

Bài 6.14: Extension methods

- ✓ Khái niệm, đặc điểm
- ✓ Cú pháp triển khai
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Khái niệm và đặc điểm

- ✓ Các phương thức mở rộng cho phép ta thêm các phương thức mới vào các kiểu đã tồn tại trước đó mà không cần tạo mới một kiểu dẫn xuất hay sửa đổi kiểu gốc ban đầu.
- ✓ Các phương thức mở rộng là các phương thức static nhưng được gọi giống như là instance methods.
- ✓ Phổ biến nhất là các phương thức mở rộng viết cho LINQ, tập hợp.
- ✓ Thứ tự ưu tiên của phương thức mở rộng thấp hơn phương thức instance.
- ✓ Do đó nếu ta tạo một phương thức mở rộng có dấu hiệu nhận biết giống hệt một phương thức instance đã có trong kiểu nào đó thì phương thức mở rộng này sẽ không bao giờ được gọi.
- ✓ Phương thức mở rộng có thể được thêm vào một lớp hoặc một interface nhưng không override lớp hay interface.

Các đặc điểm

- ✓ Các phương thức mở rộng thường được ưu tiên sử dụng trong trường hợp mã nguồn không thuộc sở hữu hoặc không nằm trong tầm kiểm soát của bạn.
- ✓ Hoặc trường hợp tạo các kiểu dẫn xuất là không thể hoặc không phù hợp.
- ✓ Khi phương thức mở rộng được thêm vào các lớp ngoài tầm kiểm soát của bạn, nó có thể bị hỏng nếu chủ sở hữu thay đổi cách triển khai của các lớp liên quan tới extension method của bạn.
- ✓ Các phương thức mở rộng có phạm vi sử dụng ở cấp độ namespace. Khi cần sử dụng ta khai báo namespace chứa phương thức mở rộng vào để sử dụng.

Cú pháp triển khai

- ✓ Để định nghĩa và gọi phương thức mở rộng, thực hiện các bước sau:
- ✓ B1: tạo một lớp static để chứa extension methods. Access modifier của nó phải khả dụng với code muốn sử dụng ở bên ngoài.
- ✓ B2: triển khai extension methods như phương thức static với tối thiểu cùng cấp độ truy cập với access modifier của lớp chứa nó.
- ✓ B3: xác định kiểu cần mở rộng trong tham số đầu tiên. Thêm keyword **this** vào trước tên kiểu(class, interface). Sau đó bổ sung các tham số khác theo yêu cầu.
- ✓ B4: tại nơi gọi phương thức mở rộng, khai báo namespace chứa phương thức mở rộng bạn vừa định nghĩa ở trên qua cú pháp using NamespaceName;
- ✓ B5: gọi phương thức mở rộng giống gọi như phương thức thông thường.

Ví dụ

```
namespace CustomExtensions
{
    2 references
    public static class Extensions
    {
        1 reference
        // phương thức mở rộng để đếm từ có trong chuỗi đầu vào str
        public static int CountWord(this string str)
        {
            var spliter = new char[] { ' ', '.', '?', ',', ';', ':', '!', '\\t' };
            var data = str.Split(spliter, StringSplitOptions.RemoveEmptyEntries);
            return data.Length;
        }

        // phương thức mở rộng để tách các từ tại vị trí các dấu câu và trả về kết quả
        1 reference
        public static string[] SplitWord(this string str)
        {
            var spliter = new char[] { ' ', '.', '?', ',', ';', ':', '!', '\\t' };
            var data = str.Split(spliter, StringSplitOptions.RemoveEmptyEntries);
            return data;
        }
    }
}
```

Ví dụ

```
namespace CSharpCourse
{
    using CustomExtensions;
    0 references
    class Lesson614
    {
        0 references
        static void Main()
        {
            Console.WriteLine("Enter a string: ");
            var str = Console.ReadLine();
            Console.WriteLine("Word list: ");
            foreach (var item in Extensions.SplitWord(str))
            {
                Console.Write(item + " ");
            }
            Console.WriteLine("\nNumber of word in string: " +
                Extensions.CountWord(str));
        }
    }
}
```



Nội dung tiếp theo

Enum