

# Лабораторна робота №1

## Створення програм мовою C# у Microsoft Visual Studio

### План

1. Створення консольного проекту на C# у Microsoft Visual Studio
2. Компіляція та запуск програми
3. Використання налагоджувача
4. Завдання

1. IDE (Integrated Development Environment – інтегроване середовище розробки) Microsoft Visual Studio (далі VS) є нативним середовищем програмування для платформи .NET Framework, що підтримує можливість швидкої розробки як консольних програм, так і програм з графічним інтерфейсом різними мовами програмування. У VS вбудовані редактор коду, компілятори, налагоджувач та багато інших інструментів, що дозволяють ефективно створювати програмні застосунки різних типів для платформи .NET.

Останню версію VS можна завантажити з офіційного сайту <https://visualstudio.microsoft.com/>. Слід зазначити, що VS є комерційним програмним продуктом, проте є можливість завантаження безкоштовної версії Community. Після завантаження та інсталяції запуск програми здійснюється або за допомогою головного меню ОС Windows, або за допомогою відповідної піктограми на робочому столі. Після запуску VS на екрані з'явиться вікно приблизно (залежно від версії) такого типу (рис. 1).

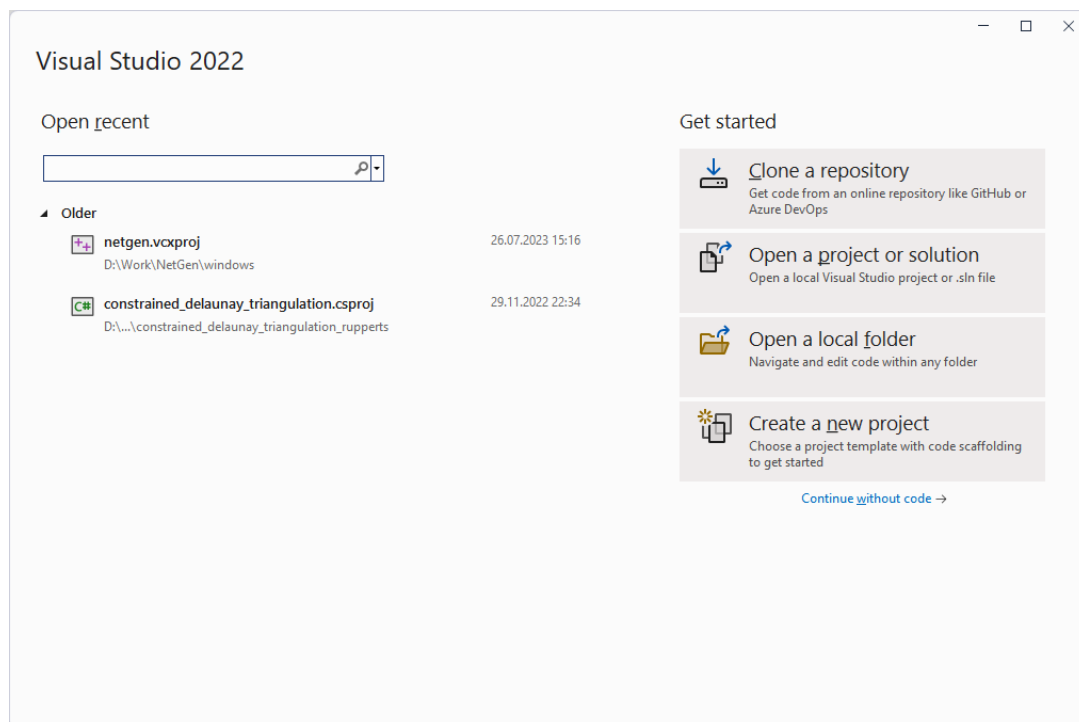


Рис. 1 – Загальний вигляд початкового вікна Microsoft Visual Studio

За його допомогою користувач може відкрити або один із раніше створених проектів (їх список знаходиться в лівій частині вікна), або ж у правій частині вибрати потрібну йому дію (наприклад, «Open a project or solution» (відкрити проект або рішення) для відкриття наявного проекту).

