

Лекція 9

ОСНОВИ WINDOWS FORMS



Лекція 9. Основи Windows Forms

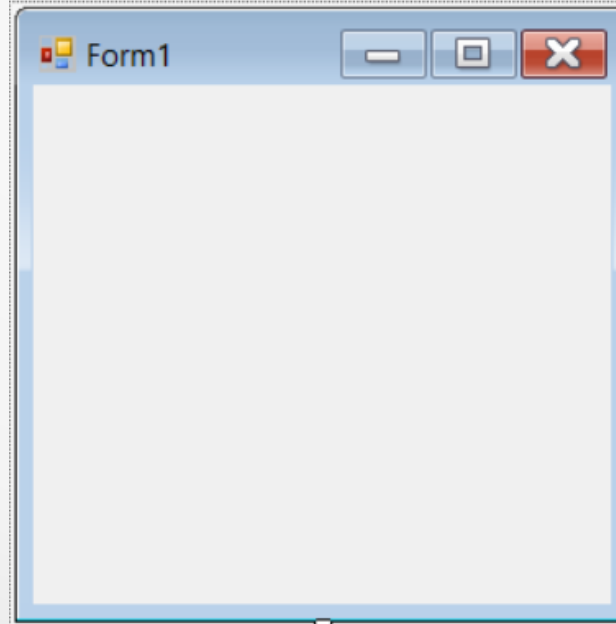
План

1. Створення програми з GUI.
2. Основи роботи з формами.
3. Приклад створення програми.

1. Створення програми з GUI

Для створення програм з графічним інтерфейсом користувача (**GUI** – Graphical User Interface) у платформі .NET Framework найчастіше використовується технологія **Windows Forms**.

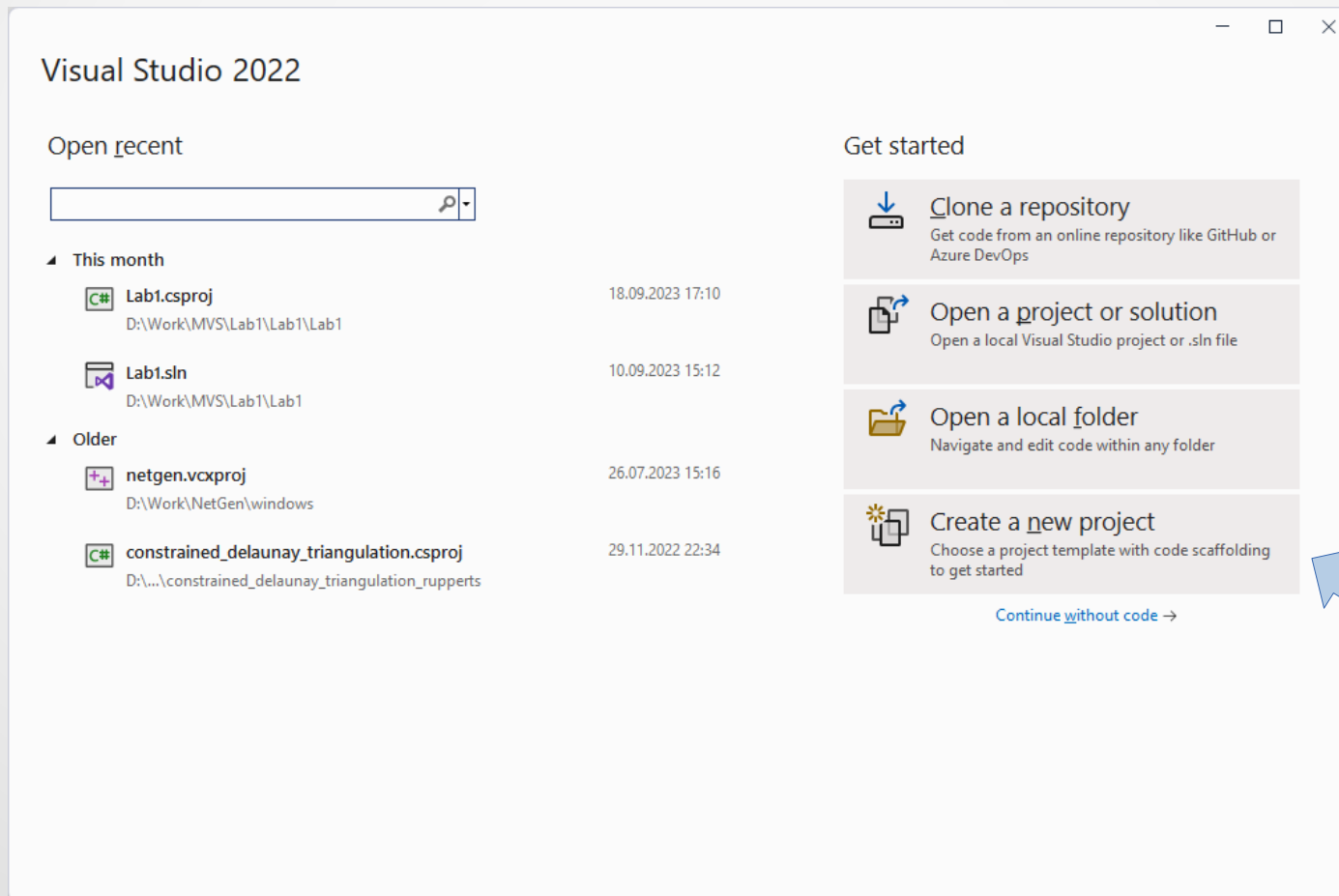
У Windows Forms термін «**Форма**» є синонімом **вікна** (головного вікна програми, діалогового вікна і т.п.).



