

Bài 4.6: Phương thức Main()

- ✓ Mục đích sử dụng
- ✓ Các biến thể của Main
- ✓ Chạy chương trình trong CMD
- ✓ Ví dụ minh họa

Mục đích sử dụng

- ✓ Phương thức Main là phương thức bắt buộc phải có để có thể vận hành ứng dụng .exe build từ C#.
- ✓ Phương thức Main là nơi kích hoạt chương trình khi ứng dụng khởi chạy.
- ✓ Lúc này chưa có bất kì đối tượng nào, do đó phương thức Main phải đi kèm keyword static để đảm bảo chương trình có thể được kích hoạt mà không cần đối tượng.
- ✓ Một lớp có thể có hoặc không có phương thức Main.
- ✓ Nếu một project có nhiều phương thức Main, ta phải chỉ rõ sẽ chạy Main của class nào.
- ✓ Ta có thể nạp chồng phương thức Main nhưng mỗi class chỉ 1 phương thức Main hợp lệ được sử dụng để kích hoạt ứng dụng.

Các biến thể của Main

- ✓ Phương thức Main có nhiều biến thể khác nhau tùy vào mục đích sử dụng:
 - ✓ Có cần trả về hay không? Nếu có thì trả về int, không thì trả về void.
 - ✓ Có cần chạy bất đồng bộ không? Nếu có thì dùng async Task.
 - ✓ Có cần cung cấp tham số cho chương trình khởi chạy không? Có thì dùng tham số string[] args.
- ✓ Sau đây là các biến thể của Main được sử dụng để kích hoạt ứng dụng:

```
// phương thức Main không tham số
static void Main() { }
static int Main() { }
static async Task Main() { }
static async Task<int> Main() { }

// phương thức Main có tham số string[]
static void Main(string[] args) { }
static int Main(string[] args) { }
static async Task Main(string[] args) { }
static async Task<int> Main(string[] args) { }
```

Kiểu trả về của Main

- ✓ Kiểu trả về của phương thức Main có thể là void, int, Task hoặc Task<int>.
- ✓ Nếu kiểu trả về của phương thức Main là void, mặc định nó sẽ return 0 cho hệ thống khi chương trình thực hiện hoàn chỉnh không có vấn đề gì.
- ✓ Nếu kiểu trả về của Main là int, khi chương trình kết thúc không có lỗi ta return 0.
- ✓ Khi chương trình có vấn đề không kết thúc bình thường, ta return giá trị khác 0.
- ✓ Mục đích là để báo trạng thái chạy của ứng dụng cho nơi thực hiện lời gọi đến phương thức Main vừa rồi. Từ đó đưa ra hướng xử lý phù hợp.

```
static int Main()
{
    // nếu chương trình kết thúc thành công
    return 0;
}

static int Main()
{
    // nếu chương trình kết thúc do có lỗi
    return 1;
}
```

Tham số của Main

- ✓ Phương thức Main có thể không có, có 1 hoặc nhiều tham số.
- ✓ Nhưng trong số này chỉ có phương thức nhận tham số `string[] args` và phương thức không tham số là được sử dụng để kích hoạt chương trình.
- ✓ Nếu các tham số khác được bổ sung, phương thức Main không được coi là entry point của class chứa nó nữa.
- ✓ Ví dụ:

```
static int Main() { } // entry point
```

```
static void Main() { } // entry point
```

```
static int Main(string[] args) { } // entry point
```

```
static void Main(string[] args) { } // entry point
```

```
static void Main(int number) { } // not an entry point
```

Chạy chương trình với lệnh CMD

- ✓ Tham số `string[] args` được sử dụng để nhận các đối số cung cấp khi khởi chạy chương trình.
- ✓ Đoạn chương trình sau mô tả số lượng và nội dung cụ thể của các tham số đã cung cấp cho chương trình trong lệnh kích hoạt.

```
0 references
class Lesson46
{
    0 references
    static void Main(string[] args)
    {
        Console.WriteLine("So luong cac doi so: " + args.Length);
        Console.WriteLine("Danh sach cac doi so: ");
        for (int i = 0; i < args.Length; i++)
        {
            Console.WriteLine($"args[{i}]: " + args[i]);
        }
    }
}
```

- ✓ Để biên dịch chương trình ta mở Window PowerShell hoặc Developer PowerShell lên gõ lệnh: `csc tên_file.cs` sau đó nhấn enter.

Chạy chương trình với lệnh CMD

✓ Ví dụ file Lesson46.cs ta gõ:

```
Developer PowerShell
+ Developer PowerShell
*****
** Visual Studio 2022 Developer PowerShell v17.1.0
** Copyright (c) 2021 Microsoft Corporation
*****
PS C:\Users\trieu\source\repos\CSharpCourse\CSharpCourse> csc Lesson46.cs
Microsoft (R) Visual C# Compiler version 4.1.0-5.22109.6 (0c82c411)
Copyright (C) Microsoft Corporation. All rights reserved.
```

✓ Sau đó ta kiểm tra thư mục chứa file .cs vừa biên dịch ta sẽ thấy file Lesson46.exe.

✓ Để chạy chương trình ta gõ: `./Lesson46.exe danh_sách_đối_số` và nhấn enter.

✓ Ví dụ:

```
PS C:\Users\trieu\source\repos\CSharpCourse\CSharpCourse> ./Lesson46.exe "One" "Two" "Three" 1 2 3
Số lượng các đối số: 6
Danh sách các đối số:
args[0]: One
args[1]: Two
args[2]: Three
args[3]: 1
args[4]: 2
args[5]: 3
PS C:\Users\trieu\source\repos\CSharpCourse\CSharpCourse>
```

Chạy chương trình với lệnh CMD

✓ Ví dụ thực hiện phép toán đơn giản:

```
int a = int.Parse(args[0]);
int b = int.Parse(args[1]);
string op = args[2];
switch (op)
{
    case "add":
        Console.WriteLine($"{a} + {b} = {a + b}");
        break;
    case "sub":
        Console.WriteLine($"{a} - {b} = {a - b}");
        break;
    case "mul":
        Console.WriteLine($"{a} * {b} = {a * b}");
        break;
    case "div":
        Console.WriteLine($"{a} / {b} = {1.0f * a / b}");
        break;
    case "mod":
        Console.WriteLine($"{a} % {b} = {a % b}");
        break;
    default:
        Console.WriteLine("Wrong operation");
        break;
}
```

```
PS C:\Users\trieu\source\repos\CSharpCourse\CSharpCourse> ./Lesson46.exe 100 500 add
100 + 500 = 600
PS C:\Users\trieu\source\repos\CSharpCourse\CSharpCourse> ./Lesson46.exe 100 500 mul
100 * 500 = 50000
PS C:\Users\trieu\source\repos\CSharpCourse\CSharpCourse> ./Lesson46.exe 100 500 mod
100 % 500 = 100
PS C:\Users\trieu\source\repos\CSharpCourse\CSharpCourse> ./Lesson46.exe 100 500 div
100 / 500 = 0.2
PS C:\Users\trieu\source\repos\CSharpCourse\CSharpCourse> ./Lesson46.exe 100 500 pow
Wrong operation
PS C:\Users\trieu\source\repos\CSharpCourse\CSharpCourse>
```




Nội dung tiếp theo

Phương thức đệ quy