Для створення нового проекту користувач повинен вибрати пункт «Create a new project» (створити новий проект), після чого з'явиться нове вікно (рис. 2)

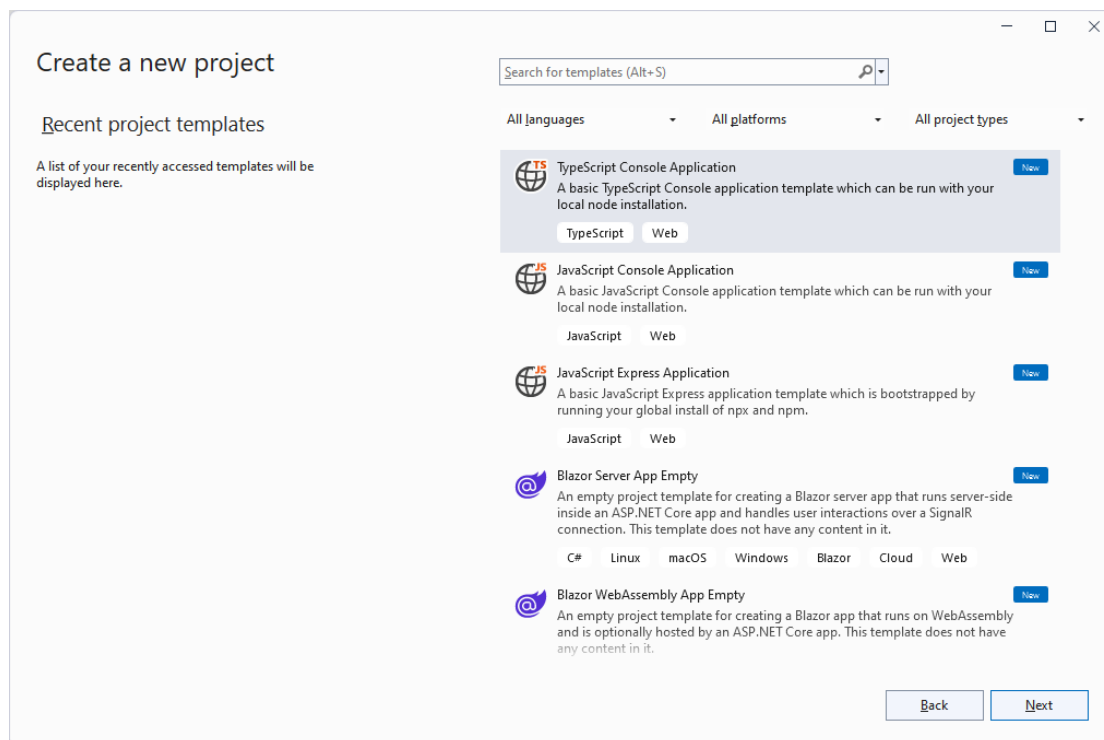


Рис. 2 – Вікно параметрів проекту

Залежно від кількості встановлених компонентів, пошук потрібного типу проекту може бути досить тривалим. Введемо у вікно «Search for templates» (пошук шаблонів) ключове слово «C#» (латинськими символами), після чого список доступних проектів буде зменшено згідно з вказаним ключем (рис. 3).

Виберемо пункт «Console App» (консольна програма) для C# і натиснемо кнопку «Next» (далі). З'явиться ще одне вікно налаштування проекту, в якому потрібно вказати такі параметри, як «Project Name» (назва проекту), «Location» (місце розташування) та «Solution Name» (назву рішення, де останнє – це контейнер для одного або кількох проектів) (рис. 4).

Після чого може з'явитися ще одне вікно (рис. 5), в якому деталізується бажана версія Framework.

Нарешті, після натискання кнопки «Create» (створити), якщо все було зроблено правильно, новий проект буде створено та відкрито в середовищі VS (рис. 6).

**2.** На цьому процедура створення консольного проекту для .NET на C# може вважатися закінченою. Спробуємо виконати компіляцію та запуск нашого проекту.