Приклад форми

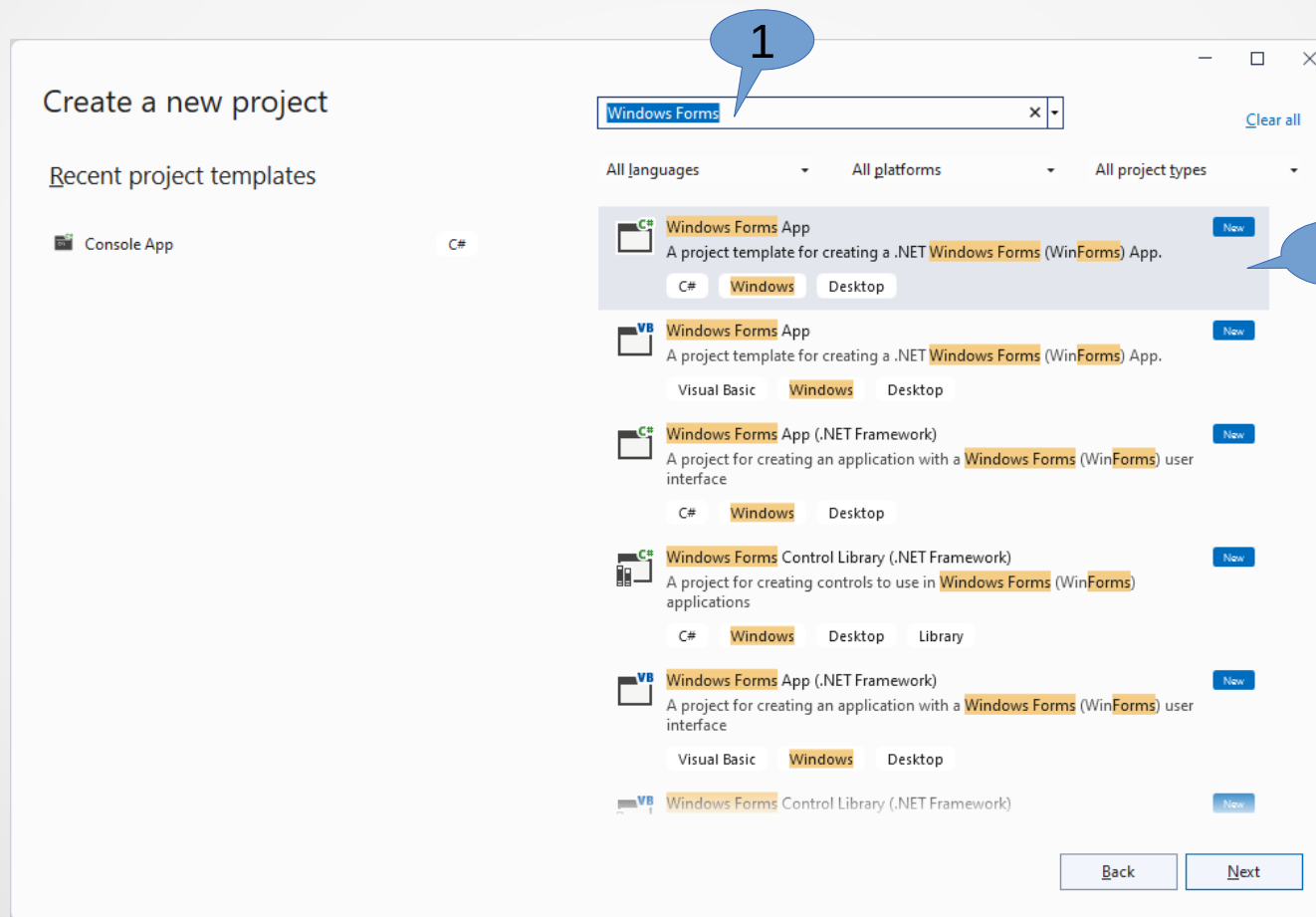
1. Створення програми з GUI

Для створення проекту **Windows Forms** у стартовому вікні **Microsoft Visual Studio** потрібно вибрати пункт «**Create a new project**»



1. Створення програми з GUI

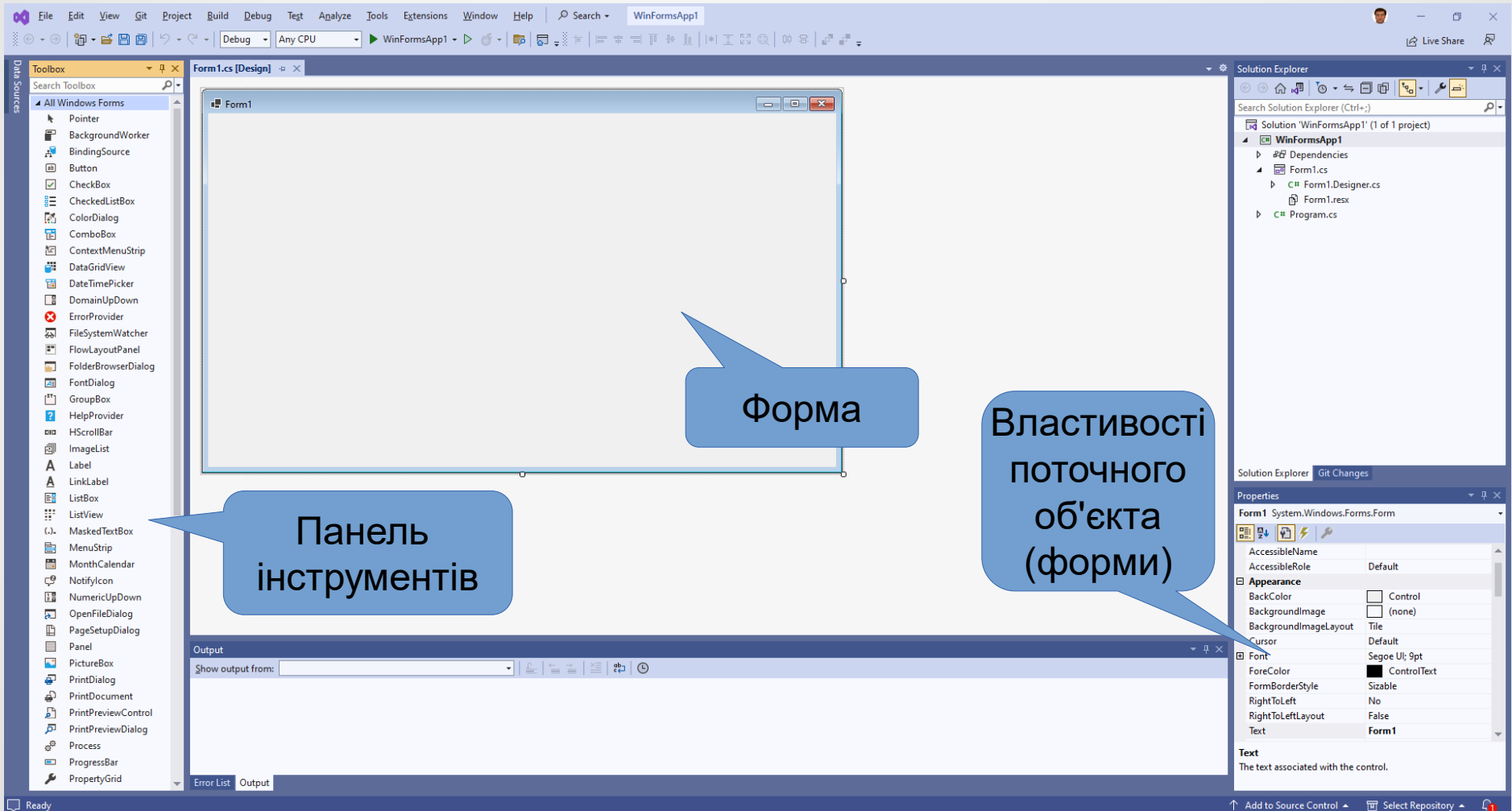
Потім у вікні «Create a new project» у рядок пошуку ввести «Windows Forms», після чого вибрати проект «Windows Forms App» для C#.



Важливо. Слід вибирати проект .NET Framework, а не .NET Core!

