

Практична робота 1

Робота з проектами C# в Visual Studio. Консольне застосування

Для виконання роботи потрібна версія Visual Studio 2017 версії 15.3 або більш пізня з встановленою компонентою **Console App (. Net Framework)**.

Мета роботи

Вивчити основні компоненти і набір інструментів інтегрованого середовища розробки Microsoft Visual Studio, які використовуються для розробки консольних застосунків мовою C#.

Вивчити структуру проектів програм, створюваних автоматично на мові C# з використанням бібліотеки базових класів платформи .NET.

Навчитись створювати власні проекти консольних застосунків.

Зміст

1. Швидкий початок

1.1. Проекти і рішення

1.2. Створення консольного застосування.

- Додавання коду в застосування.
- Збірка застосування.
- Відлагодження і тестування програми.
- Збірка остаточної версії застосування.

1. Швидкий початок

1.1. Проекти і рішення

Середовище Visual Studio містить:

текстовий редактор з розвиненими засобами підказки (IntelliSense) і синтаксичного контролю тексту програм, що вводиться (Design-Time Debugging);

візуальний редактор конструктора форм, що забезпечує одночасно з візуальною побудовою графічного об'єкту його текстове представлення у вигляді операторів мови; *робочі вікна*;

елементи керування для автоматичного виклику компілятора, відлагоджувача і довідкової системи.

При першому запуску Visual Studio відображається стартова сторінка (Start Page), яка є HTML- сторінкою (рис. 2), що містить посилання на корисні Web-сайти, і дає можливість відкрити існуючий проект (Recent Project).