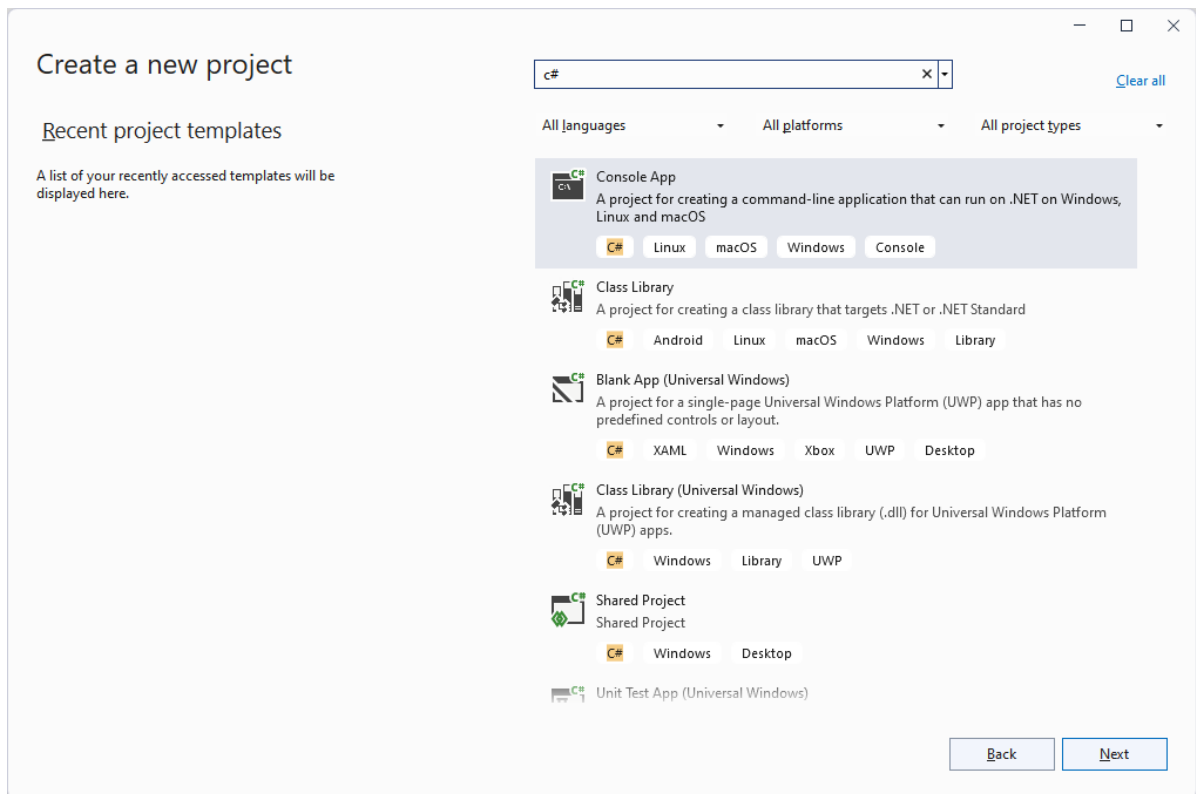


Рис. 3 – Пошук потрібного типу проекту

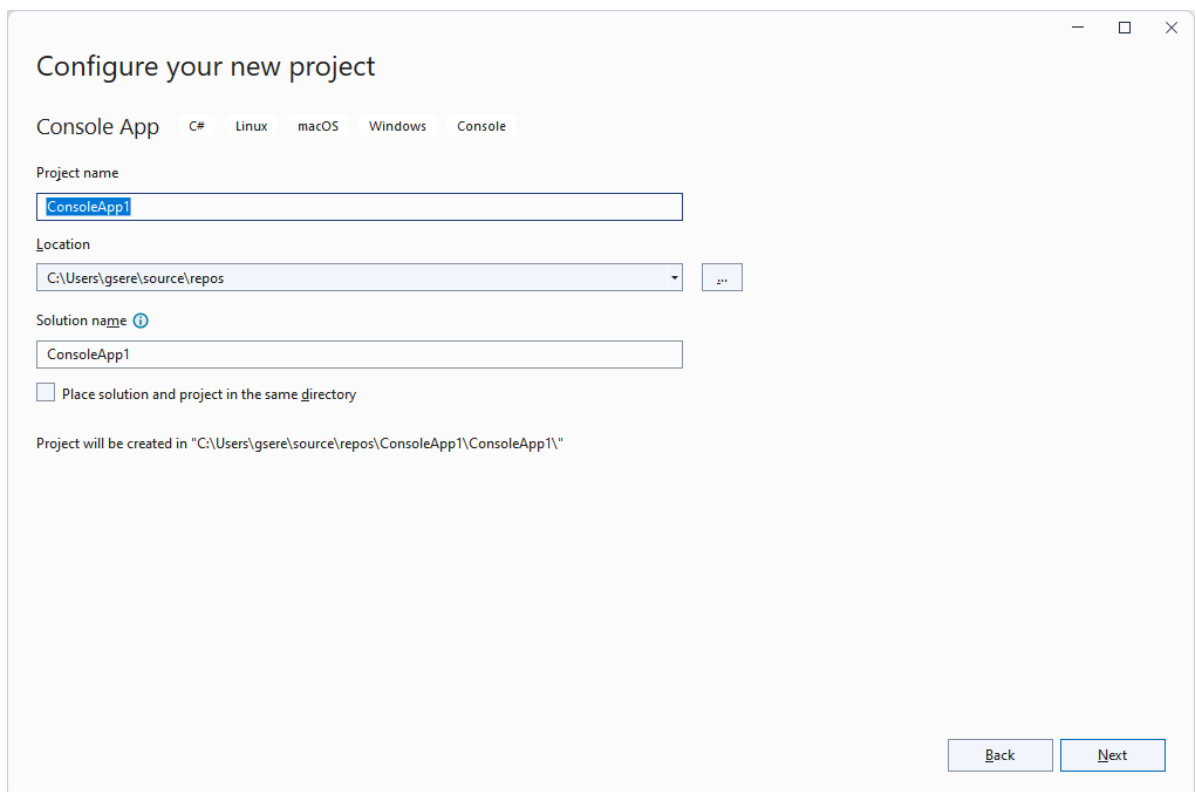


Рис. 4 – Визначення місця розташування та назви проекту

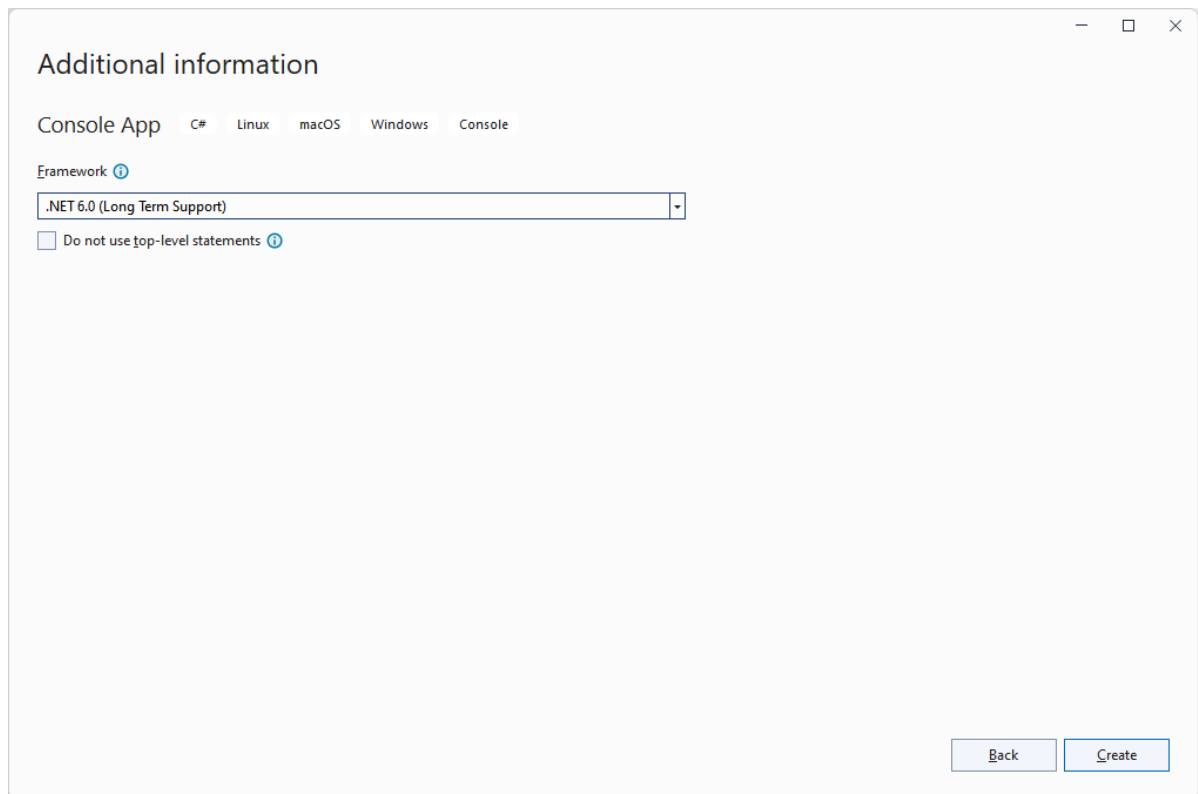


Рис. 5 – Вибір бажаної версії Framework