1. Створення програми з GUI

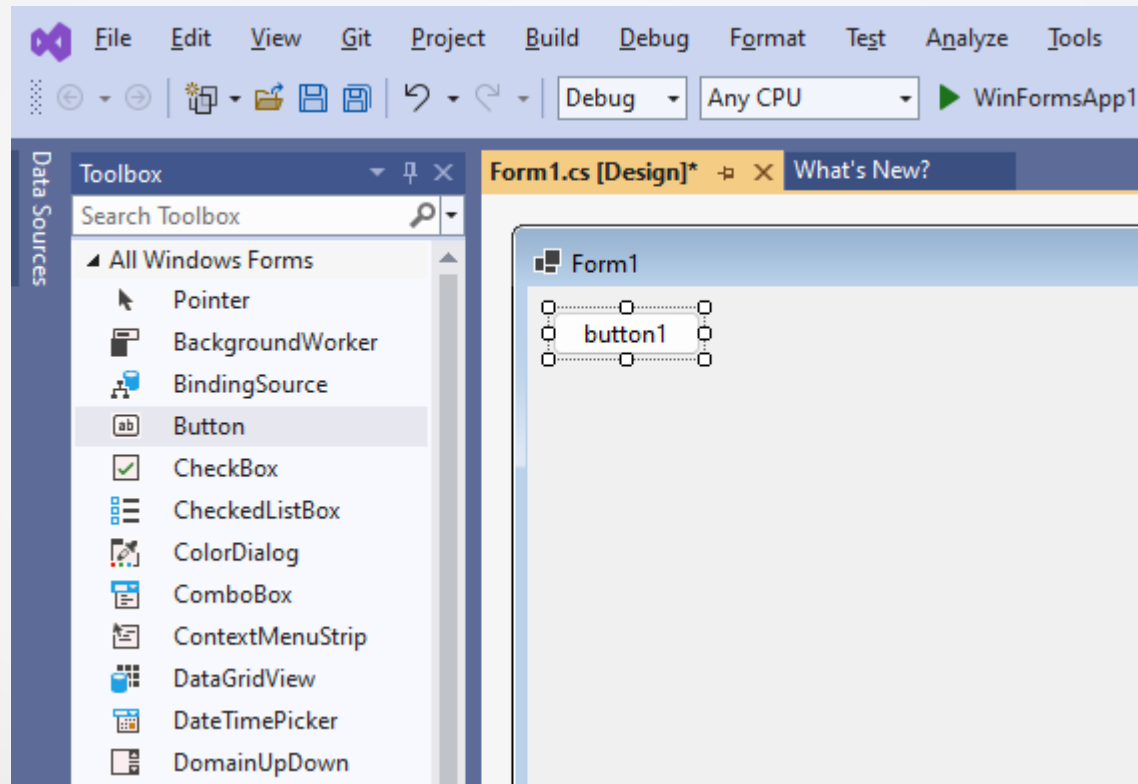
Далі, вказавши назву та місце розташування проекту, отримаємо:



Примітка. Для відображення панелі інструментів потрібно виконати команду головного меню «View|Toolbox»

2. Основи роботи з формами

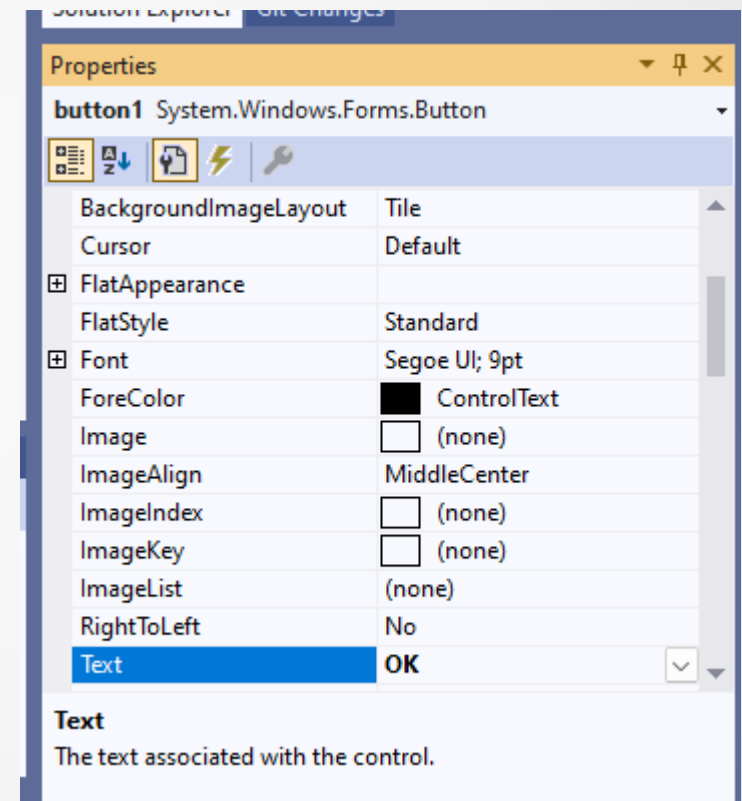
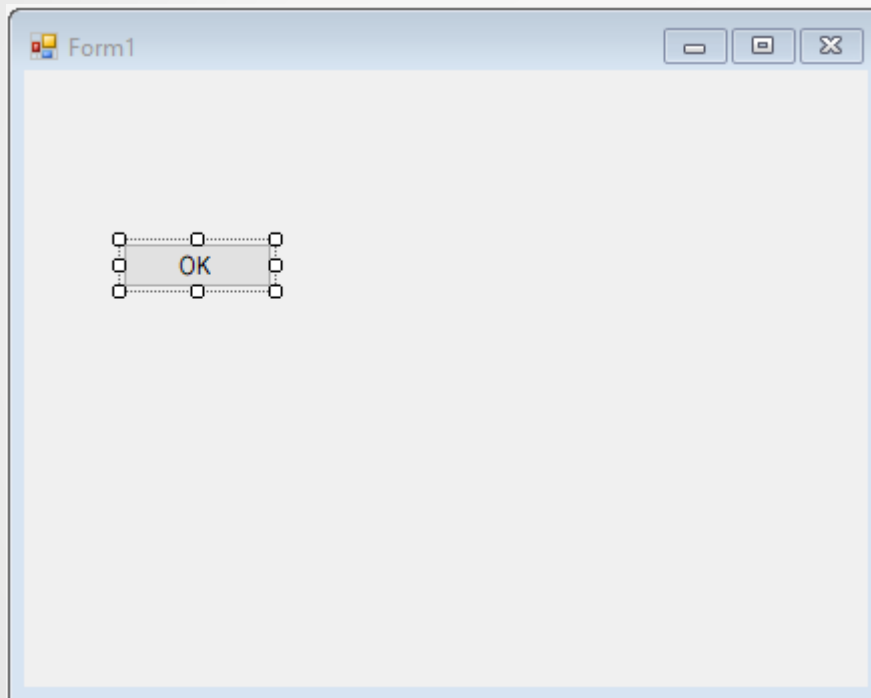
Для додавання нового компонента на форму його потрібно вибрати у вікні **Toolbox** і перетягнути на форму.



Додавання кнопки на форму

2. Основи роботи з формами

Для налаштування компонента його потрібно виділити (клікнувши мишкою), після чого внести відповідні правки до його властивостей у вікні «**Properties**». Наприклад, властивість **Text** містить назву кнопки.

