

ЛАБОРАТОРНА РОБОТА №1

Тема: Ознайомлення з інтерфейсом інтегрованого середовища Microsoft Visual Studio 2010

Мета роботи

Ознайомитися з інтерфейсом інтегрованого середовища Microsoft Visual Studio 2010 на прикладі мови програмування C#.

Постановка завдання

Розглянути основні елементи інтерфейсу, вивчити процедуру створення проекту, палітру компонентів, панель інструментів інтегрованого середовища Microsoft Visual Studio 2010.

Теоретичні відомості

Microsoft Visual Studio — серія продуктів фірми Microsoft, які включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight. Visual Studio побудована в архітектурі, що підтримує можливість використання доповнень (Add-Ins), — плагінів від сторонніх розробників, що дозволяє розширювати можливості середовища розробки.

Visual Studio 2010 був представлений 12 квітня 2010 року. Він підтримує розробку програм з використанням .NET Framework 4.0.

У середовищі Visual Studio можна писати програми на Basic, C++, C# та F# (функційне програмування), але ми поки що будемо вивчати лише C#.

C# (вимовляється Сі-шарп) — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтанутом та Пітером Гольде під егідою Microsoft Research (при фірмі Microsoft). C# — це відповідь Microsoft на Java від фірми Sun. C# успадкувала від Java концепції віртуальної машини (середовище .NET), байт-коду (MSIL) і безпеку вихідного коду програм. Нововведенням C# стала можливість легшої взаємодії, порівняно з мовами-попередниками, з кодом програм, написаних на інших мовах, що є важливим при створенні великих проектів.

Станом на сьогодні C# визначено флагманською мовою корпорації Microsoft, бо вона найповніше використовує нові можливості .NET. Решта мов програмування, хоч і підтримуються, але мають прогалини щодо використання .NET.

Хід роботи

1. Запуск Visual Studio 2010

Запустимо Visual Studio 2010 відкривши меню «Пуск» → «Всі програми» → «Microsoft Visual Studio 2010» та запустивши «Microsoft Visual Studio 2010» (Рис. 1).

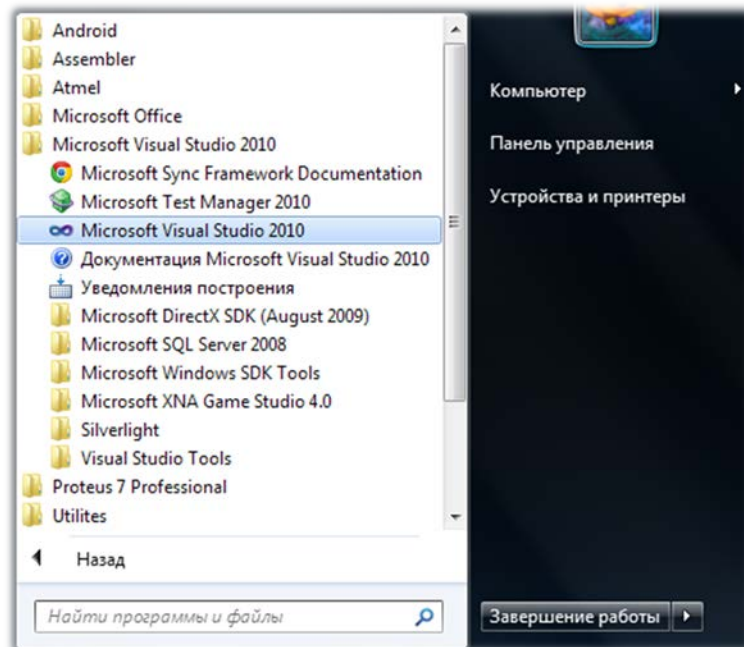


Рис. 1. Меню «Пуск» з вибраним ярликом Visual Studio 2010

Коли програма завантажиться, ми побачимо головне вікно програми з відкритою сторінкою привітання (Рис. 2).