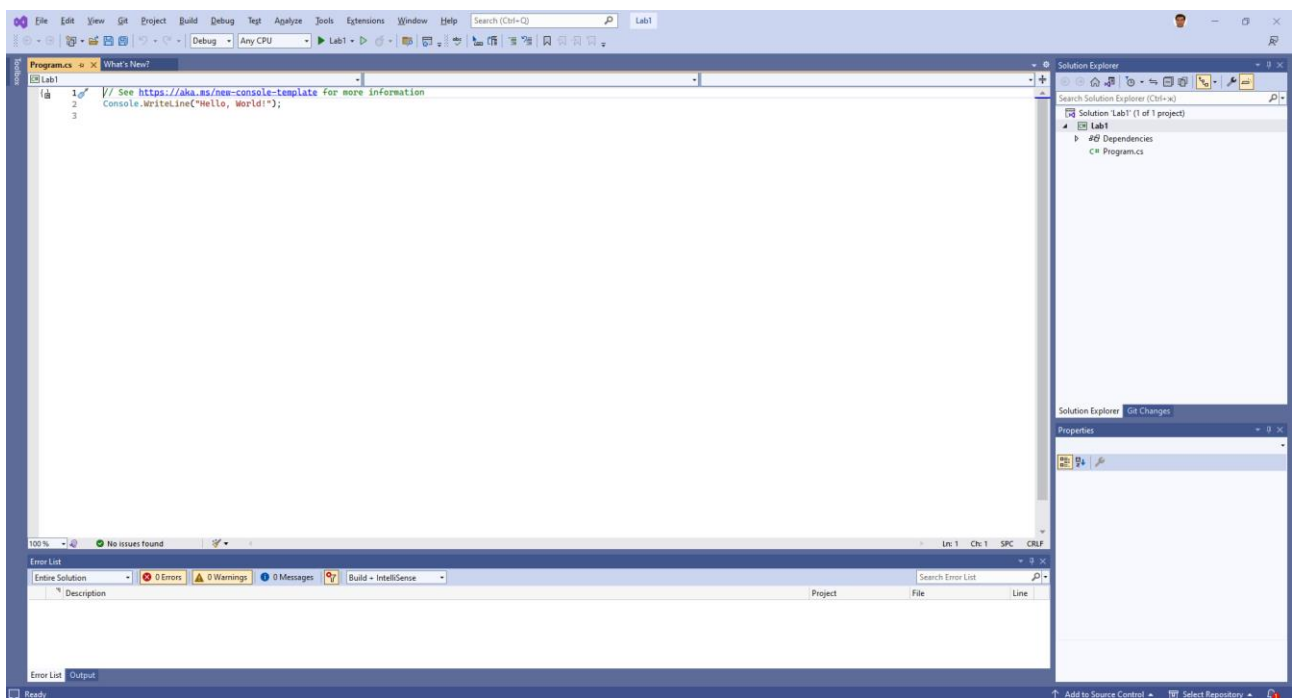


Рис. 6 – Новий консольний проект у середовищі Microsoft Visual Studio

Зробити це можна кількома способами. Наприклад:

1) у головному меню вибрати пункт «Build» (побудувати) | «Build Solution» (побудувати рішення);

2) натиснути на клавіатурі комбінацію клавіш <Ctrl>+<B> або <Ctrl>+<Shift>+<B>;

3) клацнути по кнопці ► на панелі інструментів у верхній частині головного вікна VS (запуск програми для налагодження, якщо проект не компілювався або змінювався, спочатку запусниться процес трансляції).

