

Bài 6.2: Phương thức khởi tạo

- ✓ Khái niệm và đặc điểm
- ✓ Nạp chồng constructor
- ✓ Keyword this
- ✓ Ví dụ minh họa

Khái niệm và đặc điểm

- ✓ Phương thức khởi tạo là phương thức đặc biệt dùng để khởi tạo các thông tin ban đầu của đối tượng.
- ✓ Phương thức khởi tạo bao gồm access modifier tùy chọn, tên phương thức là tên của class; danh sách các tham số.
- ✓ Phương thức khởi tạo được gọi 1 lần duy nhất khi ta tạo mới một đối tượng của lớp.

```
class Person
{
    0 references
    public Person(string first, string last, int age)
    {
        FirstName = first;
        LastName = last;
        Age = age;
    }

    1 reference
    public string FirstName { get; set; }
    1 reference
    public string LastName { get; set; }
    1 reference
    public int Age { get; set; }
}
```

Nạp chồng constructor

- ✓ Theo mặc định, khi một lớp được tạo ra, nó đã có sẵn một constructor mặc định không tham số để khởi tạo các thành phần dữ liệu của lớp về giá trị mặc định của kiểu tương ứng.
- ✓ Constructor là phương thức, do đó ta có thể nạp chồng bằng cách thay đổi bộ tham số của nó.
- ✓ Để tạo nhanh một constructor trong Visual Studio bạn gõ **ctor** rồi **nhấn tab tab**.
- ✓ Khi ta định nghĩa một phương thức khởi tạo có tham số trong lớp, phương thức khởi tạo ngầm định sẽ bị ẩn đi. Nếu muốn sử dụng ta phải khai báo constructor này tường minh.
- ✓ Các constructor của một lớp được sắp xếp theo thứ tự tăng dần số lượng tham số.

Ví dụ

0 references

```
class Lesson62
{
    Person person = new Person(); // error!
}
```

3 references

```
class Person
{
    1 reference
    public Person(string first, string last, int age)
    {
        FirstName = first;
        LastName = last;
        Age = age;
    }

    1 reference
    public string FirstName { get; set; }
    1 reference
    public string LastName { get; set; }
    1 reference
    public int Age { get; set; }
}
```

Ta bổ sung constructor không tham số

```
class Lesson62
{
    Person person = new Person(); // error!
}

4 references
class Person
{
    1 reference
    public Person() // phương thức khởi tạo không tham số
    {
        FirstName = "";
        LastName = "";
        Age = 0;
    }

    // phương thức khởi tạo có tham số
    0 references
    public Person(string first, string last, int age)
    {
        FirstName = first;
        LastName = last;
        Age = age;
    }

    2 references
    public string FirstName { get; set; }
    2 references
    public string LastName { get; set; }
    2 references
    public int Age { get; set; }
}
```

Biểu thức bản thân

- ✓ Từ C# 6, nếu một phương thức có thể rút gọn trên 1 dòng, ta có thể sử dụng biểu thức bản thân để làm gọn code. Constructor có thể rút gọn bằng loại biểu thức này từ C# 7.
- ✓ Ví dụ:

```
1 reference
public Person()
{
    FirstName = "";
    LastName = "";
    Age = 0;
}
0 references
public Person(string first)
{
    FirstName = first;
}
0 references
public Person(string first, string last, int age)
{
    FirstName = first;
    LastName = last;
    Age = age;
}
```

```
1 reference
public Person()
{
    FirstName = "";
    LastName = "";
    Age = 0;
}
0 references
public Person(string first) => FirstName = first;
0 references
public Person(string first, string last, int age)
{
    FirstName = first;
    LastName = last;
    Age = age;
}
```

Keyword this

- ✓ Keyword this đại diện cho đối tượng hiện thời đang thực hiện các thao tác với phương thức.
- ✓ Keyword này được sử dụng trong các ngữ cảnh sau:
 - ✓ Trong phương thức tại đó tên tham số và thuộc tính/trường dữ liệu là trùng nhau. Lúc này bên nào có this thì bên đó là thuộc tính/trường của lớp.
 - ✓ Trong lời gọi phương thức tại đó cần truyền vào đối số là đối tượng hiện thời.
 - ✓ Trong khai báo indexer.

```
3 references
class Person
{
    1 reference
    public Person(string firstName, string lastName)
    {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public string firstName;
    public string lastName;
}
```

Fields

Parameters

Chuỗi lời gọi tới constructor

- ✓ Để tái sử dụng các phương thức khởi tạo, ta có thể gọi tới các constructor khác từ constructor hiện tại.
- ✓ Thứ tự lời gọi nên được thực hiện theo thứ tự: constructor ít tham số gọi constructor nhiều tham số hơn.
- ✓ Những giá trị bị khuyết ta truyền vào giá trị mặc định của kiểu hoặc theo quy ước của bạn.

Ví dụ



7 references

class Person

{

0 references

public Person() : this("", "")

{

// do other thing if needed

}

0 references

public Person(string first) : this(first, "")

{

// do other thing if needed

}

3 references

public Person(string firstName, string lastName)

{

this.firstName = firstName;

this.lastName = lastName;

}

public string firstName;

public string lastName;

}



Nội dung tiếp theo

Lớp static và các thành phần static