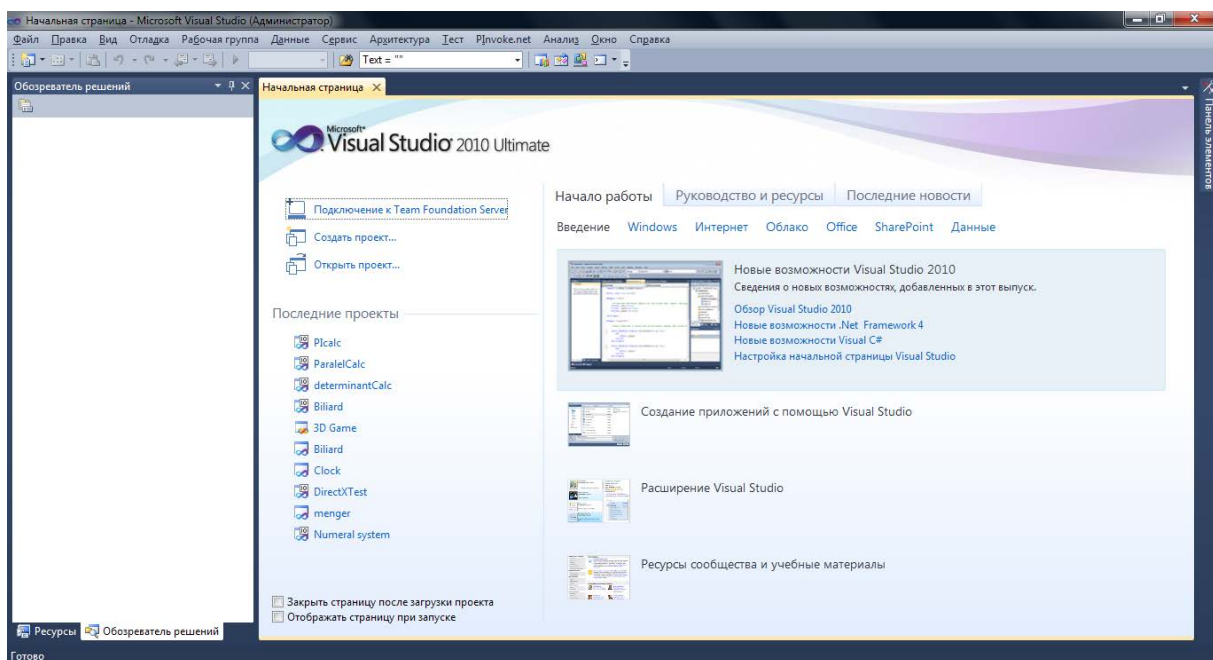


Рис. 2. Головне вікно Visual Studio 2010 з сторінкою привітання

2. Створення нового проекту

Створимо новий проект. Для цього слід вибрати пункт головного меню «Файл» → «Создать» → «Проект» (Рис. 3) або натиснути комбінацію **Ctrl+Shift+N**.

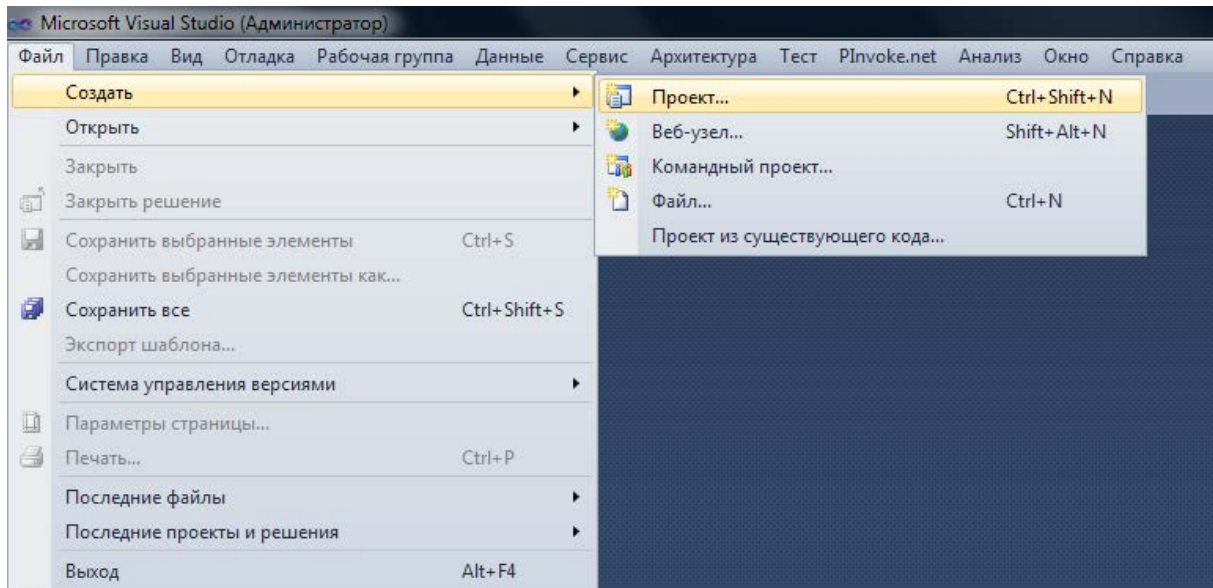


Рис. 3. Створення нового проекту

Далі відкриється вікно вибору типу проекту (Рис. 4).

