Bài 6.10: Lớp trừu tượng

- ✓ Khái niệm và mục đích sử dụng
- ✓ Cú pháp và đặc điểm
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Khái niệm và mục đích sử dụng

- ✓Keyword abstract được sử dụng để chỉ ra rằng một thành phần nào đó chưa được triển khai hoàn chỉnh.
- ✓Keyword abstract có thể được sử dụng với class, methods, properties, indexers, events.
- ✓Keyword abstract được sử dụng với class để chỉ ra rằng class đó được tạo ra chỉ nhằm mục đích cho một nhóm lớp có liên quan khác kế thừa. Không phải để tạo đối tượng từ lớp đó.
- ✓Các thành phần chứa keyword abstract trong khai báo phải được triển khai bởi các lớp non-abstract kế thừa từ các lớp abstract.
- ✓Lớp abstract thể hiện tính trừu tượng: chỉ nêu ra vấn đề cần giải quyết nhưng không triển khai cụ thể. Tức là nói muốn làm gì nhưng không nói cụ thể cách làm.

Cú pháp và đặc điểm

- ✓Để tạo một thành phần abstract ta đặt keyword abstract ngay trước kiểu trả về hoặc trước class.
- ✓Trong lớp non-abstract bạn phải ghi đè các phương thức đó để triển khai cụ thể chức năng của nó. Nếu không, lớp đó cung phải trở thành lớp abstract.

```
1 reference
abstract class Shape
{
    0 references
    public int X { get; set; }
    0 references
    public int Y { get; set; }
    1 reference
    public abstract double Area(); // phương thức abstract
}

0 references
class Rectangle : Shape
{
    1 reference
    public double Width { get; set; }
    1 reference
    public double Height { get; set; }

    1 reference
    public override double Area() => Width * Height; // triển khai
}
```

Các đặc điểm của lớp, phương thức abstract

- ✓Không thể tạo đối tượng của lớp abstract.
- ✓Lớp abstract có thể chứa các phương thức abstract và các accessor abstract.
- ✓Không thể sử dụng keyword sealed với lớp abstract.
- ✓Lớp non-abstract kế thừa từ lớp abstract phải triển khai tất cả các phương thức abstract và accessor abstract có trong lớp cha.
- ✓Phương thức abstract ngầm định là virtual
- ✓Khai báo phương thức abstract chỉ được phép xuất hiện trong abstract class.
- ✓Phương thức abstract không có phần thân, kết thúc bằng dấu ; sau cụm ().
- ✓Không được phép sử dụng keyword static, virtual trong abstract methods.

Các thuộc tính abstract

- ✓Không sử dụng abstract với các thuộc tính static.
- ✓Thuộc tính abstract được kế thừa có thể được override trong lớp con bằng cách sử dụng override.

```
    static void Main()
    {
        Rectangle rectangle = new(); // tạo mới đối tượng Rectangle
        rectangle.Width = 20;
        rectangle.Height = 30;
        Console.WriteLine(rectangle.Area());

        // ta không thể tạo đối tượng của lớp abstract
        Shape shape = new Shape(); // error
    }
}

3 references
abstract class Shape
{
    0 references
    public int X { get; set; }
    0 references
    public int Y { get; set; }
    2 references
    public abstract double Area(); // phương thức abstract
}
```

Ví dụ

```
public abstract class Point
{
    protected int _x;
    protected int _y;
    // các accessor abstract
    1 reference
    public abstract int X { get; }
    1 reference
    public abstract int Y { get; }
}

0 references
public class DerivedPoint : Point
{
    // ghi đè các accessor abstract
    1 reference
    public override int X { get { return _x + 10; } }
    1 reference
    public override int Y { get { return _y + 10; } }
}
```

Nội dung tiếp theo

Tìm hiểu về interface