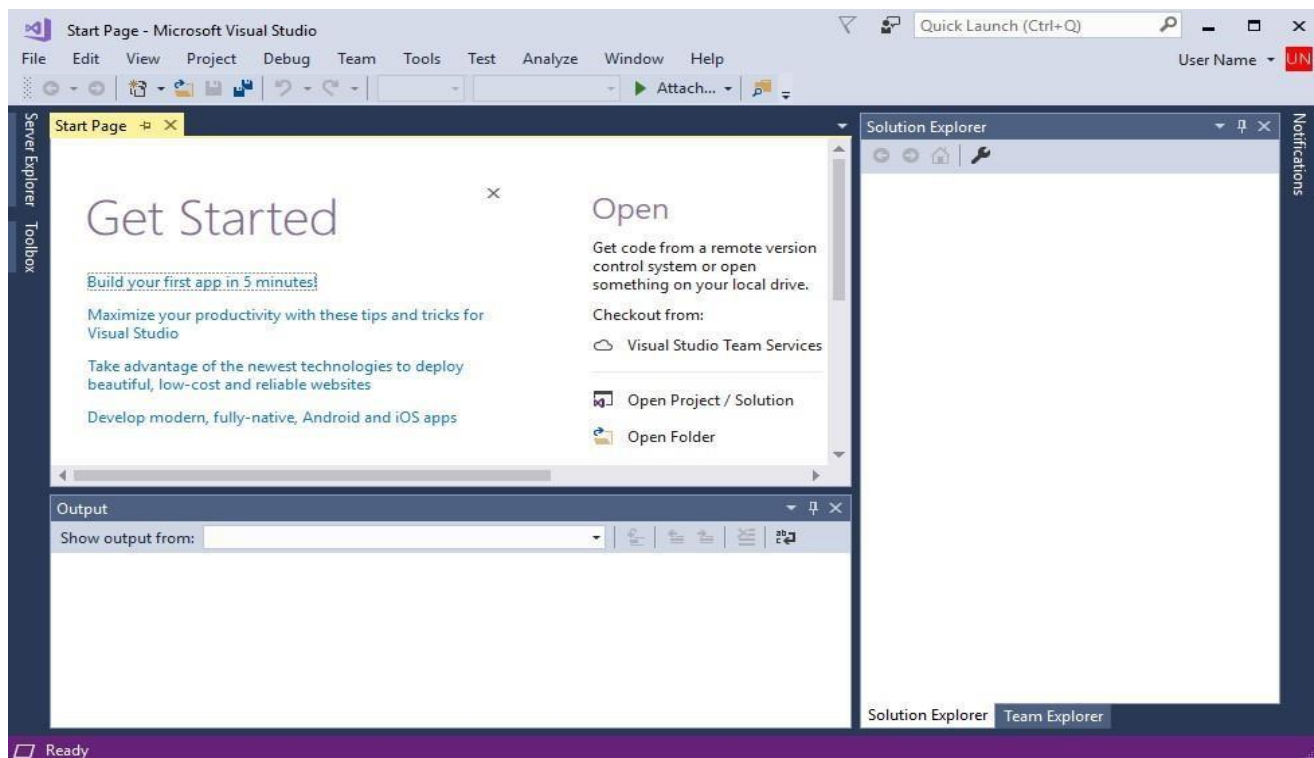
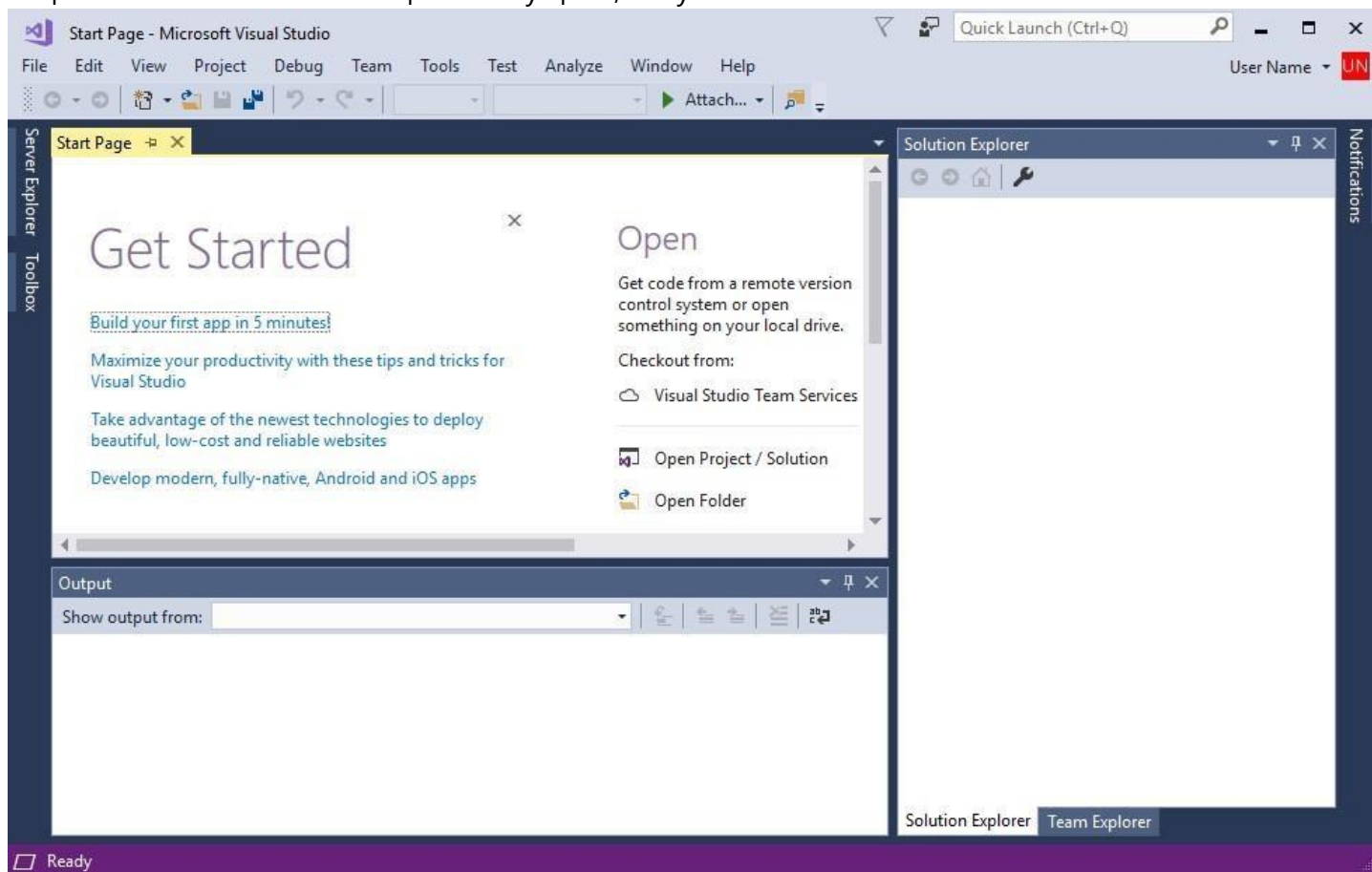