Вибираємо «Visual C#» → «Приложение Windows Forms».

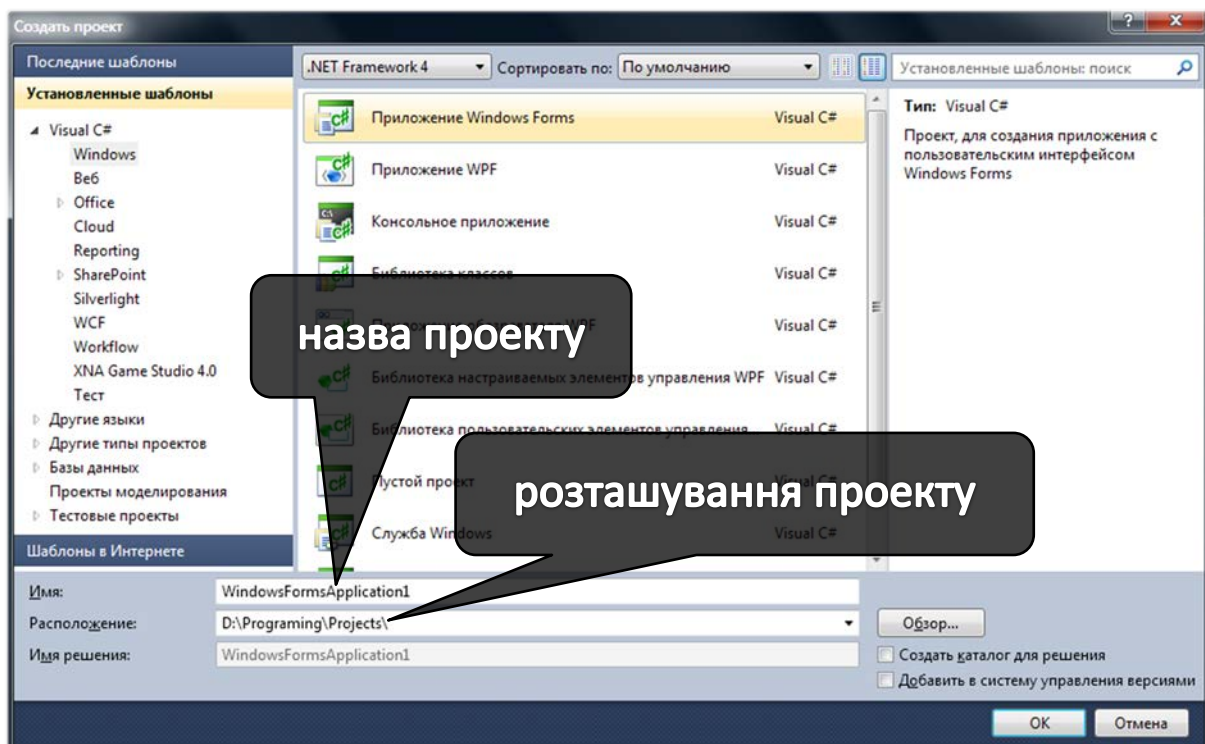


Рис. 4. Вибір типу проекту

3. Головне вікно програми після створення проекту

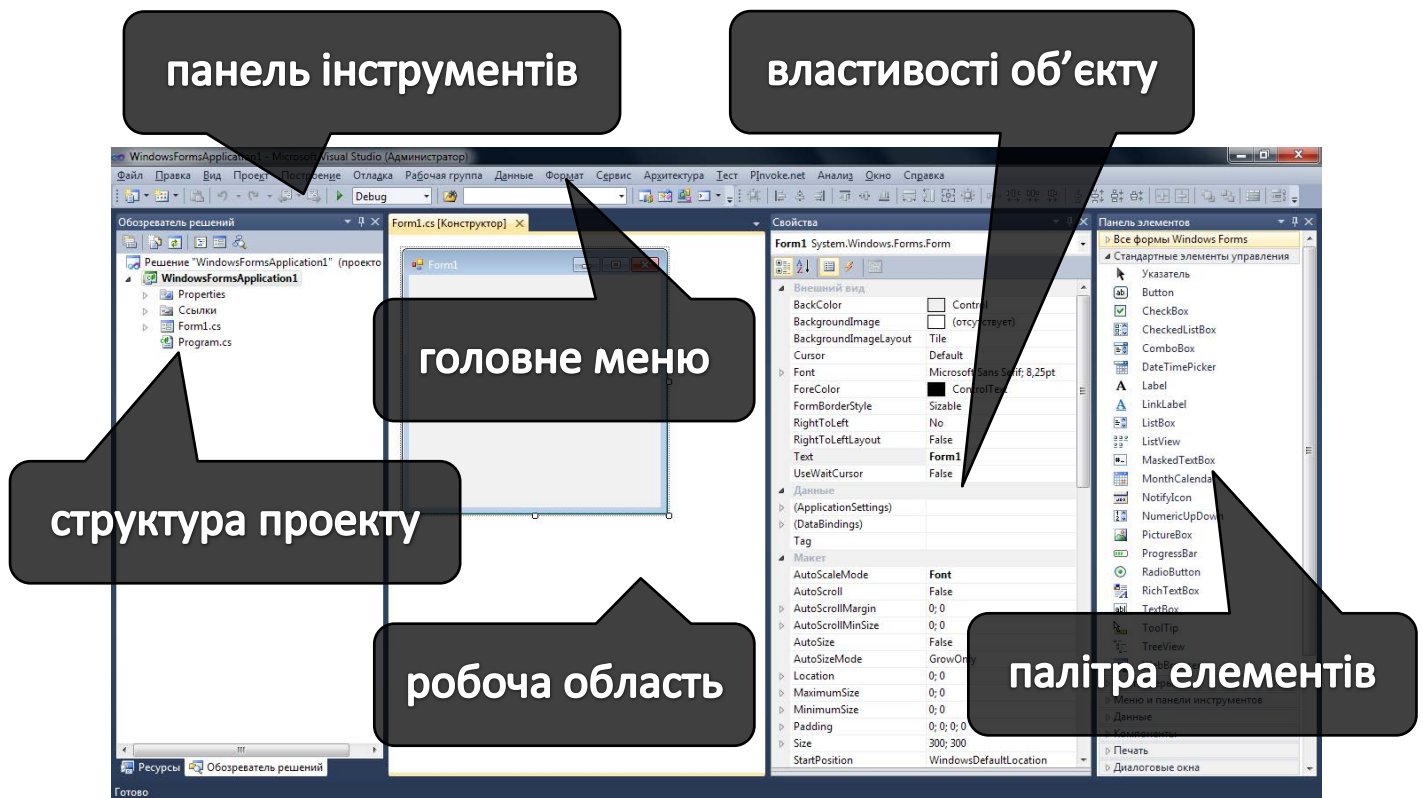


Рис. 5. Інтерфейс головного вікна

Панель інструментів дозволяє виконувати стандартні дії з проектом та його файлами, а також інші специфічні операції.

У **структурі проекту** відображаються всі файли проекту.

У **властивостях об'єкту** відображаються всі властивості об'єкту (це може бути форма, об'єкт на ній, файл у структурі проекту та інше).

За допомогою **палітри елементів** можна додати на форму новий об'єкт. Основна **робоча область** відображає форму, код програми, тощо.

4. Додавання компонентів на форму

Щоб додати компонент на форму потрібно перетягнути його з палітри елементів на форму. Також можна вибрати потрібний компонент на палітрі та натиснути на вільне місце на формі.

Додайте на форму компонент **Button** (Рис. 6).

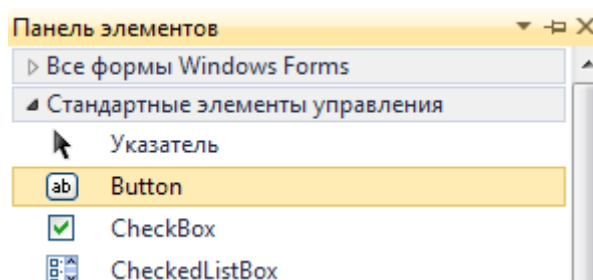


Рис. 6. Виділений компонент Button

5. Редагування властивостей об'єкта

Виберемо на формі кнопку button1. Тепер у панелі властивостей об'єкту відобразяться її властивості (Рис. 7).