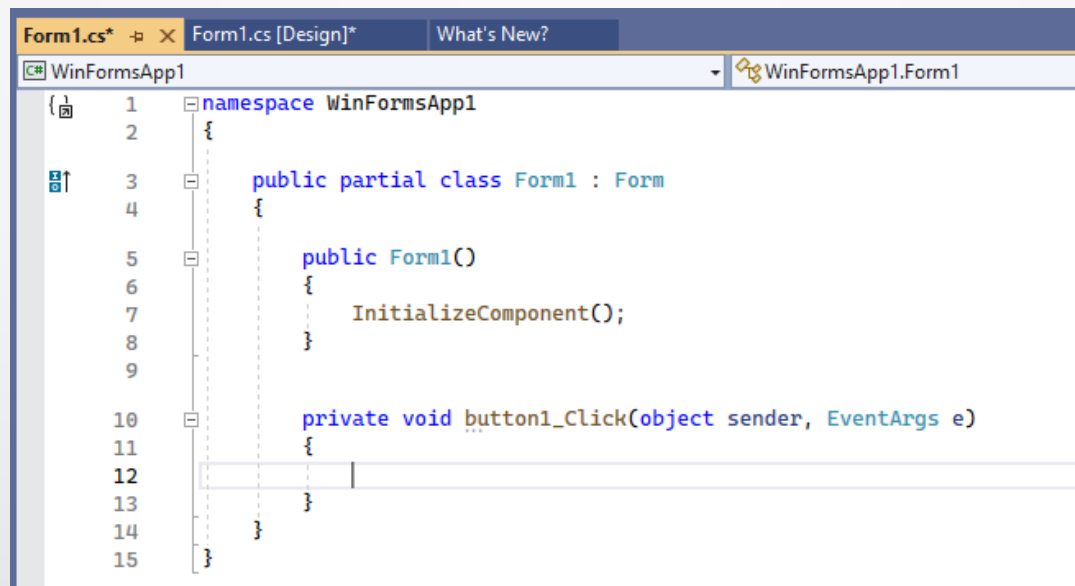


Примітка. За замовчуванням вікно «**Properties**» містить властивості поточної форми.

2. Основи роботи з формами

Крім **властивостей** і **методів** кожен об'єкт **GUI** має низку **подій** – спеціальних методів, що автоматично викликаються для обробки тієї чи іншої події, яка виникає в процесі роботи об'єкта.

Наприклад, клас **System.Windows.Forms.Button**, що описує кнопку, має подію **Click**, яка викликається при натисканні на кнопку. Для додавання до коду обробника події **Click** потрібно двічі клікнути мишкою на відповідну кнопку, розташовану на формі.

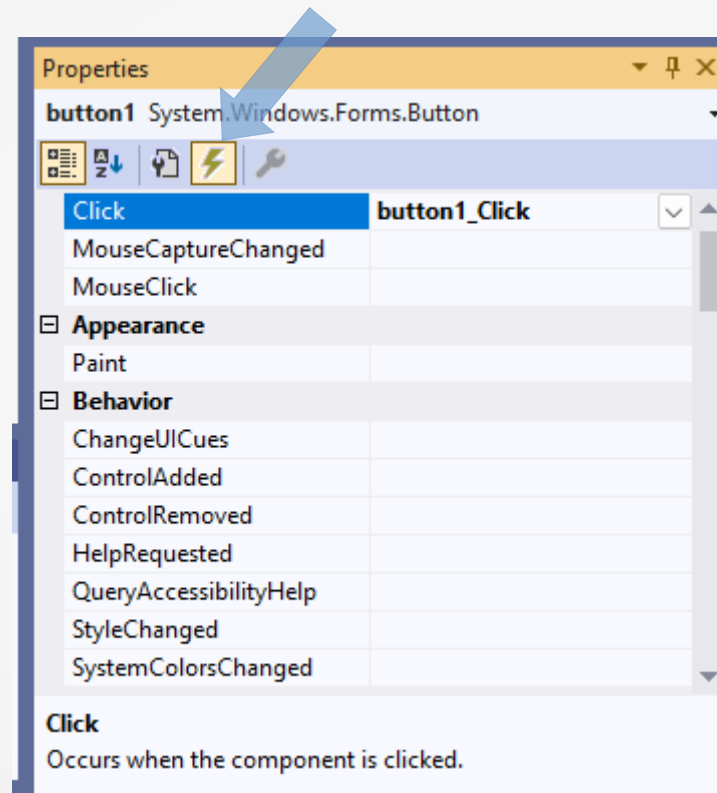


```
1 namespace WinFormsApp1
2 {
3     public partial class Form1 : Form
4     {
5         public Form1()
6         {
7             InitializeComponent();
8         }
9
10        private void button1_Click(object sender, EventArgs e)
11        {
12        }
13    }
14 }
15
```

Обробник події Click

2. Основи роботи з формами

Список доступних **подій** поточного об'єкта можна побачити у вікні «**Properties**», натиснувши кнопку «**Events**».



Список доступних подій

2. Основи роботи з формами

Таблиця 1. Основні характеристики класу Form

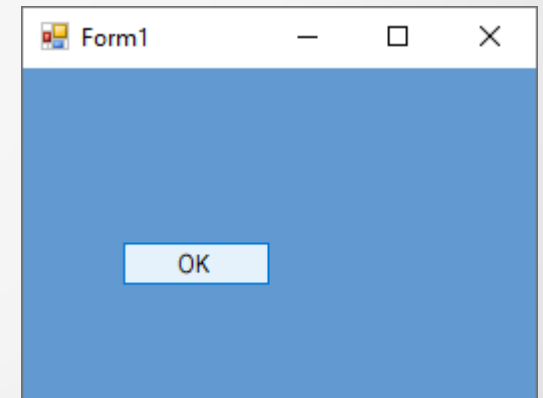
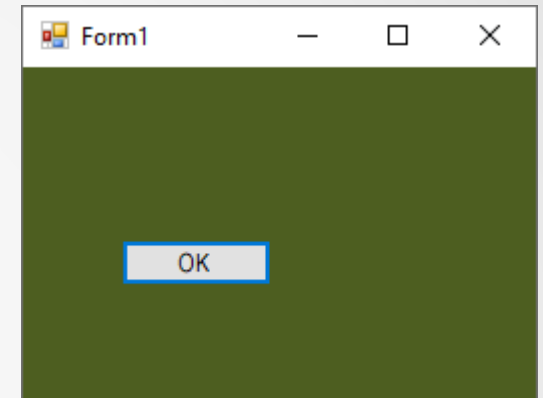
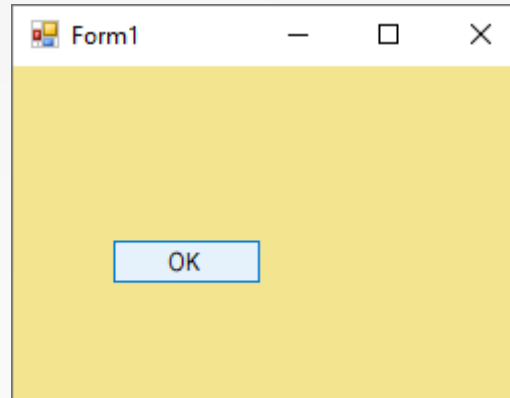
Властивість	Опис
Name	Визначає ім'я об'єкта
BackColor	Колір робочої області форми
ControlBox	Визначає, чи відображається системне меню форми
Cursor	Задає курсор форми
Enabled	Визначає доступність форми (False – форма заблокована)
Font	Задає шрифт форми та всіх її компонентів
FormBorderStyle	Вказує, як буде відображатися границя форми та рядок заголовка
Opacity	Задає прозорість форми
Size	Визначає початковий розмір форми
StartPosition	Задає початкову позицію форми на екрані
Text	Визначає заголовок форми
Visible	Визначає видимість форми
WindowState	Задає стан форми під час запуску (нормальний, згорнутий або розгорнутий)