Рис. 2. Стартове вікно Visual Studio

1.2. Створення консольного застосування

Якщо система Visual Studio ще не запущена, запустіть її.



Робоча область інтегрованого середовища розробки Visual Studio містить три основні частини: *вікна інструментів*, які закріплені в лівій і правій частинах вікна програми; *меню* з

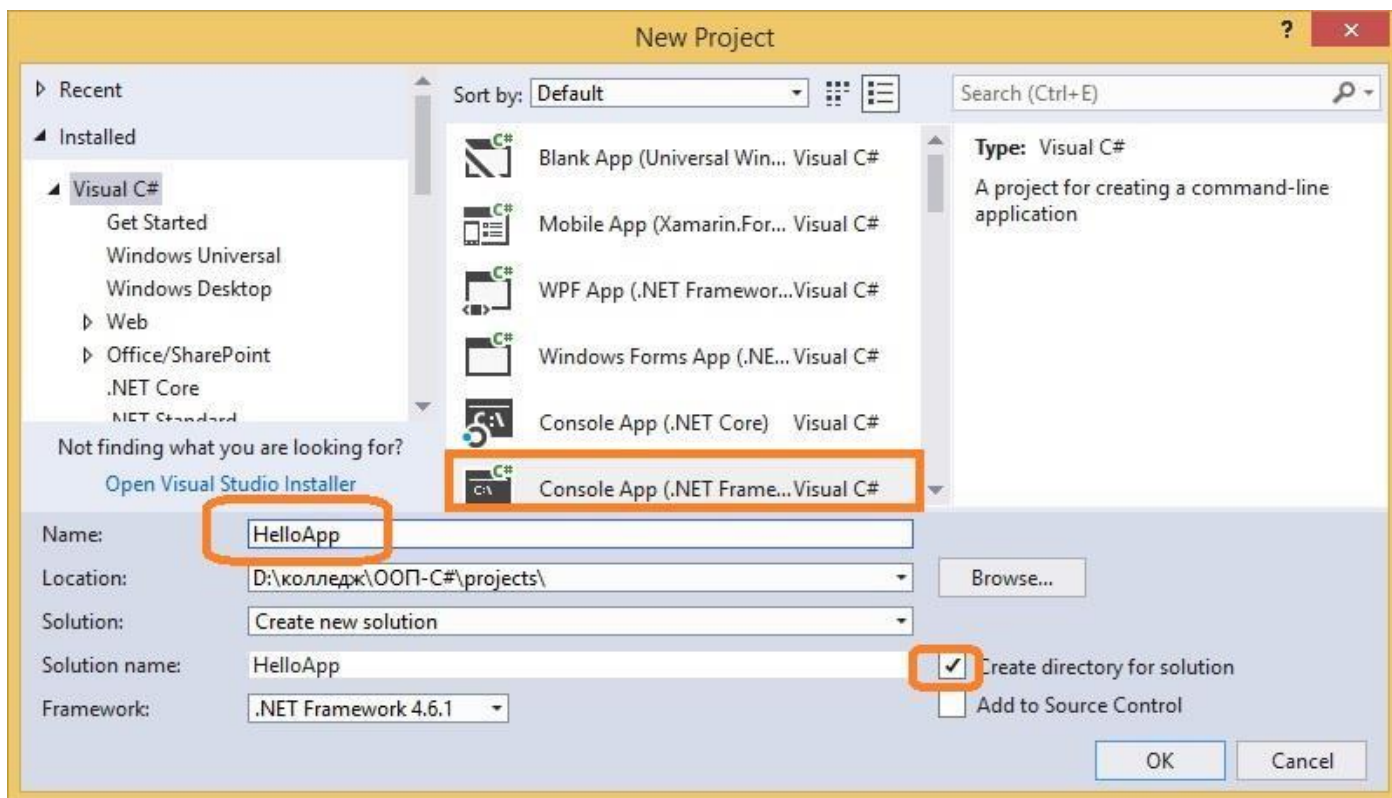
панелями інструментів - панель **Quick launch (Швидкий запуск)**, рядок меню і стандартна панель інструментів закріплені у верхній частині; *область головного вікна* розміщену в центрі області; саме тут знаходиться **Start Page (Початкова сторінка)**. Після відкриття рішення або проекту редактори і конструктори відображаються в цій області вміст файлів, код програми, тощо.