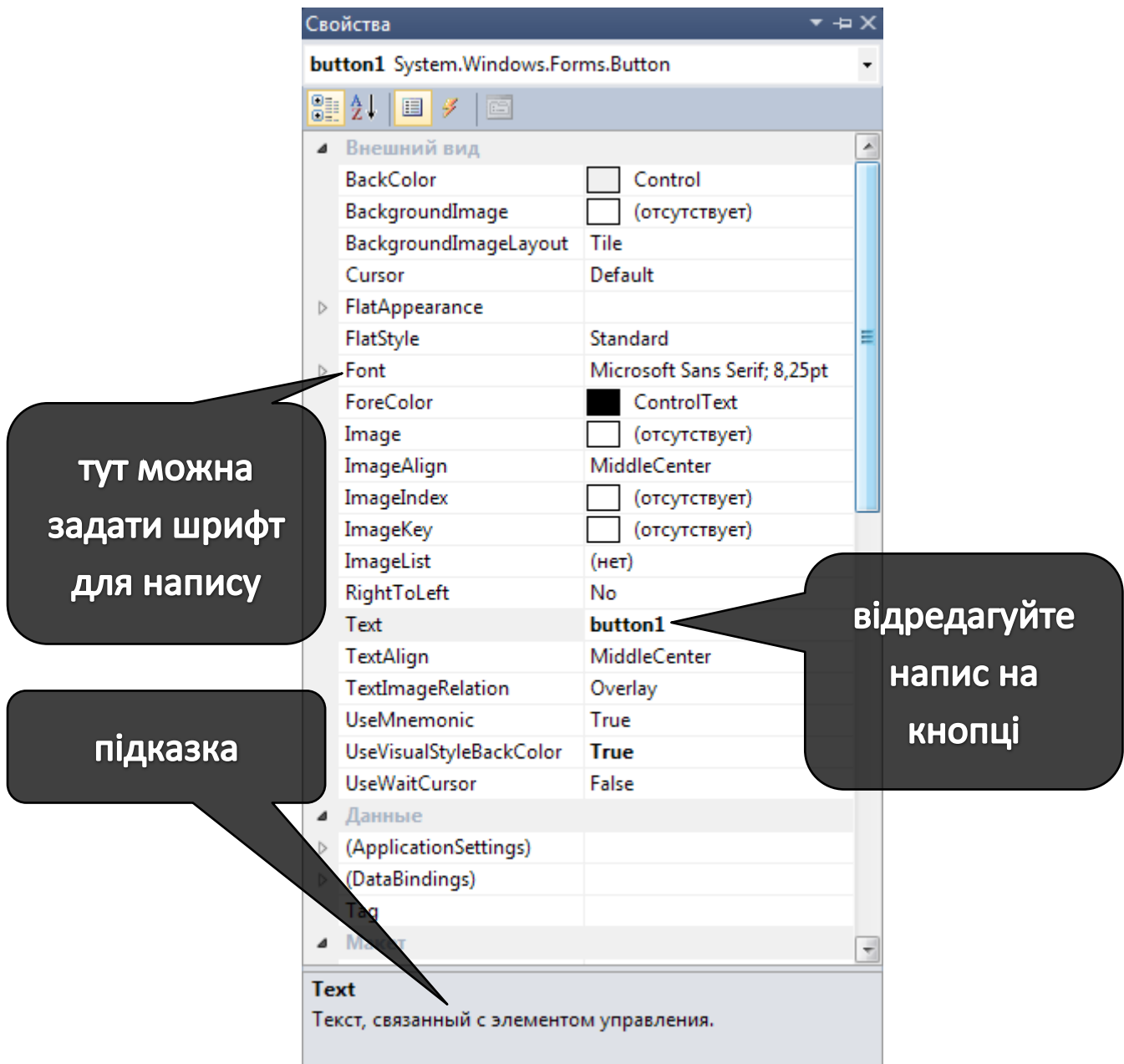


Рис. 7. Властивості кнопки button1.

Щоб змінити напис на кнопці потрібно відредагувати властивість **Text**. Змінимо напис з «button1» на «Кнопка».

Для того, щоб перемістити об'єкт на формі його потрібно захопити мишею та перетягнути на нове місце. Подібним чином можна змінити його розміри. Це схоже на те, як ми перетягуємо та змінюємо розмір вікна. У панелі властивостей є такі поля, які за це відповідають (Рис. 8):

- Location (X, Y) – координати розташування відносно форми;
- Size (Width, Height) – розміри об'єкта.