2. Основи роботи з формами

Приклад зміни кольору фону форми під час натискання кнопки.

```
using System;
using System.Drawing;
using System.Windows.Forms;

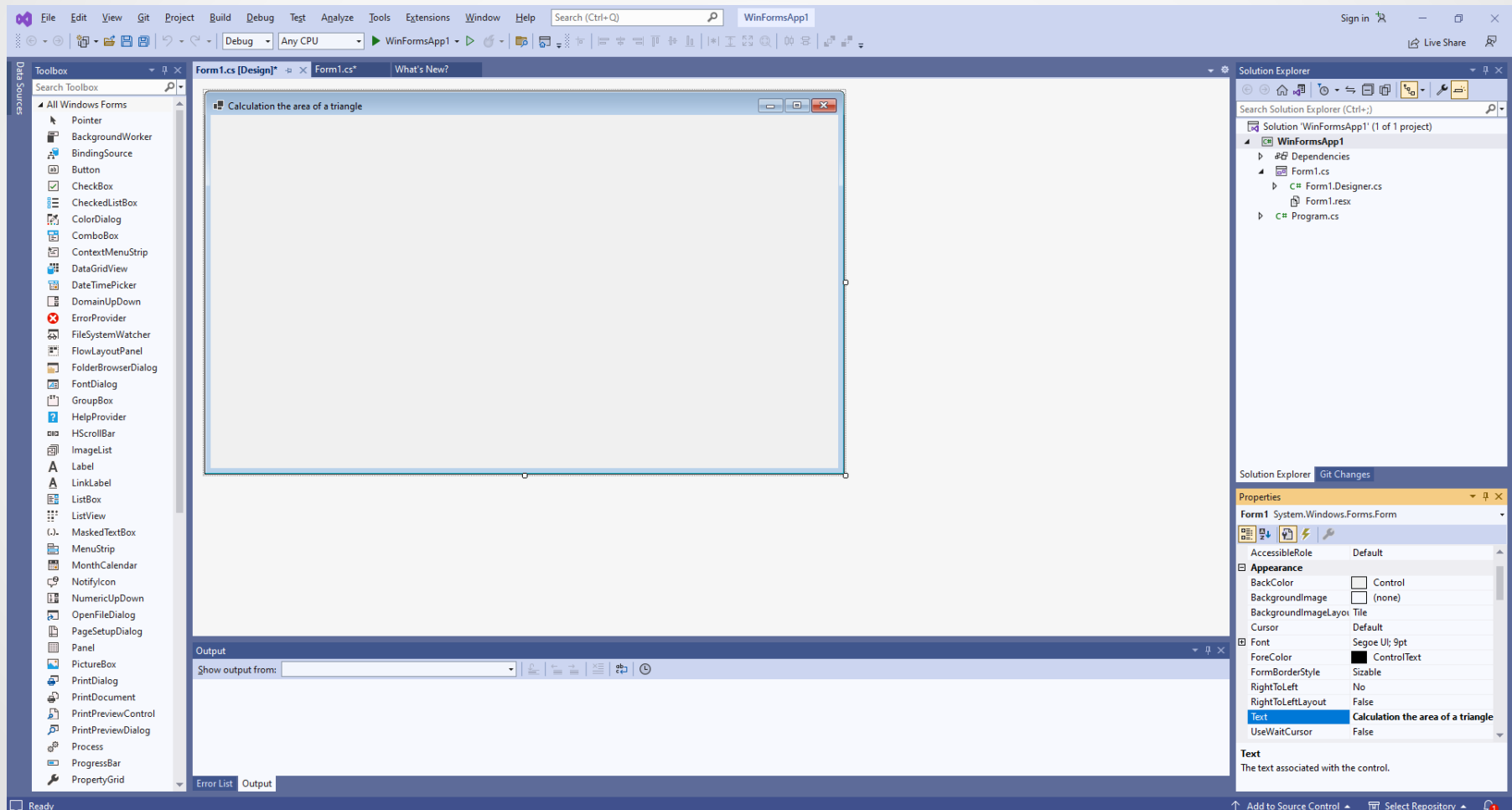
namespace WindowsFormsDBTest
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            Random rnd = new Random();
            int red = rnd.Next(0, 255),
                green = rnd.Next(0, 255),
                blue = rnd.Next(0, 255);
            this.BackColor = Color.FromArgb(red, green, blue);
        }
    }
}
```



3. Приклад створення програми

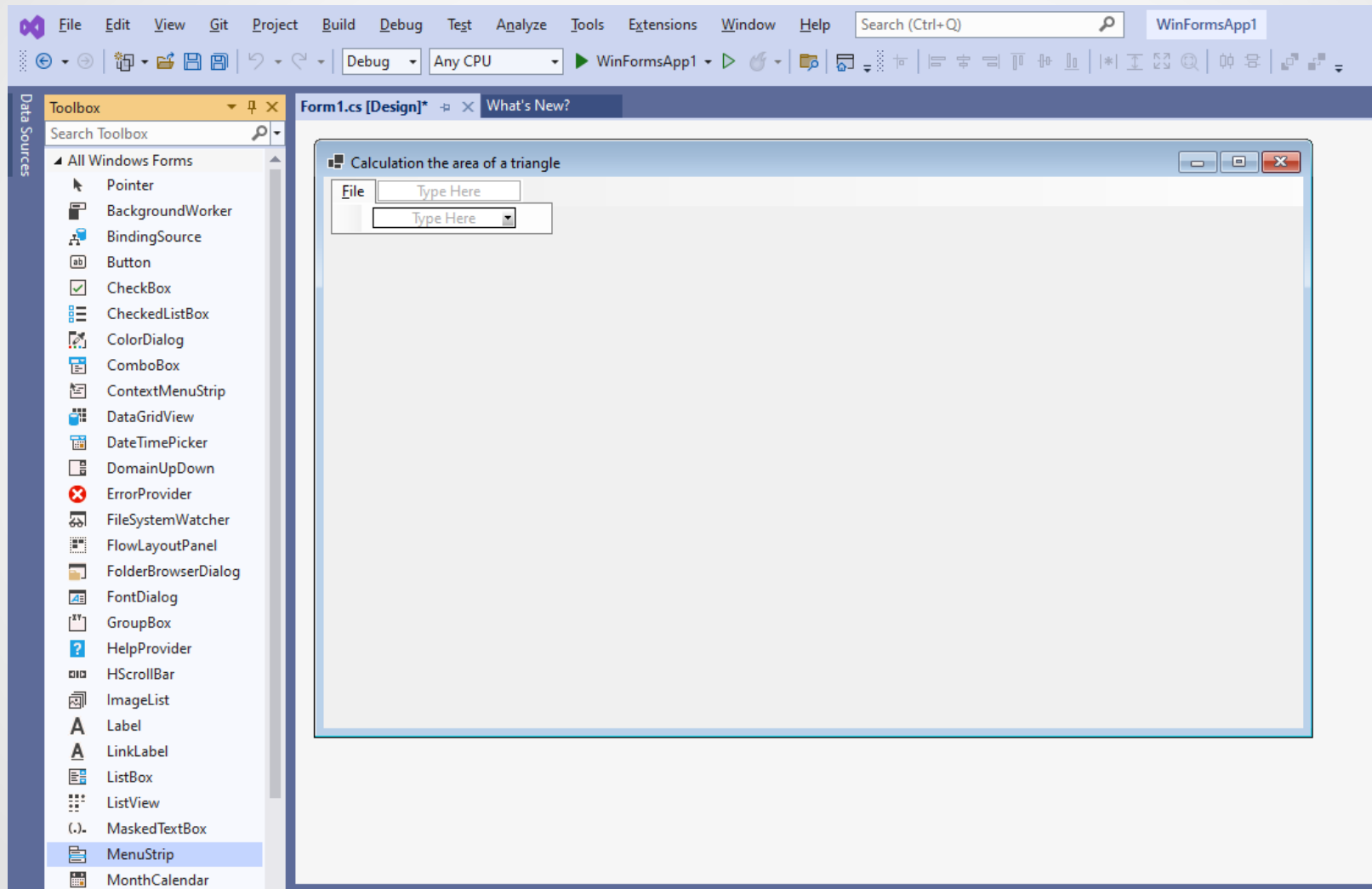
Розглянемо приклад створення програми з GUI, яка виконує обчислення площі трикутника за формулою Герона.