У Visual Studio група файлів застосування (код, ресурси, файли з параметрами налагоджень) називається **проектом (Project)**. Декілька проектів об'єднуються в рішення (**Solution**).

Проект містить всі параметри, опис конфігурації і правила, які використовуються для збірки коду, який буде виводитись. В файлі проекту вказані зв'язки між усіма файлами проекту і будь-якими зовнішніми файлами. Щоб створити застосування, спочатку створюється проект і рішення.

Створення проекту консольного застосування

1. В рядку меню послідовно виберіть пункти **File > New > Project (Файл > Створити > Проект)**, щоб відкрити діалогове вікно **New Project (Новий проект)**
2. В діалоговому вікні **New Project** виберіть **Installed > Visual C#**, якщо цей пункт ще не вибрано.
3. В центральній області виберіть шаблон **Console App (.NET Framework)**. В поле введення **Name** введіть *HelloApp*.
4. В діалоговому вікні **Location (Розташування)** вкажіть шлях до папки, де буде розміщено проект. Поле **Solution (Рішення)** та перемикач заповніть так, як на малюнку.



5. Щоб створити проект і рішення застосування, натисніть кнопку **ОК**.

Створюються проект HelloApp і рішення з основними файлами для консольного застосування Windows, після чого вони автоматично завантажуються в **Solution Explorer: (Браузері рішень)**. Файл *Program.cs* відкривається редакторі коду.

Для консольного додатку з ім'ям рішення HelloApp створюються наступні папки і файли:

Папка **HelloApp** – для рішення проекту застосування; В папці проекту **HelloApp**:

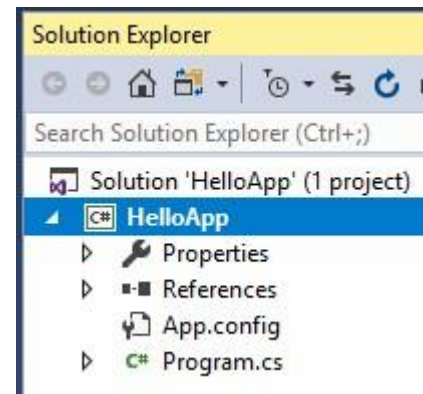
- а) XML-файл **HelloApp.csproj** – для опису параметрів проекту (файл проекту);
- б) папка **Properties** – для зберігання властивостей проекту з файлом AssemblyInfo.cs, де міститься додаткова інформація про проект;
- в) тека **References** – для посилань на системні бібліотеки .NET;
- г) файл **Program.cs** – код головного модуля застосування; Всі файли коду C# мають розширення **.cs**.

