

Bài 4.4: Named and optional parameters

- ✓ Khái niệm và đặc điểm
- ✓ Đối số được đặt tên
- ✓ Đối số tùy chọn
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Khái niệm và đặc điểm

- ✓ C# 4 giới thiệu đối số được đặt tên và đối số tùy chọn.
- ✓ Đối số được đặt tên cho phép ta chỉ định một đối số của một tham số cụ thể qua tên của nó thay vì vị trí của tham số trong danh sách đối số.
- ✓ Sử dụng đối số được đặt tên cho phép ta cung cấp các đối số trong lời gọi phương thức không bắt buộc phải tuân theo thứ tự của tham số.
- ✓ Đối số tùy chọn cho phép ta bỏ qua các đối số của một số các tham số nào đó.
- ✓ Cả hai kỹ thuật trên đều có thể sử dụng với phương thức, indexer, constructor, delegate.
- ✓ Khi sử dụng hai kỹ thuật trên, các đối số được đánh giá theo thứ tự nó xuất hiện trong danh sách đối số, không phải theo thứ tự xuất hiện trong danh sách tham số.
- ✓ Đối số được đặt tên và đối số tùy chọn cho phép ta cung cấp các đối số cho các tham số được chọn.

Đối số được đặt tên

- ✓ Đối số được đặt tên cho phép bạn phá vỡ quy tắc thứ tự của danh sách tham số cần được cung cấp giá trị trong lời gọi.
- ✓ Mỗi tham số của từng đối số có thể được chỉ định cụ thể qua tên tham số muốn truyền vào.
- ✓ Cú pháp truyền đối số: `parameterName: argumentValue`
- ✓ Trong đó:
 - ✓ Phần `parameterName` là tên của tham số cụ thể trong phương thức.
 - ✓ Phần `argumentValue` là đối số cần truyền vào tương ứng. Có thể là biểu thức, hằng số, biến, giá trị cụ thể, lời gọi phương thức...
- ✓ Thứ tự các tham số được chỉ định không quan trọng, tùy ý bạn xác định.
- ✓ Nếu trong lời gọi có cả đối số được đặt tên và đối số ngầm định thì đối số được đặt tên phải đứng sau danh sách đối số ngầm định (theo thứ tự tham số).

Ví dụ

```
static void Main()
{
    var first = "Hoang";
    var last = "Ngo";
    var mid = "Minh";
    var age = 21;
    var gpa = 3.65f;
    // gọi thông thường
    ShowInfo(first, last, mid, age, gpa); // ok
    // gọi phương thức với đối số được chỉ rõ tên
    ShowInfo(firstName: first, age: age,
        gpa: gpa, lastName: last, midName: mid);
    ShowInfo(age: age, gpa: gpa,
        firstName: first, lastName: last, midName: mid);
    // kết hợp cả đối số theo thứ tự và đặt tên
    ShowInfo(first, last, mid, gpa: gpa, age: age);
    // theo đúng thứ tự tham số kết hợp ngầm định
    ShowInfo(first, last, midName: mid, age, gpa: gpa); // từ C# 7.2
}
```

5 references

```
static void ShowInfo(string firstName, string lastName,
    string midName, int age, float gpa)
{
    Console.WriteLine($"First Name: {firstName}");
    Console.WriteLine($"Last Name: {lastName}");
    Console.WriteLine($"Mid Name: {midName}");
    Console.WriteLine($"Age: {age}");
    Console.WriteLine($"GPA: {gpa}");
    Console.WriteLine("=====");
}
```

C:\WINDOWS\system32\cmd.exe

```
First Name: Hoang
Last Name: Ngo
Mid Name: Minh
Age: 21
GPA: 3.65
=====
```

```
First Name: Hoang
Last Name: Ngo
Mid Name: Minh
Age: 21
GPA: 3.65
=====
```

```
First Name: Hoang
Last Name: Ngo
Mid Name: Minh
Age: 21
GPA: 3.65
=====
```

```
First Name: Hoang
Last Name: Ngo
Mid Name: Minh
Age: 21
GPA: 3.65
=====
```

```
First Name: Hoang
Last Name: Ngo
Mid Name: Minh
Age: 21
GPA: 3.65
=====
```

Press any key to continue . . .

Đối số tùy chọn

- ✓ Trong định nghĩa của phương thức, constructor, indexer, delegate có thể chỉ định tham số của nó là bắt buộc hoặc tùy chọn.
- ✓ Trong lời gọi bắt buộc phải cung cấp đủ các tham số bắt buộc nhưng có thể bỏ qua tham số tùy chọn.
- ✓ Mỗi tham số tùy chọn được gán một giá trị mặc định khi khai báo tham số.
- ✓ Nếu không có đối số nào được truyền vào cho tham số tùy chọn, giá trị mặc định sẽ được sử dụng.
- ✓ Giá trị mặc định phải thuộc một trong các dạng biểu thức sau:
 - ✓ Một biểu thức hằng.
 - ✓ Một biểu thức dạng `new ValueType()` với `ValueType` là một kiểu giá trị như `enum` hay `struct`.
 - ✓ Một biểu thức dạng `default(Value)` với `Value` là một kiểu giá trị.
- ✓ Tham số tùy chọn luôn đặt ở cuối danh sách tham số, sau các tham số bắt buộc.
- ✓ Lời gọi phải cung cấp đủ đối số bắt buộc sau đó mới đến tham số tùy chọn nếu muốn.

Ví dụ

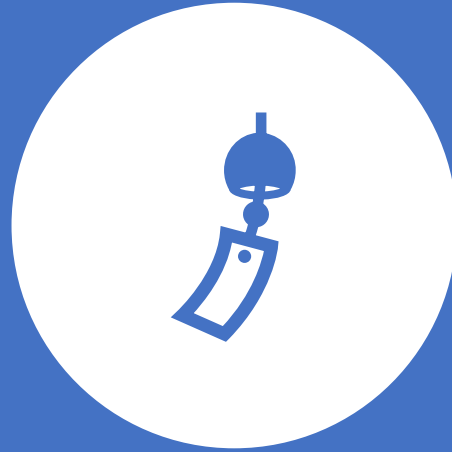
```
// gọi phương thức hiển thị thông tin sinh viên
ShowInfo("Nguyen Thi Loan", 20); // ok
ShowInfo("Nguyen Thi Loan", 20, "Da Nang"); // still ok
ShowInfo("Nguyen Thi Loan", 20, address: "Da Nang"); // also still ok
}

3 references
static void ShowInfo(string fullName, int age, string address = "Ha Noi")
{
    Console.WriteLine("Full name: " + fullName);
    Console.WriteLine("Age: " + age);
    Console.WriteLine("Address: " + address);
}
```

Ảnh hưởng nạp chồng

Sử dụng đối số được đặt tên và đối số tùy chọn sẽ ảnh hưởng tới lựa chọn phương thức nạp chồng:

- ✓ Một phương thức, indexer, constructor đều là ứng viên để gọi thực thi nếu mỗi bộ đối số truyền vào có thể chuyển sang kiểu tương ứng của tham số.
- ✓ Nếu có nhiều hơn 1 ứng viên được tìm thấy thì thành phần nạp chồng ưu tiên các đối số được chỉ định rõ ràng. Bỏ qua các đối số cho tham số tùy chọn.
- ✓ Nếu 2 ứng cử viên là tương đương, ưu tiên sẽ chuyển sang ứng viên không có tham số tùy chọn mà đối số đã bị bỏ qua trong lời gọi. Giải pháp nạp chồng sẽ thích các ứng viên ít tham số hơn.



Nội dung tiếp theo

Phương thức cục bộ