1. Створимо новий проект і змінимо назву форми.



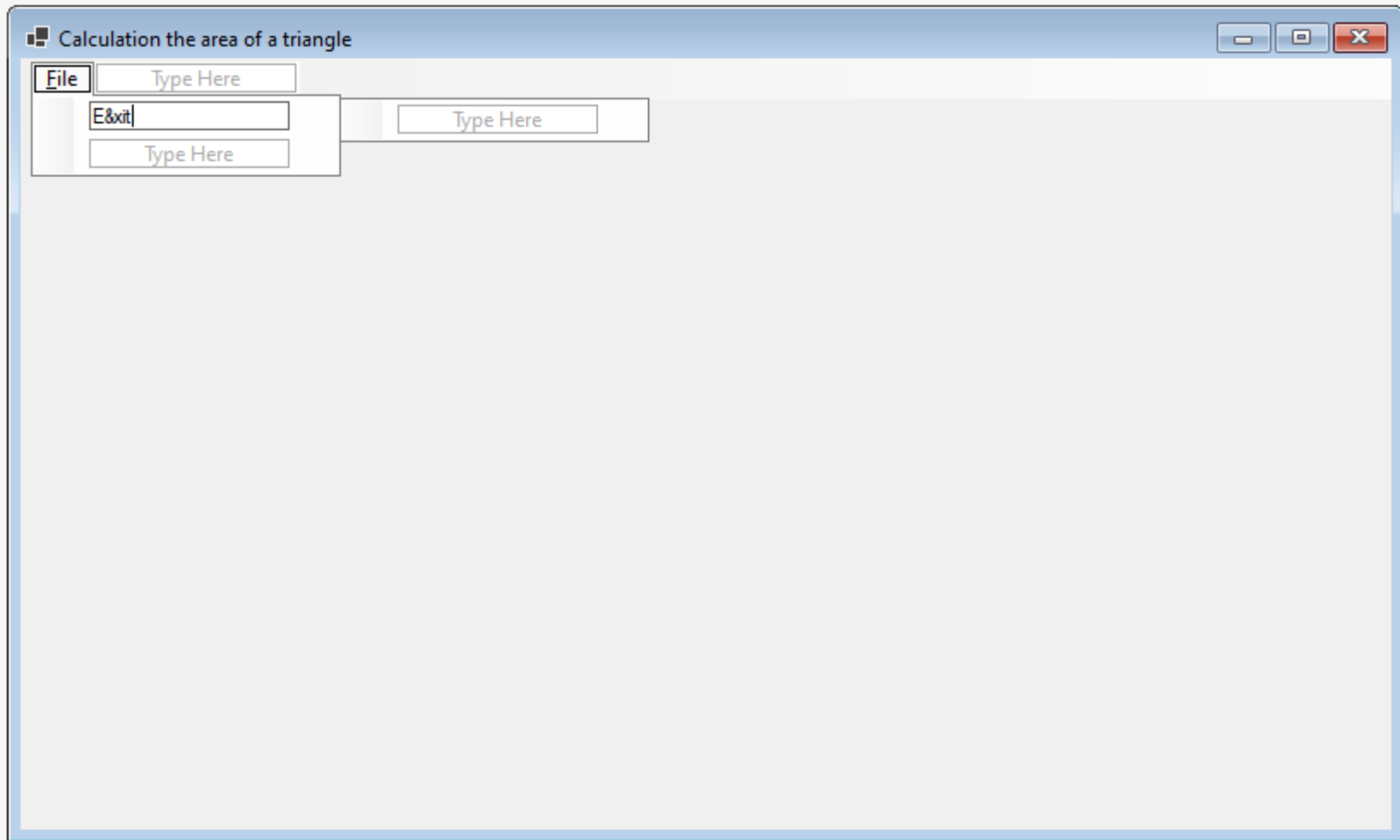
3. Приклад створення програми

2. Додамо на форму головне меню.