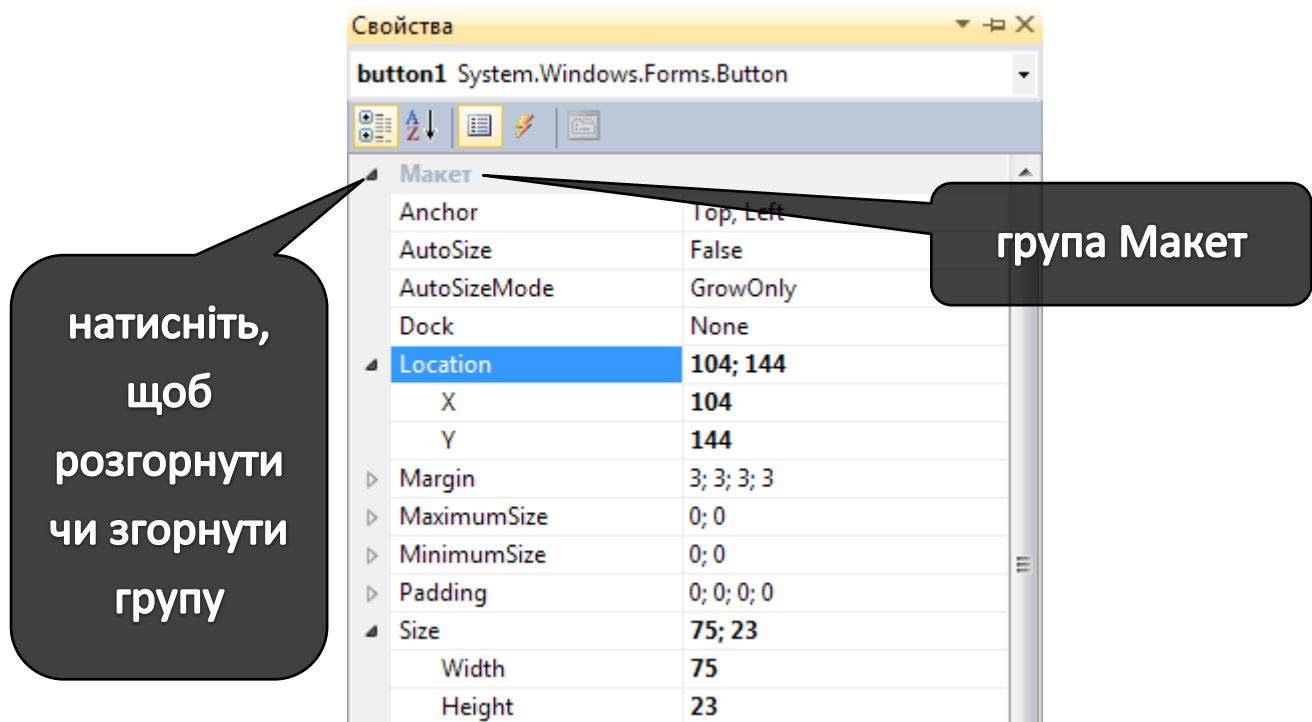



Рис. 8. Координати розташування та розмір button1.

6. Запуск програми

Для запуску програми у режимі налагодження (Debug) натисніть  на панелі інструментів або клавішу **F5**. Це діє тільки коли у проекті вибрана конфігурація налагодження (Рис. 9).

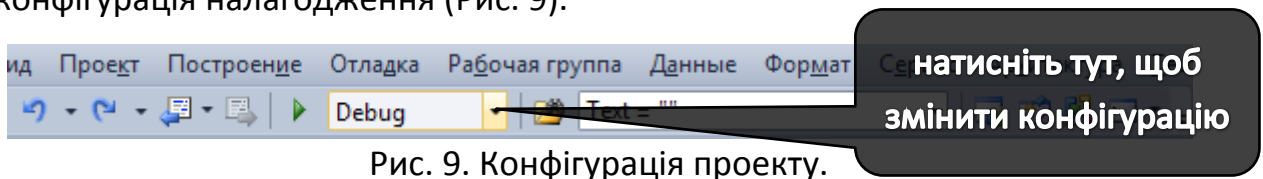
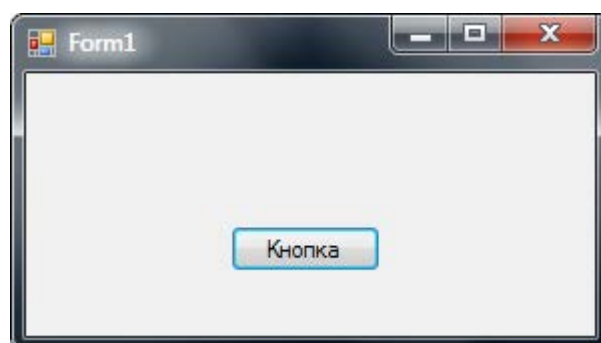


Рис. 9. Конфігурація проекту.

Конфігурація Release (випуск) дозволяє скомпілювати кінцеву версію програми. Вона на відміну від Debug не містить інформації для налагодження (ця інформація зовсім не потрібна кінцевому споживачу), і код програми (мається на увазі низькорівневий байт-код) оптимізований компілятором. Тому програма має менший розмір і швидше працює. У цій конфігурації для створення програми натисніть клавішу F6.

