

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (Object-Oriented Programming)

Nguyễn Thị Hồng Anh

Chương 5. Interface

Nội dung



- Khái niệm
- Cách khai báo
- Thực thi Interface
- Abstract và Interface

Chương 5. Interface

Khái niệm



- Một **Interface trong Java** là một bản thiết kế của một lớp.
- Interface được sử dụng để định nghĩa **bộ hành vi** cho các đối tượng.
- Interface là một kỹ thuật để thu được tính trừu tượng hoàn toàn và đa kế thừa trong Java.

Chương 5. Interface

Interface và class



- Một interface được viết trong một file với định dạng .java, với tên của interface giống tên của file.
- Bytecode của interface được lưu trong file có định dạng .class.
- Khai báo interface trong một package, những file bytecode tương ứng cũng có cấu trúc thư mục có cùng tên package.

Chương 5. Interface

Interface và class



- Không thể khởi tạo một interface như class.
- Một interface **không chứa bất cứ hàm Constructor** nào.
- **Tất cả các phương thức** của interface đều **là abstract**.
- Một interface không thể chứa một trường dữ liệu nào trừ các trường vừa static và final.
- Một interface không thể kế thừa từ lớp, nó được triển khai bởi một lớp.
- Một interface có thể kế thừa từ nhiều interface khác.

Chương 5. Interface

Cách khai báo Interface



```
interface <tên interface> {  
    [<các thuộc tính;>]// final hoặc static  
    [<các phương thức;>] // abstract methods  
}
```

Chương 5. Interface

Cách khai báo Interface



Ví dụ 5.1:

```
public interface HìnhHoc {  
    double PI=3.14;// hằng  
    public double tinhChuVi();  
    public double tinhDienTich();  
}
```

Chương 5. Interface

Kế thừa từ Interface



- Sử dụng từ khóa implements
- Cú pháp:

```
public class <tên_lớp_con> implements <interface1>, <interface 2>, ...  
{  
    <cài đặt lớp>  
    // ghi đè các hành vi của các Interface  
}
```


Chương 5. Interface

Kế thừa từ Interface



Ví dụ 5.2

```
public class HìnhTron implements HìnhHoc{  
    private double r;  
    @Override  
    public double tinhChuVi() {  
        return 2*PI*r;    }  
    @Override  
    public double tinhDienTich() {  
        return PI*r*r;  
    }  
}
```

Chương 5. Interface

Lưu ý khi triển khai Interface



- Một lớp có thể triển khai một hoặc nhiều interface tại một thời điểm.
- Một lớp chỉ có thể kế thừa một lớp khác, nhưng được triển khai **nhiều interface**.
- Một interface có thể kế thừa từ một interface khác, tương tự cách một lớp có thể kế thừa lớp khác.
- Một đối tượng có thể vừa kế thừa từ một đối tượng khác vừa thực thi interface. Lúc đó quan hệ kế thừa được thể hiện trước, thực thi bộ hành vi sau.

Chương 5. Interface

Đa kế thừa trong Java



Ví dụ 5.3

```
interface Printable {  
    void print(); }  
interface Showable{  
    void show(); }  
class A7 implements Printable,Showable {  
    public void print() {  
        System.out.println("Hello");    }  
    public void show() {  
        System.out.println("Welcome"); }  
    public static void main(String args[]){  
        A7 obj = new A7();    obj.print();    obj.show();    }  
}
```

Chương 5. Interface

Kế thừa trong Interface



Ví dụ 5.4

```
public class HìnhVuong extends TuGiac implements
HìnhHoc{
    private double canh;
    @Override
    public double tinhChuVi() {
        return 4*canh;    }
    @Override
    public double tinhDienTich() {
        return canh*canh;    }
}
```

Chương 5. Interface

Abstract và interface



1) Abstract class có phương thức abstract (không có thân hàm) và phương thức non-abstract (có thân hàm).	Interface chỉ có phương thức abstract . Từ java 8, nó có thêm các phương thức default và static .
Abstract class không hỗ trợ đa kế thừa.	Interface có hỗ trợ đa kế thừa
Abstract class có các biến final, non-final, static and non-static .	Interface chỉ có các biến static và final .
Abstract class có thể cung cấp nội dung cài đặt cho phương thức của interface.	Interface không thể cung cấp nội dung cài đặt cho phương thức của abstract class.

Chương 5. Interface

Abstract và interface



5) Từ khóa abstract được sử dụng để khai báo abstract class.	Từ khóa interface được sử dụng để khai báo interface.
6. Thực thi: dùng từ khóa extends	Dùng từ khóa implements
7. Ví dụ: <pre>public abstract class Shape { public abstract void draw(); public void xuất(){ System.out.print(" shape"); } }</pre>	<pre>public interface Drawable { void draw(); }</pre>

Chương 5. Interface

Bài tập



- Trong một ứng dụng có quản lý bất động sản của công ty ABC có 4 đối tượng là: đất trống, nhà ở, biệt thự và khách sạn. Biết rằng tất cả các đối tượng này đều có các thông tin sau: mã số, chiều dài, chiều rộng và các phương thức tính giá trị. Biết rằng giá trị được tính như sau:
 - Đất trống: giá bán = diện tích * 10000.
 - Nhà ở có thêm thông tin về số lầu: giá bán = diện tích * 10000 + số lầu * 100000.
 - Khách sạn có thêm thông tin về số sao: giá bán = diện tích * 10000 + số sao * 50000.
 - Biệt thự có giá bán = diện tích * 400000.

Chương 5. Interface



Tuy nhiên trong 4 bất động sản trên thì có 2 loại sản phẩm khi mua khách hàng phải đóng thêm phí kinh doanh là biệt thự và khách sạn. Biết rằng phí kinh doanh sẽ được tính như sau:

- Biệt thự: diện tích * 1000.
- Khách sạn: chiều rộng * 5000.

Xây dựng ứng dụng quản lý bất động sản. Hãy tính tổng giá trị bất động sản đang có và phí kinh doanh phải đóng của các bất động sản trên.