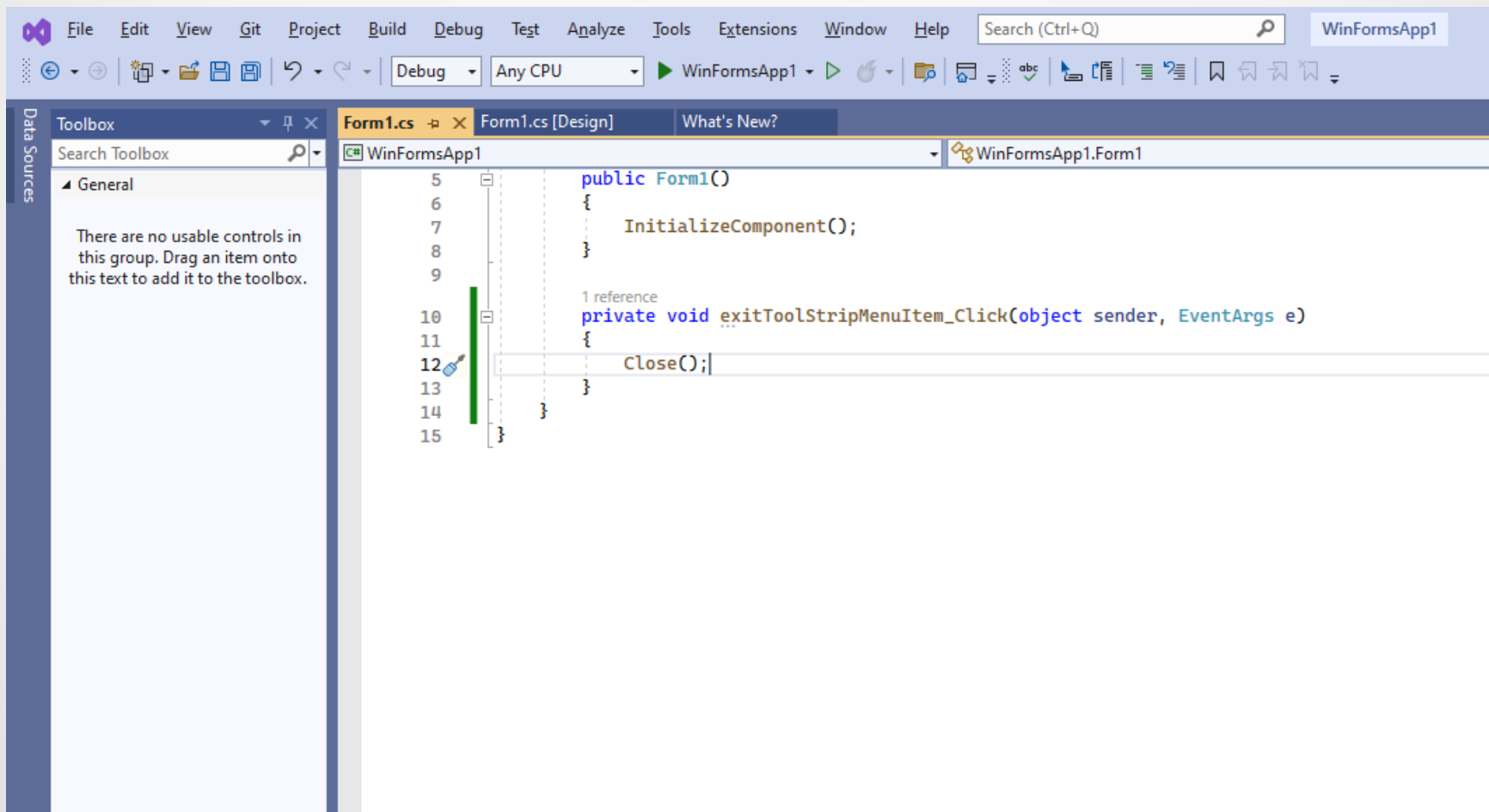
3. Приклад створення програми

3. Налаштуємо головне меню, додавши команду «File|Exit»



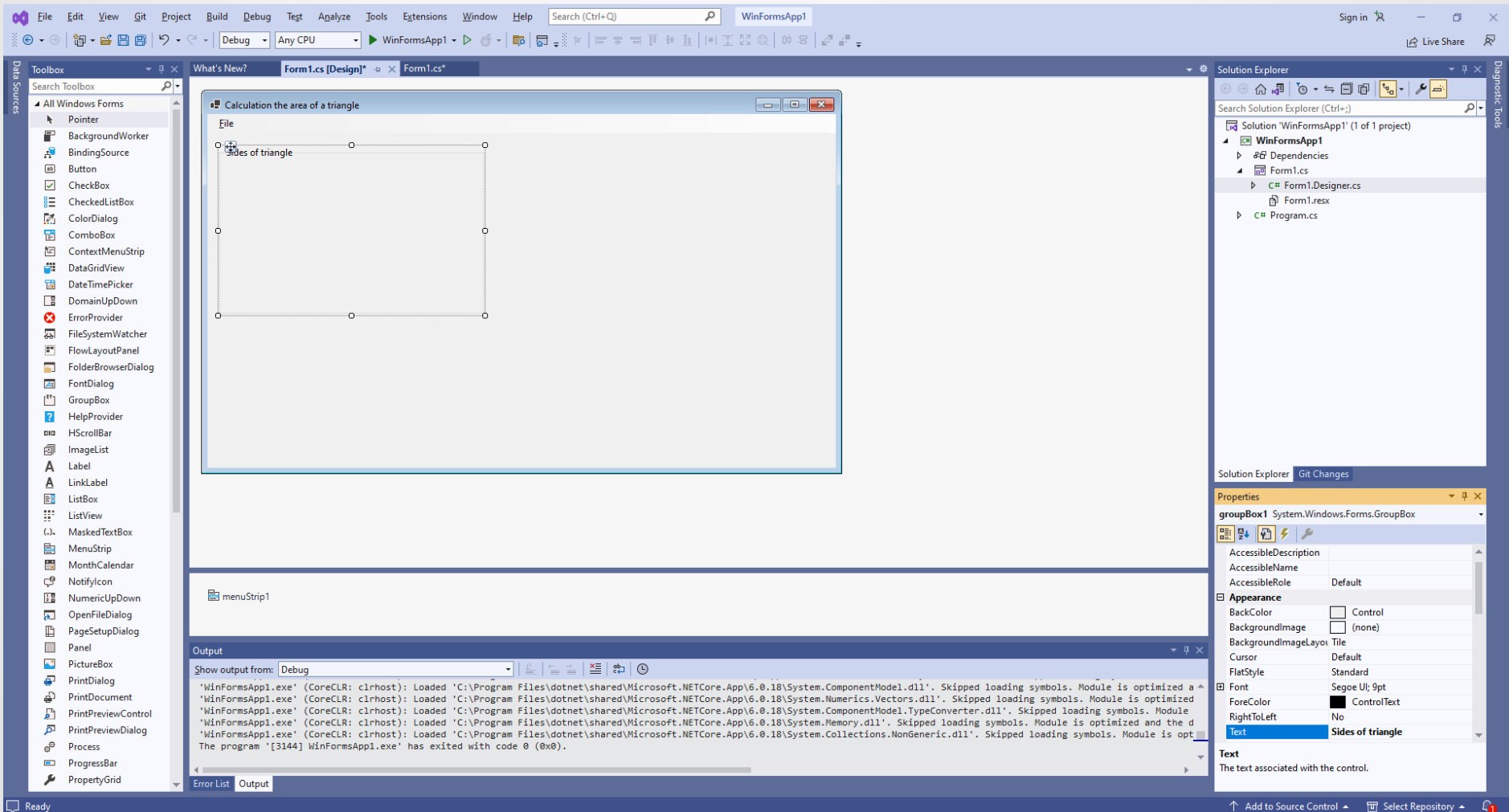
3. Приклад створення програми

4. Додамо **обробник події** – вибір команди «File|Exit». Для цього двічі клацнемо на відповідний пункт меню. В обробнику цієї події викличемо метод **Close()** класу **Form**.