Всі лабораторні роботи залишайте у конфігурації Debug, це дозволить вам запустити програму прямо з середовища Visual Studio (клавіша F5) та полегшить знаходження помилок у коді.

Запустіть програму на виконання (F5) і ви побачите вікно з кнопкою:



7. Збереження та повторне відкриття проекту

Для збереження проекту натисніть комбінацію **Ctrl+Shift+S** або виберіть пункт меню «Файл» і в ньому команду «Сохранить все».

Для відкриття проекту натисніть комбінацію **Ctrl+Shift+O** або використайте меню «Файл» → «Открыть» → «Решение или проект...».

8. Редагування коду

Код програми на C# може міститися у декількох файлах з розширенням **cs**. При створенні віконного проекту (Windows Forms) середовище автоматично створює два файли з кодом, які можна редагувати (є і третій файл, але його може змінювати лише середовище).

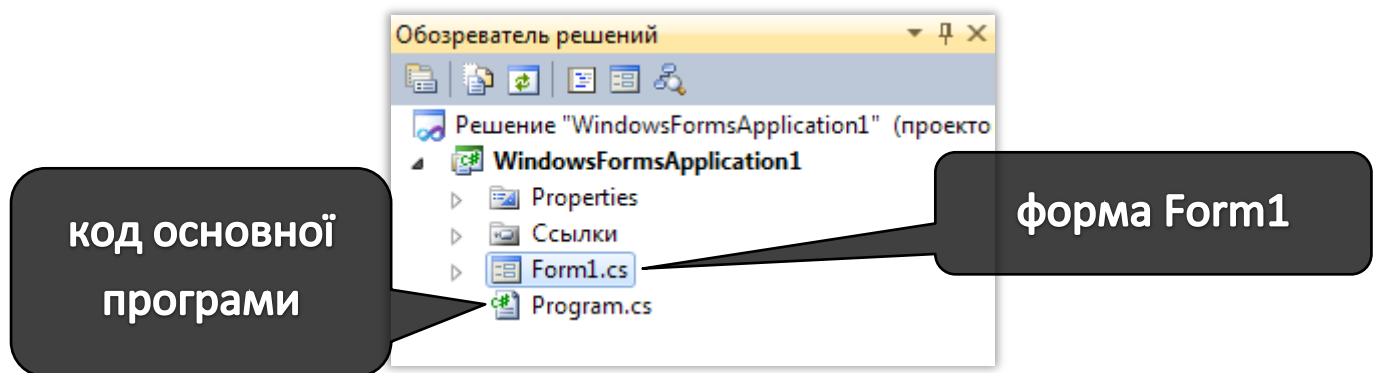



Рис. 11. Файли з кодом.

На рис. 11 показано які файл з кодом створюються у віконному проекті.

Значок  показує, що код у файлі відноситься до форми.

Значок  означає, що перед нами просто якийсь код.

Зазвичай у одному файлі описується один клас і ім'я файлу збігається з назвою класу, однак це не обов'язково. Наприклад у файлі **Form1.cs** описується клас **Form1**, який походить від класу **Form**.

Щоб відкрити який код, потрібно двічі клацнути по ньому. Якщо ви так зробите з кодом форми (піктограма ) , то замість коду відкриється конструктор форм. Щоб відкрити код, потрібно виділити форму та натиснути **F7** або викликати команду «Перейти к коду» з контекстного меню.

Відкриємо код **Program.cs** та докладно розглянемо його.

Перші чотири строки містять:

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

На початку кожної строки стоїть **директива using**. Вона дозволяє використовувати типи з простору імен, які описані після неї.

Далі стоїть строка:

```
namespace WindowsFormsApplication1
```

Вона означає початок опису простору імені **WindowsFormsApplication1**. Це головний простір для нашої програми.

Після відкритої операторної дужки { бачимо:

```
static class Program
```

Цією стрічкою ми оголошуємо статичний клас програми **Program**.

І нарешті, головна статична функція **Main**, яка виконується найпершою у програмі:

```
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new Form1());
}
```

З класу **Application** визиваються такі функції:

EnableVisualStyles() – вмикає візуальні стилі оформлення для форми;

SetCompatibleTextRenderingDefault(false) – вимикає сумісність між різними способами виводу тексту;

Run(new Form1()) – створює новий екземпляр Form1 та запускає цикл обробки повідомлень від користувача.

