

Bài 44: Lớp cha, lớp con

- ✓ Các khái niệm
- ✓ Những gì được kế thừa?
- ✓ Java chỉ hỗ trợ kế thừa đơn
- ✓ Khai báo kế thừa
- ✓ Ví dụ và bài tập thực hành

Các khái niệm

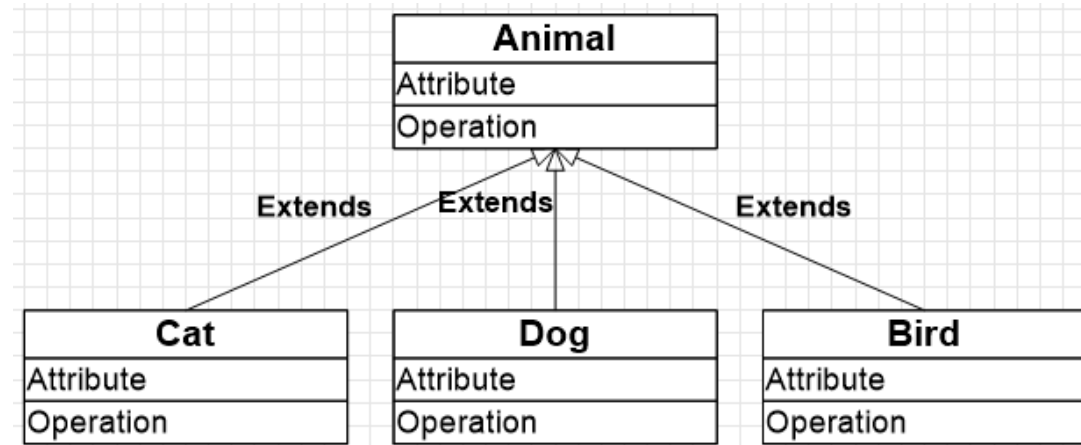
- Kế thừa(inheritance) là đặc trưng thứ hai của ngôn ngữ lập trình hướng đối tượng
- Kế thừa là kĩ thuật trong Java cho phép một lớp kế thừa lại những tính năng(trường, phương thức) của lớp khác
- Ta còn gọi tính chất kế thừa là mở rộng từ một lớp khác(extend)
- Các khái niệm liên quan đến kế thừa:
 - Lớp cha: còn gọi là super class, base class, là các lớp được các lớp khác kế thừa
 - Lớp con: còn gọi là sub class, derived class, là các lớp đi kế thừa lớp khác. Lớp con có thể bổ sung các đặc trưng của riêng nó.

Các khái niệm

- Các khái niệm liên quan đến kế thừa:
 - Tính tái sử dụng(reuseability): kế thừa hỗ trợ tái sử dụng lại code. Ta có thể tạo một lớp mới từ các lớp đã có và bổ sung thêm các thành phần cần thiết
 - Tính tổng quát hóa: lớp cha càng trên cao trong chuỗi kế thừa thì càng tổng quát, chứa càng ít thông tin nhưng là các thông tin chung nhất mà mọi lớp con của nó đều có
 - Tính cụ thể hóa: lớp con càng ở sâu phía dưới trong chuỗi kế thừa thì càng có những đặc trưng riêng cụ thể mà lớp cha không có
 - Quan hệ is a(là một): lớp con luôn là một ví dụ khác cụ thể hơn của lớp cha. Ví dụ: người là động vật, mèo là động vật, cá là một loài động vật

Ví dụ

➤ Hình ảnh kế thừa:



➤ Mục đích sử dụng của kế thừa: tái sử dụng lại code, tiết kiệm thời gian, chi phí.

Lớp con kế thừa những gì?

- Khi một lớp con nào đó kế thừa một lớp cha, nó sẽ kế thừa tất cả các thành phần(trường, phương thức) public và protected của lớp cha
- Khi đó các thành phần được kế thừa sẽ có thể truy cập một cách trực tiếp trong lớp con giống như chính lớp con khai báo các thành phần đó vậy
- Các thành phần không có access modifier của lớp cha sẽ chỉ có thể được truy cập trực tiếp trong lớp con nếu lớp con nằm cùng gói với lớp cha

Những gì không được kế thừa?