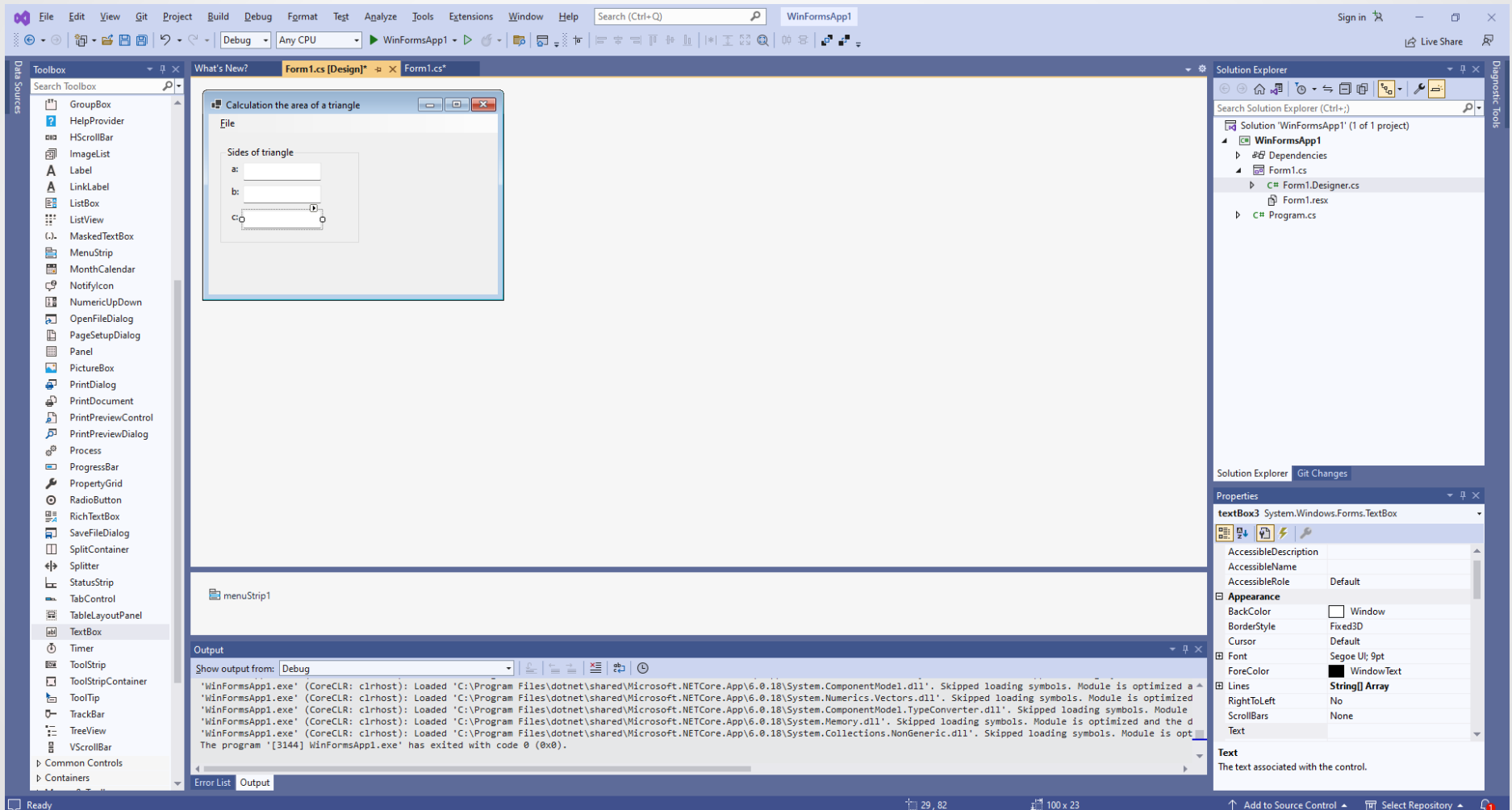
3. Приклад створення програми

5. Додамо на форму компонент GroupBox та налаштуємо його наступним чином.



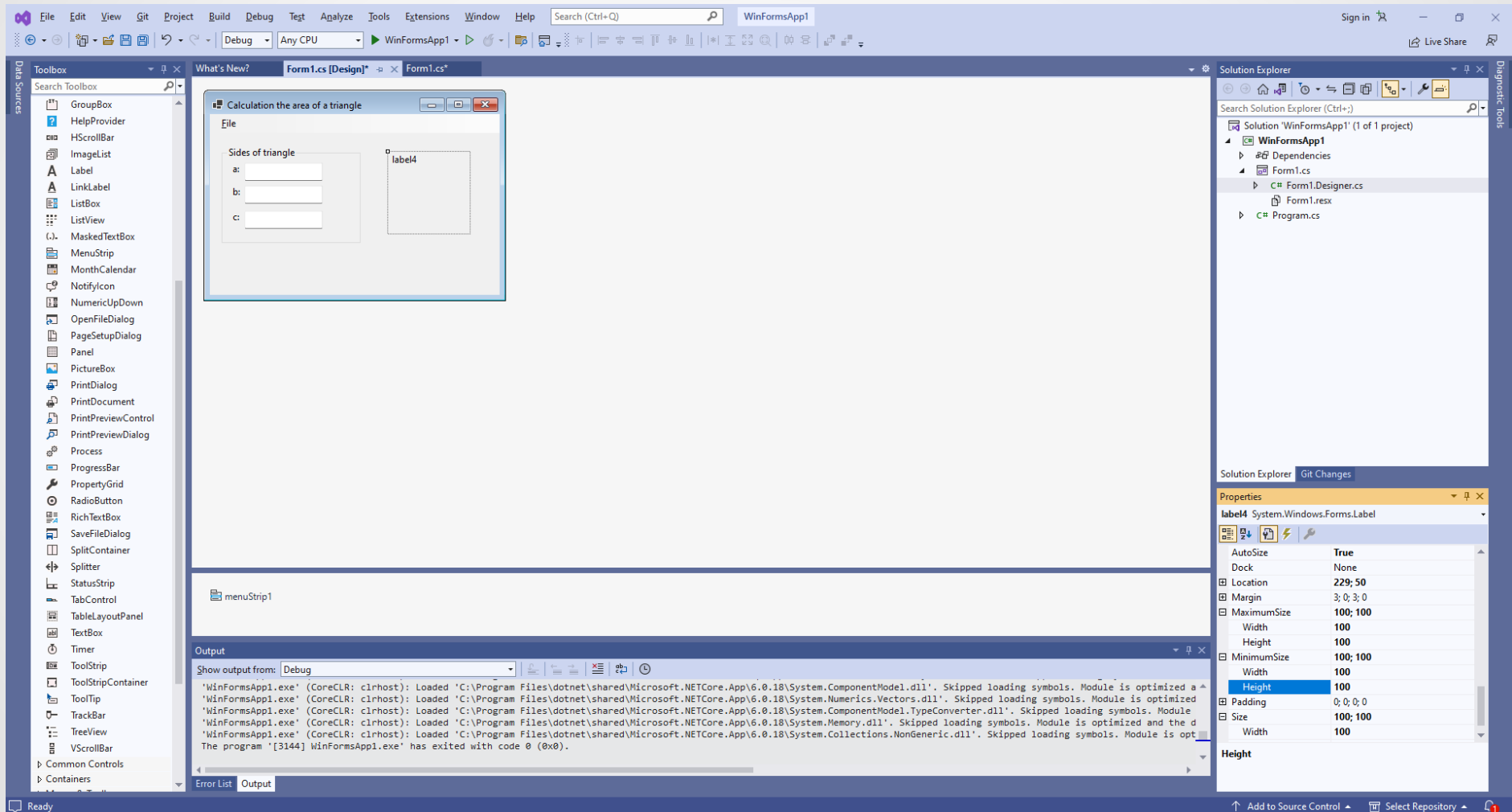
3. Приклад створення програми

5. Додамо в контейнер groupBox1 три компоненти **Label** (напис) і стільки ж полів введення тексту **TextBox**.



3. Приклад створення програми

5. Додамо форму ще один компонент Label для виведення відповіді.



3. Приклад створення програми

6. Додамо на форму кнопку «Calculate» і в обробнику події Click реалізуємо потрібний нам функціонал програми.

```
private void button1_Click(object sender, EventArgs e)
{
```

```
    double a, b, c, p, s;
```

```
    try
```

```
    {
```

```
        a = Convert.ToDouble(textBox1.Text);
```

```
        b = Convert.ToDouble(textBox2.Text);
```

```
        c = Convert.ToDouble(textBox3.Text);
```

```
        p = (a + b + c) / 2.0;
```

```
        s = Math.Sqrt(p * (p - a) * (p - b) * (p - c));
```

```
        label4.Text = String.Format("S = {0}", s);
```

```
    }
```

```
    catch (Exception err)
```

```
    {
```

```
        label4.Text = err.Message;
```

```
    }
```

```
}
```