Код форми **Form1.cs** містить подібні стрічки:

```
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 WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

Тут ми описуємо клас Form1 і через двокрапку вказуємо, що він походить від базового класу **Form**. Ключове слово **partial** вказує, що не весь клас Form1 об'явлений у даному файлі (Form1.cs). Тобто у інших файлах проекту є опис інших полів та методів цього класу (таким додатковим файлом для Form1 є Form1.Designer.cs).

Метод **Form1()** представляє собою конструктор класу Form1, у якому викликається функція **InitializeComponent()** класу Form1; ця функція описана у файлі Form1.Designer.cs.

9. Склад файлів проекту

Збережіть проект та закрийте Visual Studio.

Відкрийте теку в яку ви зберегли проект. Вона матиме такий вигляд:

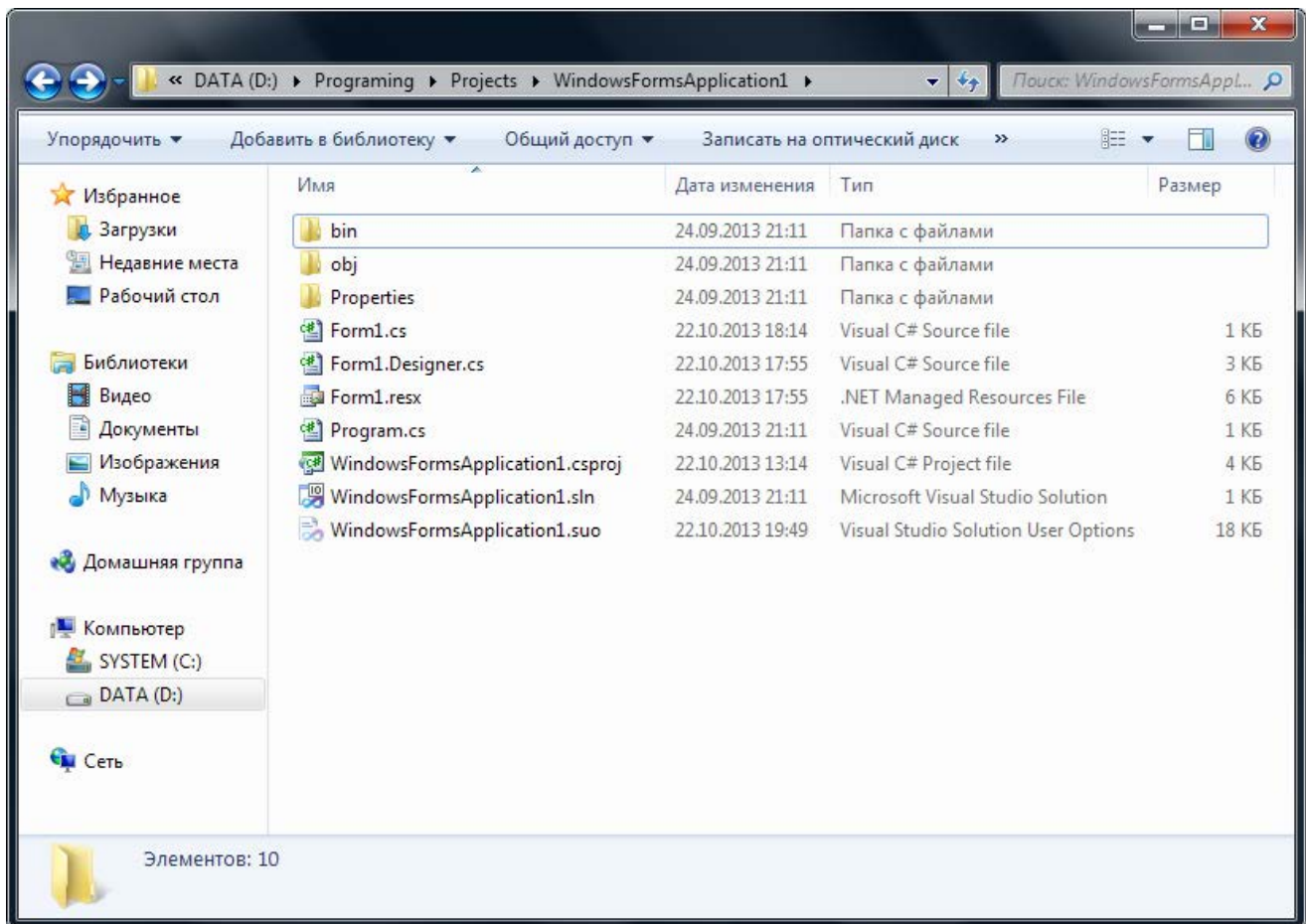


Рис. 12. Тека проекту.