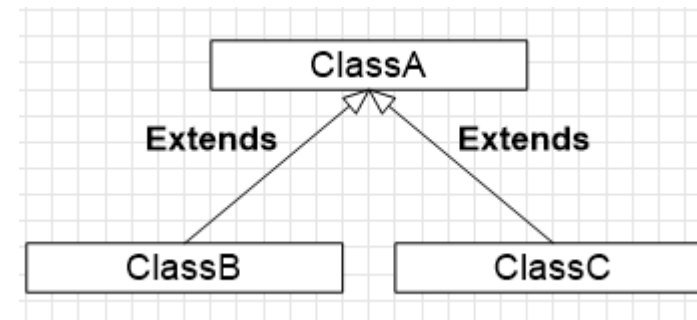
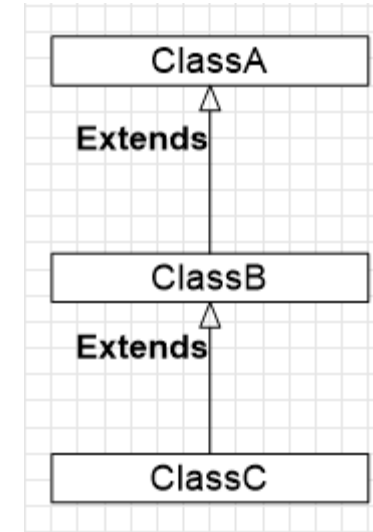
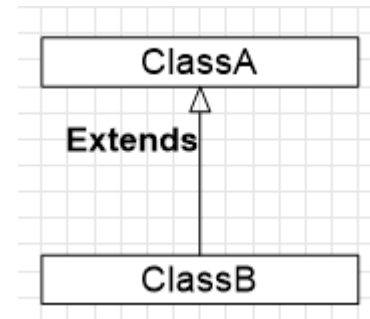
- Các thành phần private của lớp cha không được kế thừa bởi lớp con nhưng có thể được sử dụng qua các getter/setter
- Các phương thức khởi tạo của lớp cha không được kế thừa bởi lớp con. Lớp con có thể gọi đến phương thức khởi tạo của lớp cha qua keyword `super`
- Như vậy tính chất kế thừa không làm ảnh hưởng đến tính chất đóng gói dữ liệu vì lớp con không thể truy cập trực tiếp vào thành phần private của lớp cha

Java chỉ có kế thừa đơn

- Java chỉ hỗ trợ kế thừa đơn với lớp. Tức mỗi lớp chỉ có tối đa 1 lớp cha trực tiếp
- Nếu muốn đa kế thừa ta sẽ sử dụng interface
- Lý do không hỗ trợ đa kế thừa: phức tạp. Ví dụ nhiều lớp cha có cùng phương thức với cùng tên, tham số và chức năng. Vậy lúc này chương trình sẽ gọi phương thức nào để sử dụng?
- Các dạng kế thừa:
 - Kế thừa đơn: 1 lớp cha 1 lớp con
 - Kế thừa nhiều bậc: 1 lớp con đồng thời là 1 lớp cha của lớp khác
 - Kế thừa phân cấp: một lớp có thể là cha của nhiều hơn 1 lớp con khác

Ví dụ

➤ Các dạng kế thừa: đơn, nhiều bậc và phân cấp.



Khai báo kế thừa

- Sử dụng keyword **extends** để thể hiện mối quan hệ kế thừa:

```
access non-access class A extends B {  
    // ...  
}
```

- Trong đó:

- A là lớp con, đứng trước keyword **extends**
- B là lớp cha, đứng sau keyword **extends**
- Keyword **extends** là bắt buộc phải có
- Lớp A có thể bổ sung các thành phần riêng của chính nó

Ví dụ

➤ Ví dụ sau thể hiện kế thừa đơn giữa lớp Cat và Animal:

```
public class Animal {  
    private String name;  
    private float weight;  
  
    public void move() {  
        // do something...  
    }  
  
    public void eat() {  
        // do something...  
    }  
}  
  
class Cat extends Animal {  
    private String color;  
    private String favoriteFood;  
  
    public void catchMouse() {  
        // do something...  
    }  
}
```



Minh họa cụ thể

➤ Thao tác tạo lớp kế thừa trong công cụ lập trình

Nội dung tiếp theo

Ép kiểu trong mối quan hệ kế thừa