Додавання коду в застосування

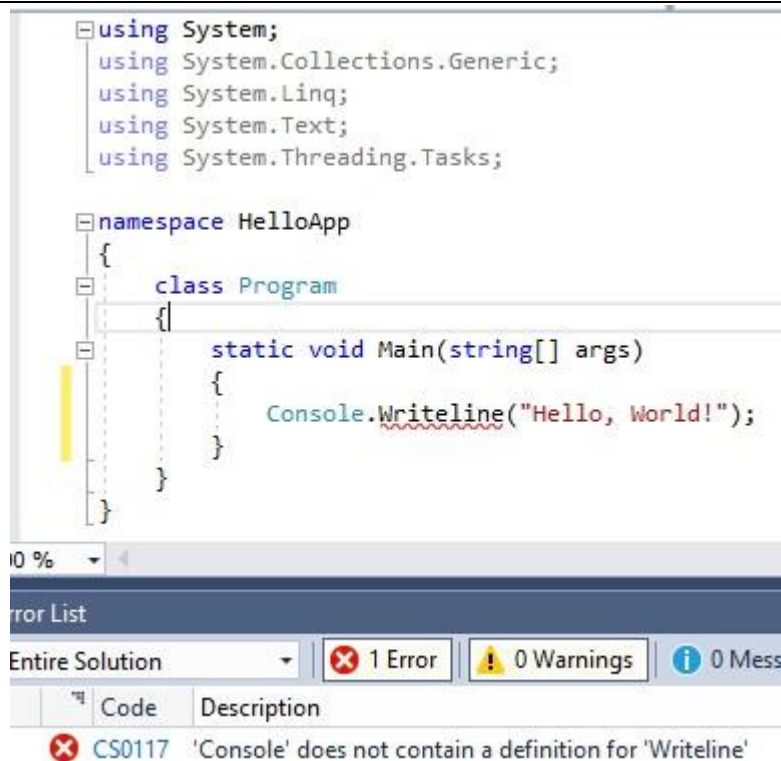
Visual Studio використовує шаблон для створення проекту консольного застосування. В шаблоні визначено клас *Program* з одним методом *Main*, який приймає в якості аргументу масив *String*. *Main* – це точка входу в додаток, метод, який автоматично викликається середовищем виконання під час запуску програми. Всі аргументи, які вказані в командному рядку при запуску програми, доступні через масив *args*. Слово *void* вказує на те, що дана програма не повертає результат обчислень у програму, яка її викликала.

В файлі *Program.cs* введіть наступний код: `Console.WriteLine("Hello, World!");`;

Червона хвиляста лінія з'явиться під `WriteLine`. При наведенні на неї вказівника миші з'являється повідомлення про помилку.



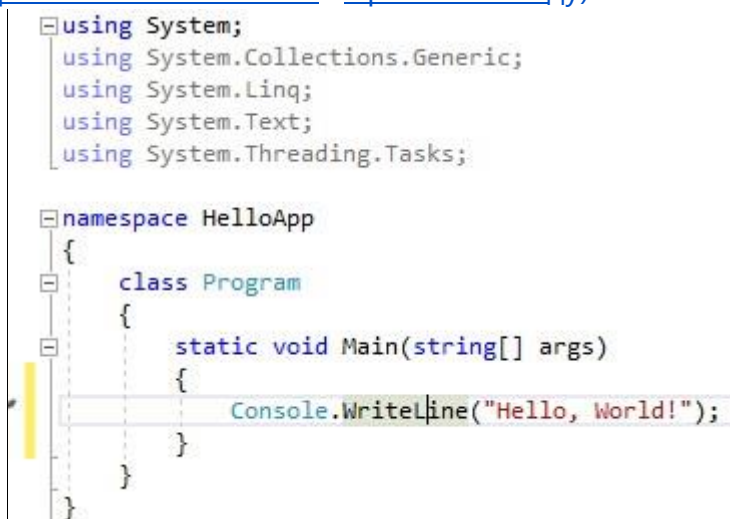
В



Повідомлення про помилку також відобразиться у вікні **Error List**. Ви можете відобразити це вікно, вибравши в рядку меню **View > Error List**.

