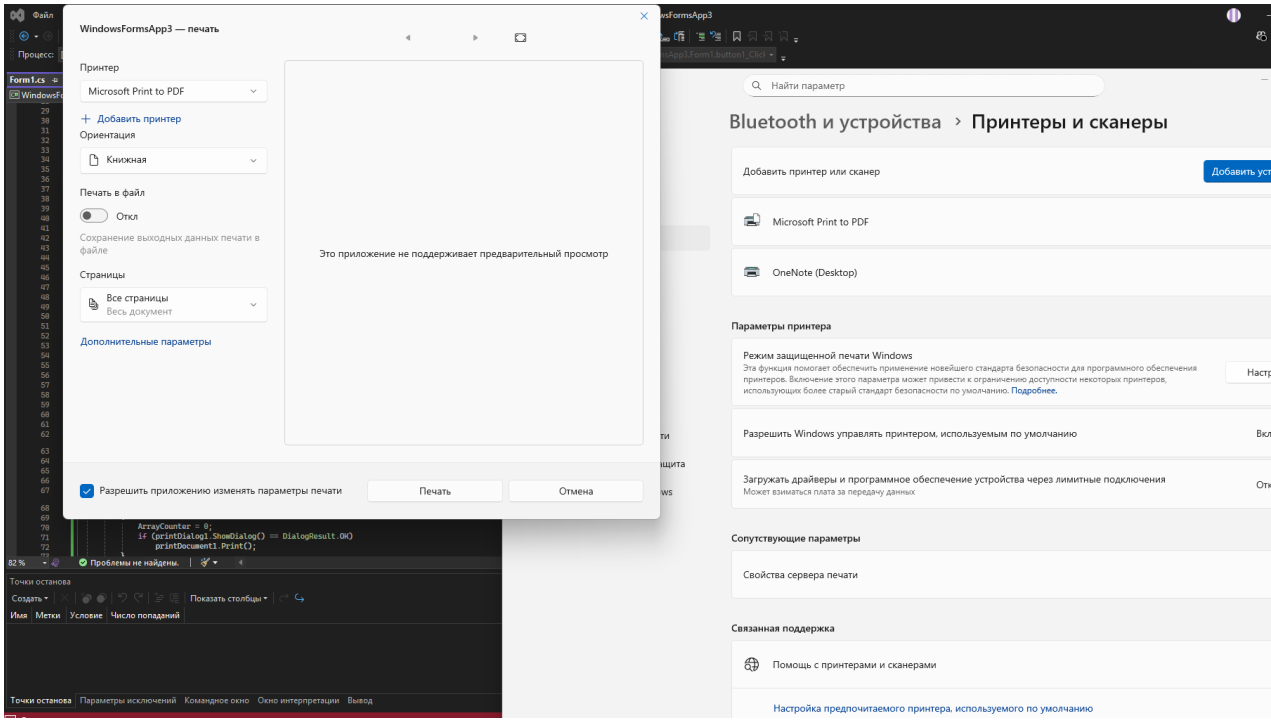


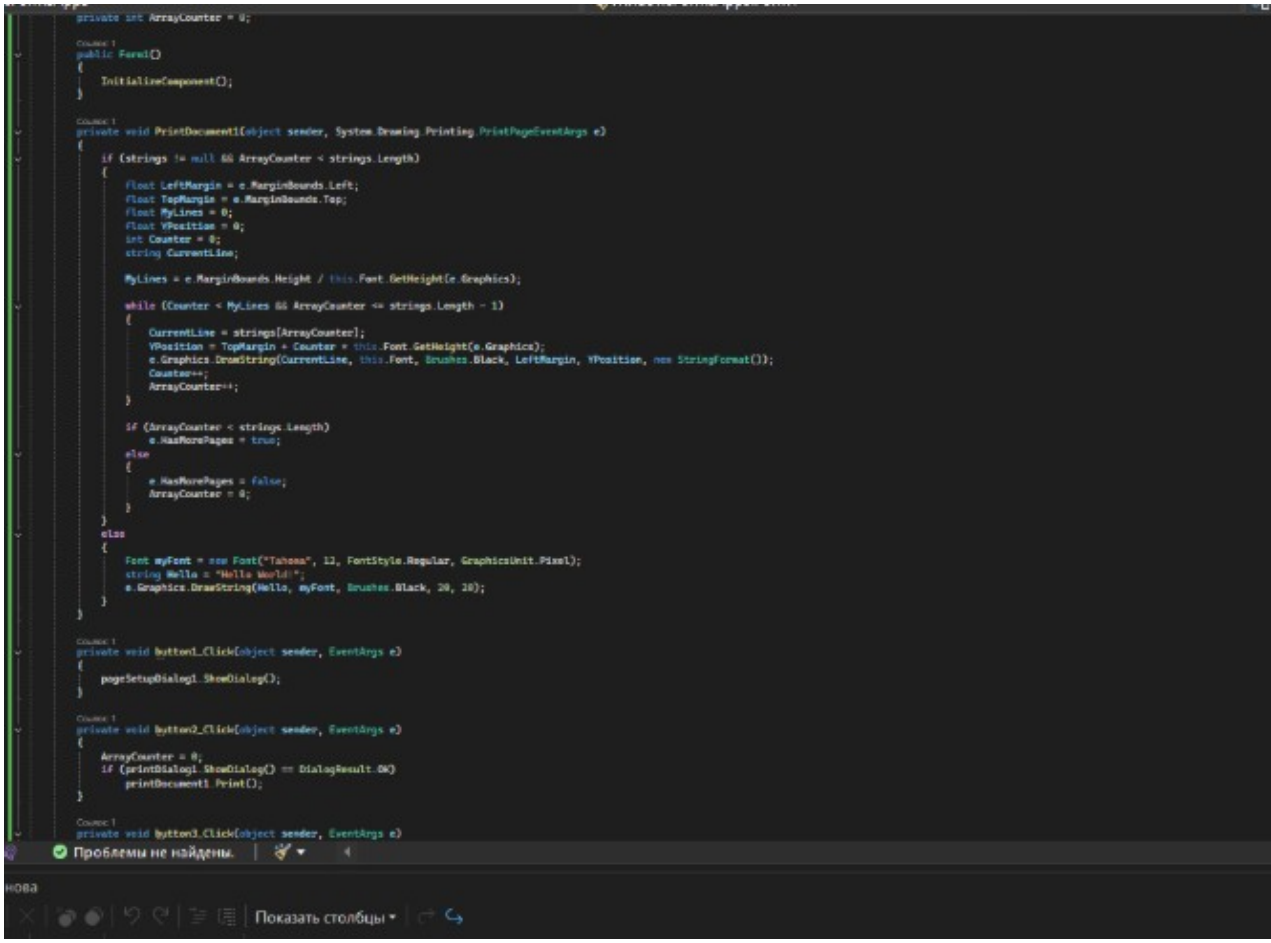

Параметры страницы
Бумага
Размер:
A4
Подача:
Ориентация
Книжная
Альбомная
Поля (мм)
левое:
10
правое:
10
верхнее:
10
нижнее:
10
OK
Отмена
ErrorProvider
EventLog
FileSystemWatcher
FlowLayoutPanel
FolderBrowserDialog


WinPrint
Page Setup
Print
Print
WinPrint
Page Setup
Print
Print
WinPrint
Необрабатываемое исключение в приложении. При нажатии кнопки "Продолжить" приложение проигнорирует ошибку и попытается продолжить работу. При нажатии кнопки "Выход" приложение немедленно завершит работу.
Элементу PageSetupDialog требуется для отображения объект PageSettings. Задайте параметры PageSetupDialog.Document (предпочтительно), PageSetupDialog.PrinterSettings или PageSetupDialog.PageSettings.
Сведения
Продолжить
Выход
printDocument1
pageSetupDialog1
Показать выходные данные из:
Сборка
Панель элементов
Поиск по панели элементов
HelpProvider
HScrollBar
ImageList



Упражнение 2 Создание документа печати


WindowsFormsApp3 — печать
Принтер
Microsoft Print to PDF
Добавить принтер
Ориентация
Книжная
Печать в файл
Откл
Сохранение выходных данных печати в файле
Страницы
Все страницы
Весь документ
Дополнительные параметры
Это приложение не поддерживает предварительный просмотр
Разрешить приложению изменять параметры печати
Печать
Отмена
Bluetooth и устройства › Принтеры и сканеры
Добавить принтер или сканер
Microsoft Print to PDF
OneNote (Desktop)
Параметры принтера
Режим защищенной печати Windows
Разрешить Windows управлять принтером, используемым по умолчанию
Загружать драйверы и программное обеспечение устройства через лимитные подключения
Сопутствующие параметры
Свойства сервера печати
Связанная поддержка
Помощь с принтерами и сканерами
Настройка предпочитаемого принтера, используемого по умолчанию


Проблемы не найдены.
Показать столбцы

Form1.cs | Form1.cs [Конструктор]

WindowsFormsApp3 | WindowsFormsApp3.Fo