Якщо в програмі немає помилок, то компіляція повинна пройти успішно, після чого можна запустити програму також кількома способами, наприклад, вибравши команду головного меню «Debug» (налагодження) | «Start Debugging» або натиснувши <Ctrl>+<F5>. Приклад роботи запущеної програми наведено на рис. 7.

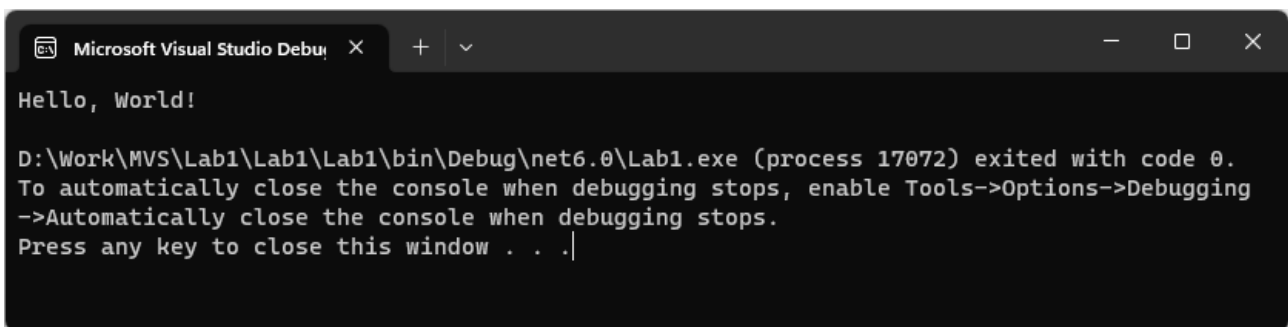


Рис. 7 – Результат виконання програми

Тут слід зробити наступне зауваження. Мова С# на відміну, наприклад, від С++ є повністю об'єктно-орієнтованою. Тобто в ній не існує «вільних» функцій (таких, що не є методами будь-яких класів). В С++ же для збереження зворотної сумісності є принаймні одна функція main(), яка не є членом будь-якого класу.

В С# мінімальна консольна програма має приблизно такий вигляд:

```
using System;

namespace MyApp // Note: actual namespace depends on the project name.
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

Тобто точкою входу в програму є статичний метод класу Program з назвою Main(). Проте, починаючи з версії .NET 6, шаблон консольного проекту в С# було істотно спрощено:

