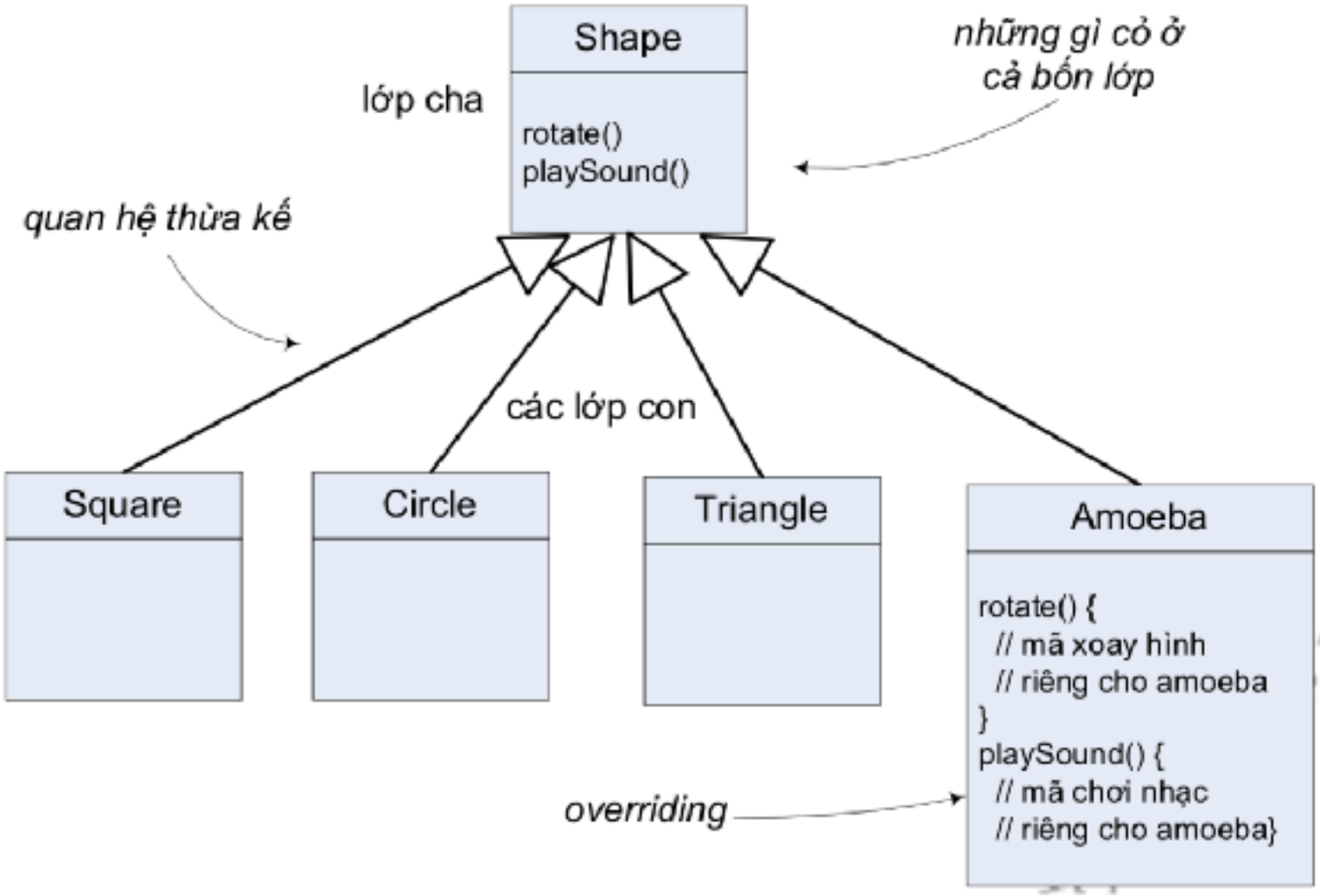
LẬP TRÌNH JAVA

Bài 9: Thừa kế trong java


Shape
lớp cha
rotate()
playSound()
những gì có ở
cả bốn lớp
quan hệ thừa kế
các lớp con
Square
Circle
Triangle
Amoeba
rotate() {
// mã xoay hình
// riêng cho amoeba
}
playSound() {
// mã chơi nhạc
// riêng cho amoeba}
overriding

Khái niệm

- Tính kế thừa trong Java là môt kỹ thuật mà trong đó một đối tượng thu được tất cả thuộc tính và hành vi của đối tượng cha
- Đặt các phần mã dùng chung tại một lớp và coi đó là lớp cha – lớp dùng chung trừu tượng hơn, các lớp cụ thể hơn là các lớp con.
- Khi chúng ta nói về tính kế thừa, từ khóa thường xuyên nhất được sử dụng là **extends** và **implements** trong java

Tại sao sử dụng tính kế thừa trong Java?

- Để làm tăng tính tái sử dụng của code.
- Để ghi đè phương thức (Method Overriding), do đó có thể thu được tính đa hình tại runtime.

Các loại đối tượng trong Java

- **Class**: Là một lớp đầy đủ các phương thức và thuộc tính.