```
            {
                Font myFont = new Font("Tahoma", 12, FontStyle.Regular, GraphicsUnit.Pixel);
                string Hello = "Hello World!";
                e.Graphics.DrawString(Hello, myFont, Brushes.Black, 20, 20);
            }
        }

        Ссылок: 1
        private void button1_Click(object sender, EventArgs e)
        {
            pageSetupDialog1.ShowDialog();
        }

        Ссылок: 1
        private void button2_Click(object sender, EventArgs e)
        {
            ArrayCounter = 0;
            if (printDialog1.ShowDialog() == DialogResult.OK)
                printDocument1.Print();
        }

        Ссылок: 1
        private void button3_Click(object sender, EventArgs e)
        {
            ArrayCounter = 0;
            printPreviewDialog1.ShowDialog();
        }

        Ссылок: 0
        private void button4_Click(object sender, EventArgs e)
        {
            DialogResult aResult = openFileDialog1.ShowDialog();

            if (aResult == DialogResult.OK)
            {
                try
                {
                    StreamReader aReader = new StreamReader(openFileDialog1.FileName);
                    s = aReader.ReadToEnd();
                    aReader.Close();
                    strings = s.Split('\n');
                    ArrayCounter = 0;

                    MessageBox.Show("Файл загружен успешно!", "Информация",
                        MessageBoxButtons.OK, MessageBoxIcon.Information);
                }
                catch (Exception ex)
                {
                    MessageBox.Show("Ошибка при чтении файла: " + ex.Message,
                        "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
        }

        Ссылок: 1
        private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
        {
        }
    }
}
```

Упражнение 3

Создание специализированной формы предварительного