

Lập trình Java

Tuần 4: Abstract class, Interface, thừa kế trong java

Nội dung

- Khái niệm abstract class, interface
- Tính kế thừa trong java

Tính kế thừa (inheritance)

- Java tồn tại một thứ bậc phân cấp gọi là cây kế thừa trong đó lớp bậc cao nhất được gọi là lớp cha (super class), các lớp bậc dưới là lớp con (sub class)
- Một lớp chỉ được kế thừa một supper class (đơn kế thừa)
- Class kế thừa class dùng từ khóa extends.

Tính kế thừa (inheritance)

- Khi kế thừa thì subclass có thể:
 - + Có các trường ở dạng public và protected của supper class.
 - + Nếu sub class và super class cùng package thì supper class có thêm các trường và phương thức ở dạng mặc định (default) của supper class.
- Tuy nhiên: sub class không kế thừa hàm tạo (constructor) và các trường private của supper class.

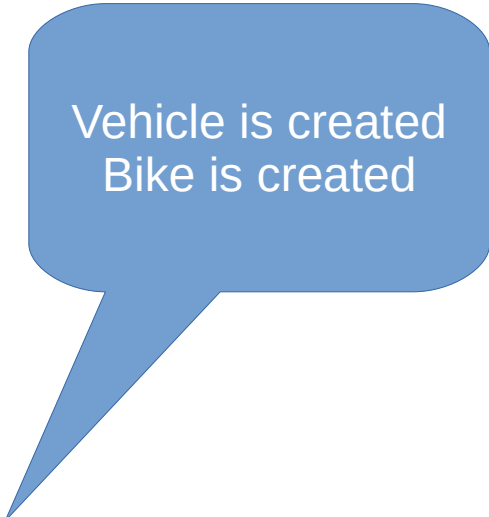
Từ khóa supper

- Dùng để truy cập các trường của supper class.
- Có thể dùng để gọi hàm tạo của supper class.
- Dùng để truy cập các phương thức của supper class.

```
class Vehicle{ int speed=50; }  
class Bike4 extends Vehicle{  
int speed=100;  
//bay gio se in speed cua Vehicle  
void display(){ System.out.println(super.speed);}  
public static void main(String args[]){  
    Bike4 b=new Bike4(); b.display(); } }
```

Từ khóa supper

```
class Vehicle {  
    Vehicle() {  
        System.out.println("Vehicle is created");  
    }  
}  
  
class Bike2 extends Vehicle {  
    Bike2() {  
        super() ; //gọi Constructor của lớp cha  
        System.out.println("Bike is created");  
    }  
  
    public static void main(String args[]) {  
        Bike2 b = new Bike2();  
    }  
}
```



Vehicle is created
Bike is created

Override

- Thể hiện khi 2 lớp kế thừa nhau, sub class sẽ kế thừa các phương thức có sẵn của supper class và có thể định nghĩa lại.
- Có thể sử dụng từ khóa supper để truy cập lại phương thức đã được ghi đè của supper class.
- Access modifier của phương thức trong lớp con phải có phạm vi bằng hoặc lớn hơn phạm vi của phương thức trong supper class.

Tính đa hình (polymorphism)

- Một tác vụ được thực hiện theo nhiều cách khác nhau, thể hiện qua việc gọi phương thức của đối tượng.
- Java dùng hai phương thức: overloading (cùng tên nhưng khác tham số) và overriding (lớp con kế thừa viết lại).