```
// See https://aka.ms/new-console-template for more information
```

```
Console.WriteLine("Hello, World!");
```

Ці дві форми представляють одну й ту саму програму. У разі використання новішої версії програмісту потрібно написати тільки тіло методу Main(). Компілятор автоматично згенерує клас Program з методом Main() і добавить всі необхідні оператори та елементи програми.

3. Для налагодження програм VS використовується вбудований дебагер. Запустити процедуру налагодження програми можна, наприклад, з використанням пункту головного меню «Debug», вибираючи у меню, що відкриється, потрібний режим.

Зручним способом налагодження програм є завдання у вихідному тексті точок зупинки (клацнувши мишкою у відповідну позицію зліва від номера потрібного рядка початкового коду або скориставшись відповідною командою меню). Після чого здійснюється запуск програми в режимі налагодження, наприклад, за допомогою клавіші <F5>. VS перевірить, чи є зібрана програма актуальною, і перезбере її за необхідності. За допомогою налагоджувача можна відстежити логіку виконання програми, а також значення змінних на кожному етапі її роботи (рис. 8).

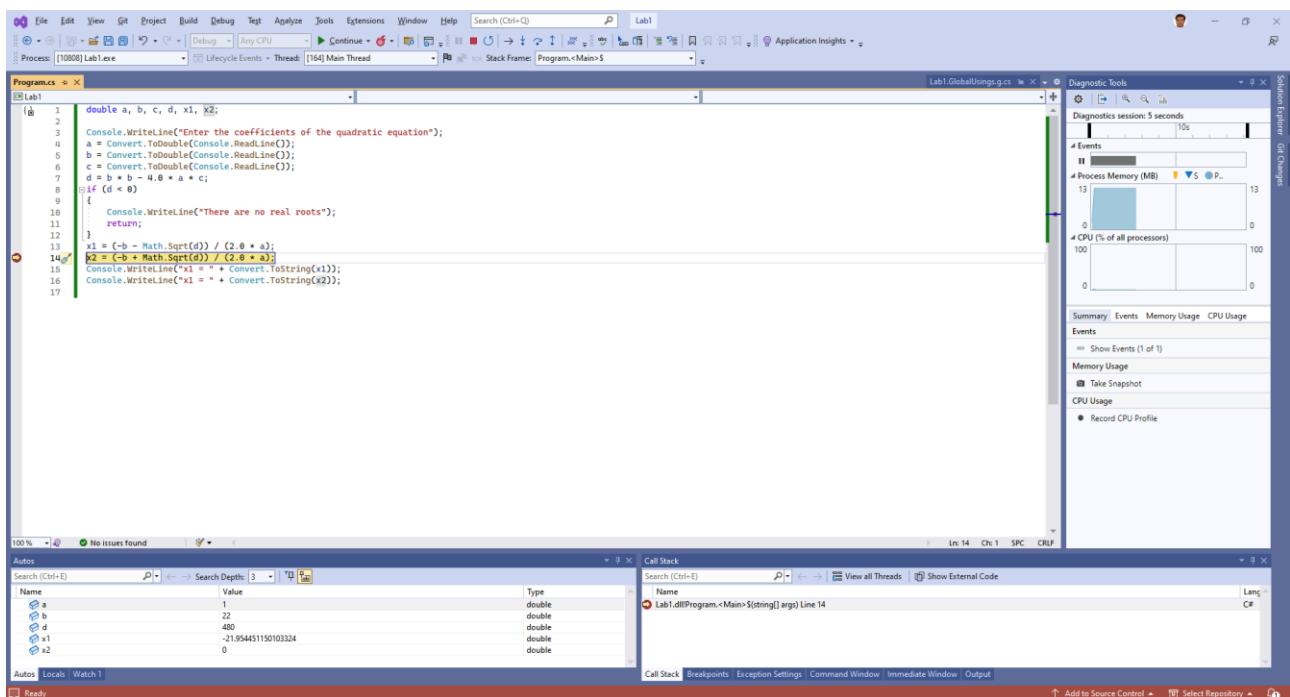


Рис. 8 – Налагодження програм (точка зупинки)

Натиснувши клавішу <F10> у режимі налагодження можна перейти до наступного рядка програми без заходу всередину функції або <F11> – із заходом. Іншими словами, якщо в наступному рядку програми при налагодженні є виклик функції, то при натисканні <F11> зневаджувач перейде до її початкового тексту.

Вміння ефективно використовувати налагоджувач є суттєвою перевагою програміста, оскільки дозволяє істотно підвищити швидкість розробки програм та пошук помилок у них і, як наслідок, веде до зростання продуктивності праці.

Приклад простої програми на C#, яка розв'язує квадратне рівняння, може виглядати так:

```
double a, b, c, d, x1, x2;

Console.WriteLine("Enter the coefficients of the quadratic equation");
a = Convert.ToDouble(Console.ReadLine());
b = Convert.ToDouble(Console.ReadLine());
c = Convert.ToDouble(Console.ReadLine());
d = b * b - 4.0 * a * c;
if (d < 0)
{
    Console.WriteLine("There are no real roots");
    return;
}
x1 = (-b - Math.Sqrt(d)) / (2.0 * a);
x2 = (-b + Math.Sqrt(d)) / (2.0 * a);
Console.WriteLine("x1 = " + Convert.ToString(x1));
Console.WriteLine("x2 = " + Convert.ToString(x2));
```

**4. Створити новий консольний проект C# VS. Написати програму мовою C#, що реалізує залежно від варіанта обчислення:**

- 1) площі паралелограма;
- 2) площі трикутника за формулою Герона;
- 3) площі бічної поверхні піраміди;
- 4) об'єм піраміди;
- 5) об'єм сфери.

Усі необхідні параметри повинні вводиться користувачем з консолі на запит програми. Передбачити виведення на екран ПІБ та номери групи студента, який здає роботу. Виконати покрокове виконання програми у налагоджувачі. Написати підсумковий звіт про виконання лабораторної роботи.