В коді неправильно вказано назву метода. Замініть **Writeline** на **WriteLine**.

Під час виправлення з'являється вікно, яке пропонує різні варіанти автозаповнення символів, які мають вводитись. Це вікно є частиною технології IntelliSense, що забезпечує для коду, який вводиться, підказки, відображає члени класів і інтерфейсів та іншу інформацію. Ця технологія значно пришвидшує роботу завдяки можливості використання готових фрагментів коду (див. статті [Використання IntelliSense](#) і [Фрагменти коду](#)).



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HelloApp
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Writeline("Hello, World!");
        }
    }
}
```

Червона хвиляста лінія під **Console.Writeline** зникне після виправлення помилки.

Щоб зберегти зміни у файлі, натисніть клавіші **CTRL+S**.

Структура програми

Перші рядки програми підключають простори імен:

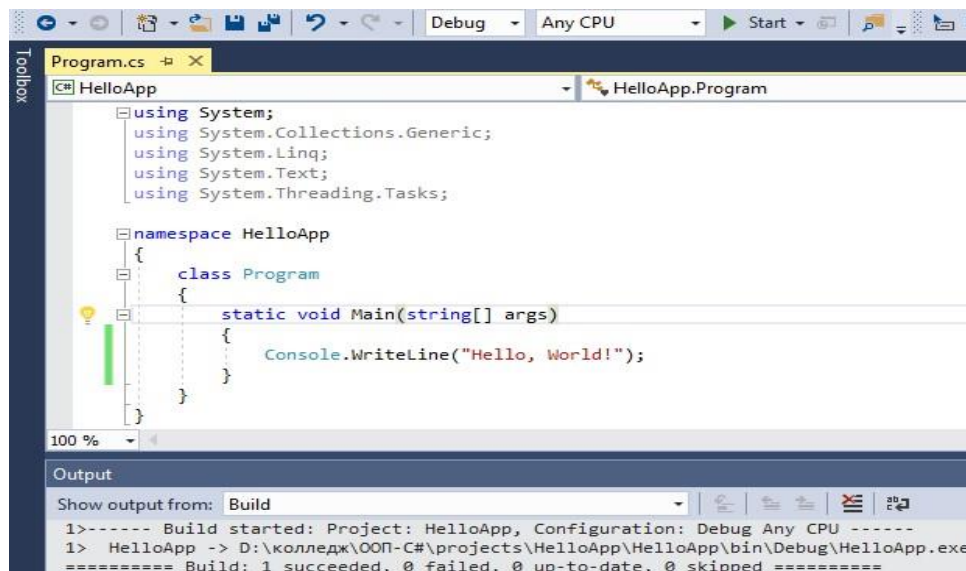
`System, System.Collections.Generic, System.Linq, System.Text, System.Threading.Text.`

Простір імен містить інформацією про об'єкти, які використовуватиме наша програма. У просторах імен згруповані пов'язані категорії коду, вони утворюють бібліотеку класів .NET. В просторі імен є батьківський клас, якому належать інші класи чи об'єкти.

Простір імен головного модуля застосування оголошується як namespace **HelloApp**. Він містить лише один клас **Program**. Метод `Main` також належить простору імен **HelloApp**.