Tính trừu tượng trong Java

- Thể hiện ở việc quan tâm đến đối tượng thay vì cách nó thực hiện. Ví dụ: chu vi, diện tích là đặc trưng khi xét tới một hình (hình vuông, hình chữ nhật, hình tròn,...)
- Có 2 cách để trừu tượng hóa trong java:
 - + Abstract class
 - + interface

Khái niệm Abstract class, interface

Cách khai báo class, abstract class và interface

```
class Student {  
}
```

```
interface Person{  
}
```

```
abstract class Employee {  
}
```

Khái niệm Abstract class, interface

```
class Student {  
    private String name;  
    public void show(){}  
}
```

```
interface Person{  
    //các thuộc tính của interface phải là hằng số tĩnh  
    public final static String NAME = "A";  
    public void show();  
}  
  
abstract class Employee {  
    private String name;  
    public abstract void move();  
}
```

Khái niệm Abstract class, interface

Class kế thừa class dùng từ khóa extends.

Class kế thừa interface dùng từ khóa implements.

```
interface Person{  
String NAME = "A";  
public void show();  
}  
  
abstract class Employee {  
private String name;  
public abstract void move();  
}
```

```
class Person1 implements Person {  
//override tất cả phương thức  
}  
  
class Employee1 extends Employee {  
//override tất cả phương thức abstract  
}
```

Abstract class

- Không thể khởi tạo 1 đối tượng trực tiếp từ một abstract class.
- Phải chứa phương thức abstract.
- Phương thức abstract là phương thức không chứa thân hàm và kết thúc bằng dấu ;

Interface

- Java chỉ có thể kế thừa từ 1 lớp duy nhất trong cùng một thời điểm, nhưng có thể dẫn xuất cùng lúc nhiều Interfaces (đơn kế thừa)
- Không được phép có những phương thức cụ thể (concrete methods) hay tất cả các phương thức phải là phương thức abstract.
- interface cần phải được hiện thực (implements), các lớp đã implements interface buộc phải viết lại (overright) các phương thức của interface.
- Một interface có thể kế thừa từ một interface khác
- Tất cả thuộc tính trong interface phải là biến tĩnh (public static final), phương thức phải là public abstract (java sẽ tự hiểu nếu người dùng không khai báo)
- Interface không có Constructor

Interface

```
interface Printable{
    void print();
}
interface Showable{
    void print();
}

class TestTnterface1 implements Printable,Showable{
    public void print() {
        System.out.println("Hello");
    }

    public static void main(String args[]) {
        TestTnterface1 obj = new TestTnterface1();
        obj.print();
    }
}
```

```
interface Printable{
    void print();
}
interface Showable extends Printable{
    void show();
}
class Testinterface2 implements Showable{

    public void print() {
        System.out.println("Hello");
    }
    public void show() {
        System.out.println("Welcome");
    }

    public static void main(String args[]){
        Testinterface2 obj = new Testinterface2();
        obj.print();
        obj.show();
    }
}
```

So sánh abstract class và interface

- Interface thay thế cho việc đa kế thừa. Java không cho phép đa kế thừa (Multiple Inheritance) (cho dù là class hay abstract class)
- Trong abstract class có thể có phương thức trừu tượng và cả phương thức bình thường, có constructor.

So sánh abstract class và interface

- Câu hỏi:
 - + Tại sao Java không cho phép đa kế thừa?
 - + Tại sao interface lại được đa kế thừa.

```
class A {  
    void msg() {  
        System.out.println("Hello");  
    }  
}  
class B {  
    void msg() {  
        System.out.println("Welcome");  
    }  
}  
public class C extends A,B {  
    public static void main(String args[]) {  
        C obj = new C();  
        obj.msg();  
    }  
}
```



Error

Bài tập tuần 4

Xây dựng lớp ChuNhat gồm 2 thuộc tính là rong và dai và các phương thức getChuVi() và getDienTich() để tính chu vi và diện tích. Phương thức xuất() sẽ xuất ra màn hình chiều rộng, chiều dài, diện tích và chu vi.

Xây dựng lớp Vuong kế thừa từ lớp ChuNhat và ghi đè phương thức xuất() để xuất thông tin cạnh, diện tích và chu vi.

Viết chương trình nhập 2 hình chữ nhật và một hình vuông sau đó xuất ra màn hình.