- **Abstract Class**: Là một lớp trừu tượng bao gồm các thuộc tính và phương thức, trong đó một số phương thức có thể không có thân hàm. Bất kỳ lớp nào thừa kế bắt buộc phải ghi đè các phương thức không có thân hàm.

- **Interface**: Là một lớp trừu tượng chỉ có các phương thức không có thân hàm, lớp con phải ghi đè tất cả phương thức của interface đã định nghĩa.

Từ khóa extends và implements

extends : kế thừa các class

- Chúng ta sử dụng từ khóa extends của lớp con để có thể kế thừa các thuộc tính và phương thức của lớp cha trừ các thuộc tính và phương thức private của lớp cha.
- Các phương thức của lớp cha có thể ghi đè (override) hoặc không.
- Từ khóa **super** để gọi phương thức của lớp cha.

implements : kế thừa các interface

- Khi kế thừa interface, bắt buộc phải ghi đè tất cả phương thức của interface.

Từ khóa extends và implements

```
1  class A {
2    public void show() {
3      System.out.println("show");
4    }
5  }
6
7  class B extends A {
8    public void display() {
9      System.out.println("display");
10   }
11
12   public void show() {
13     System.out.println("better show");
14   }
15 }
16
17 public class Main {
18
19   public static void main(String[] args) {
20
21     A a = new B(); // possible because B extends A
22     a.show(); // this will now call to show() method of class B
23   }
24 }
25
26 Output
27 better show
```

Từ khóa extends và implements

```
1 class R implements Runnable{
2 public void run(){
3    System.out.println("do nothing");
4   }
5 }
```

Một số điểm lưu ý

1) Một class **có thể extends** từ một **class** khác, **không thể extends** từ một **interface.**

```
1  interface I{
2
3  }
4
5  class A{
6  }
7
8  class B extends A{
9
10 }
11
12 class C extends I{
13
14 }
```

```
1 $ javac Main.java
2 Main.java:27: no interface expected here
3 class C extends I{
4 ^
5 1 error
```

Một số điểm lưu ý

2) Khi sử dụng **extends**, một **class** chỉ có thể kế thừa từ **một class** khác

```
1 class C extends A, B{
2
3 }
4
5 $ javac Main.java
6 Main.java:27: '{' expected
7 class C extends A, B{
8 ^
9 1 error
```

Một số điểm lưu ý

3) Một **interface** có thể **extends** từ **một interface** khác.

```
1 interface J extends I{
2
3 }
```

Một số điểm lưu ý

4) Một class có thể kế thừa từ nhiều interface khác bằng cách sử dụng implements

```
1 class D implements I, J, K{
2
3 }
```

Một số điểm lưu ý

5) Trong một lớp có thể sử dụng extends và implements

```
1 class E extends A implements I, J{
2
3 }
```

Một số điểm lưu ý

6) Một interface không thể implements từ một interface

```
1 interface L implements J{
2
3 }
4
5 javac Main.java
6 Main.java:49: '{' expected
7 interface L implements J{
8 ^
9 1 error
```

Bài tập

1) Định nghĩa một lớp Animal chứa các phương thức

- Phương thức run() để in ra tốc độ chạy của con vật đó.
- Phương thức print() để in ra tên – loại động vật.

Cài đặt lớp Dog và Cat kế thừa Animal chứa các thuộc tính tên, loại, tốc độ chạy và ghi đè các phương thức ở lớp Animal.

Xây dựng lớp AnimalTest để thực hiện các công việc:

- Nhập vào 10 con vật (cho phép chọn Dog hoặc Cat)
- In ra màn hình các thông số thông qua 2 phương thức run() và print()
- Sắp xếp 10 con vật đó theo tốc độ chạy giảm dần

Bài tập

2) . Xây dựng một interface có tên là HCNInterface chứa một số phương thức sau:
- Phương thức dientichHCN() dùng để tính diện tích hình chữ nhật.
- Phương thức getChieuDai() và getChieuRong() dùng để lấy giá trị chiều dài và chiều rộng của của hình chữ nhật.
- Phương thức setDaiRong(cd, cr) dùng để cập nhật giá trị cd, cr cho hai cạnh của hình chữ nhật.

Sử dụng interface HCNInterface trên để xây dựng lớp Hinhchunhat chứa hai thuộc tính là: chieudai, chieurong và các phương thức dientichHCN(), getChieuDai(), getChieuRong(), setDaiRong(cd,cr) đã khai báo trong Interface HCNInterface trên?

Xây dựng lớp HCNTest sử dụng lớp Hinhchunhat chứa phương thức main thực hiện các công việc sau:
a. Khai báo một mảng chứa n hình chữ nhật (với n là số nguyên dương bất kỳ được nhập từ bàn phím). Sau đó nhập chiều dài và chiều rộng cho n hình chữ nhật đó.
b. In ra màn hình thông tin: chiều dài, chiều rộng và diện tích của n hình chữ nhật trên.
c. In ra màn hình thông tin về hình chữ nhật có diện tích lớn nhất.