Збірка застосування

У рядку меню послідовно виберіть **Build > Build Solution**. Visual Studio створює збірку для рішення *HelloApp* та інформує про виконання цього процесу у вікні **Output**.



Відлагодження і тестування програми

За допомогою відлагоджувача можна подивитися, чи відображається слово "Hello" у вікні консолі після запуску *HelloApp*.

Відлагодження програми

Щоб запустити відлагоджувач, виберіть **Debug > Start Debugging (Відлагодження> Почати відлагодження)** в рядку меню.

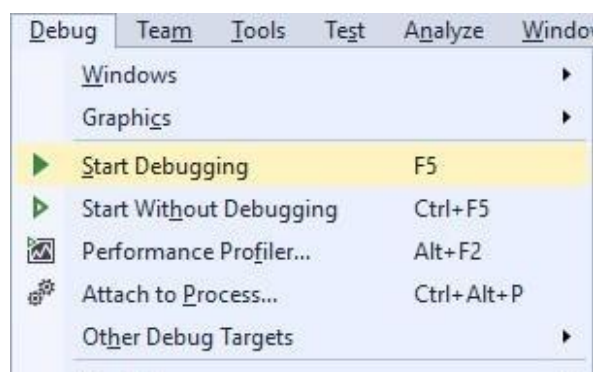
Відлагоджувач запуститься і виконає код. Вікно консолі (окреме вікно, подібне командному рядку) відображається протягом декількох секунд і швидко закривається після закінчення виконання програми. Щоб переглянути текст, необхідно встановити точку зупинки (**Breakpoint**) виконання програми.

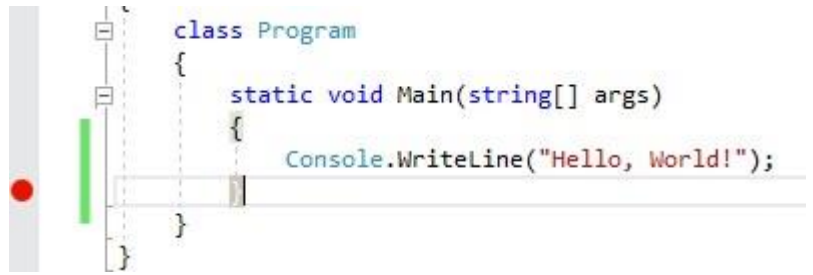
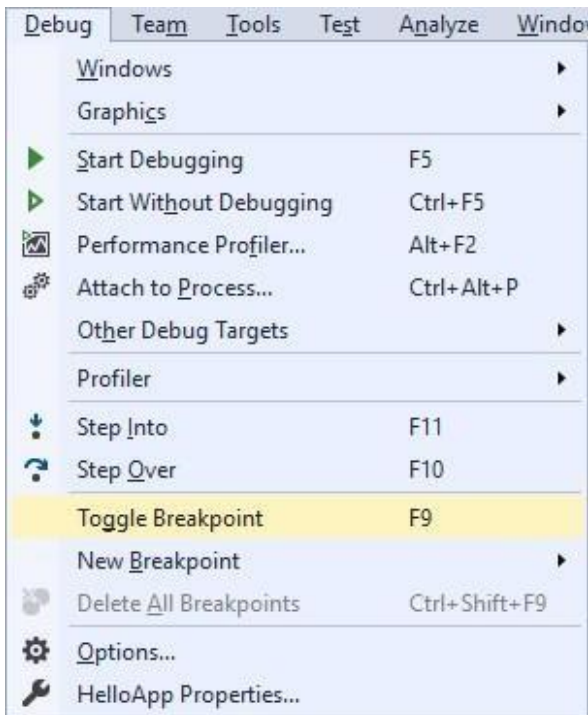
Додавання точки зупинки

1. У редакторі встановіть курсор в рядок наступний за оператором виведення. У рядку меню виберіть

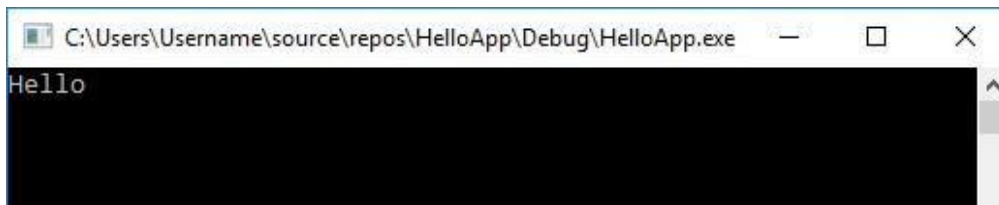
Debug > Toggle Breakpoint (Відлагодження> Переключити точку зупинки). Для того, щоб встановити точку зупинки можна також клацнути на лівій межі області коду.

Поруч з рядком коду в крайньому лівому полі вікна редактора з'явиться червоний кружок.





2. Щоб почати відлагодження, натисніть клавішу F5. Запускається відлагоджувач і з'являється вікно консолі, в якому виводиться слово **Hello**.



Зауваження. Вікно може бути прихованим під Visual Studio. Щоб його побачити – клікніть на позначці Visual Studio на панелі задач або згорніть Visual Studio.

3. Щоб зупинити відлагодження, натисніть клавіші **SHIFT+F5**.

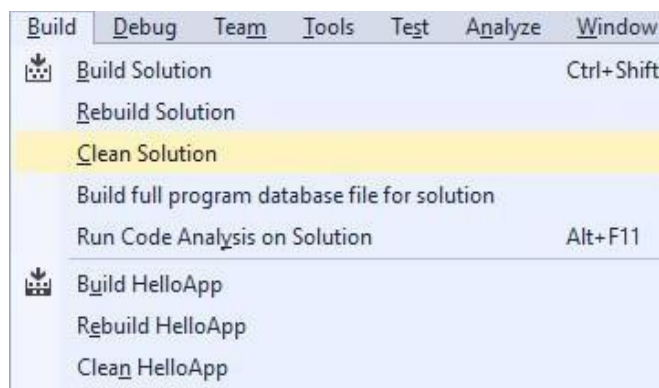
Додаткові відомості про відлагодження консольного проекту див. в статті [Консольні проекти](#).

Збірка (Building) кінцевої версії (Release)

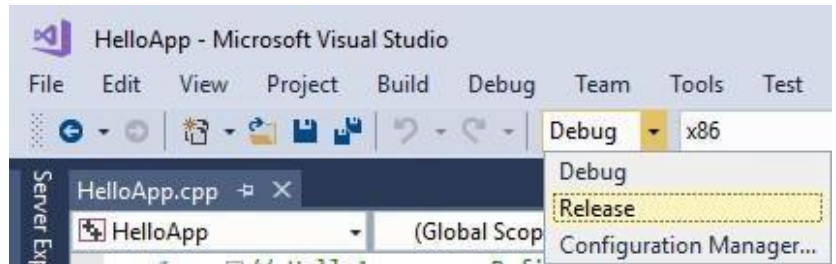
Якщо програма працює коректно, то можна підготувати кінцеву збірку (Release builds). Збірка Release не містить інформацію для відлагоджувача, а під час її компіляції використовуються створюється швидший код меншого розміру.

Очищення файлів рішення і збір остаточної версії

1. Виберіть в рядку меню **Build > Clean Solution (Збірка> Очистити рішення)**, щоб видалити проміжні і вихідні файли, створені в ході попередніх збірок.



2. Щоб змінити конфігурацію рішення для HelloApp з **Debug** на **Release (Відлагодження на Випуск)**, на панелі інструментів відкрийте список поруч з елементом управління "Конфігурація рішення" і виберіть **Release (Випуск)**.



3. Виконайте збірку застосування для рішення. В рядку меню послідовно виберіть **Build > Build Solution (Збірка > Зібрати рішення)**.

Після закінчення збірки отримуємо застосування, яке можна скопіювати і запустити в будь-якому вікні командного рядка.