Bài 50: Các chuẩn thiết kế kế thừa

- ✓ 6 chuẩn khi kế thừa trong Java

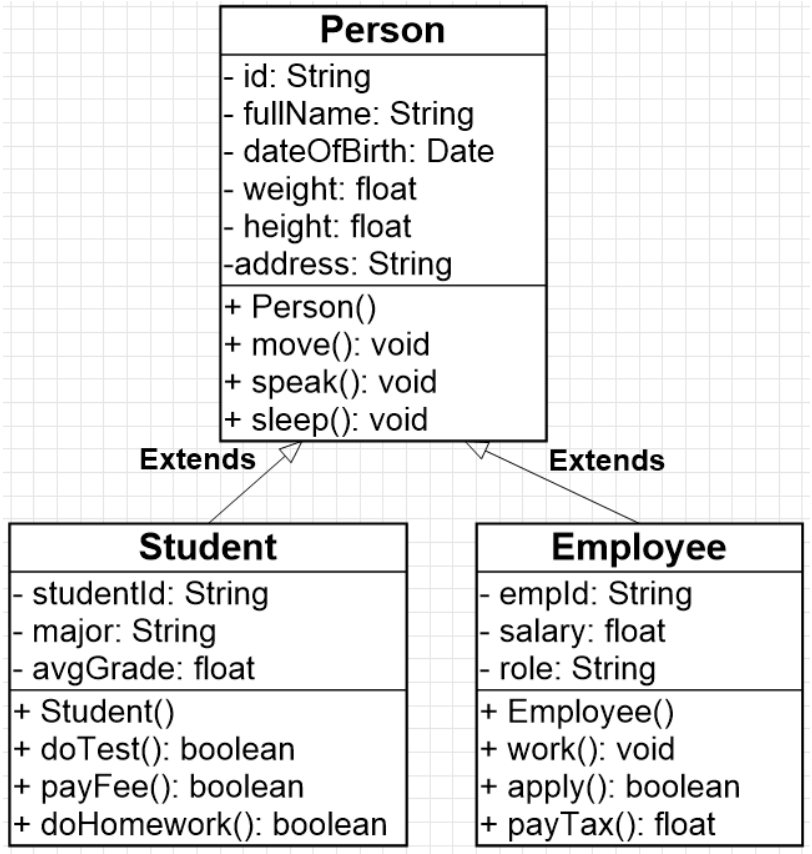
Chuẩn 1

Đặt các thành phần chung nhất ở lớp cha

- Tất cả các lớp con kế thừa từ 1 lớp cha sẽ sử dụng được các thành phần chung của lớp cha
- Do vậy ta để các phương thức, các trường dữ liệu chung nhất có ở tất cả các lớp con vào lớp cha
- Lớp con sau đó chỉ cần kế thừa lại lớp cha và bổ sung các thông tin đặc trưng của riêng mình

Ví dụ

➢Ví dụ về lớp cha, lớp con:



Chuẩn 2

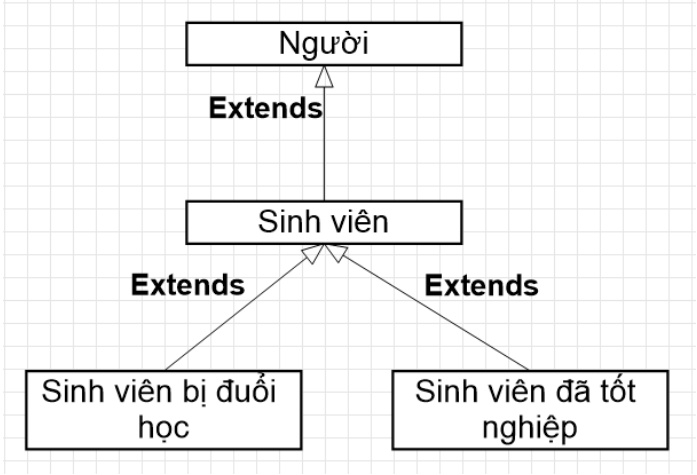
Không sử dụng trường protected

- Tất cả dữ liệu của lớp cần tuân thủ tính đóng gói dữ liệu
- Các trường protected thuận tiện cho việc truy cập từ lớp con nhưng không an toàn
- Kẻ xấu có thể lợi dụng đặc điểm của thành phần protected để khai báo lớp con hoặc thêm một lớp/sử dụng lớp trong cùng gói với lớp cha để truy cập vào các trường protected này
- Các phương thức protected thường hữu ích
- Khi muốn sử dụng phải override lại bởi các lớp con

Chuẩn 3

Sử dụng kế thừa để mô hình hóa mối quan hệ is-a

- Chỉ sử dụng kế thừa để thể hiện mối quan hệ is-a
- Không lạm dụng kế thừa chỉ vì muốn tái sử dụng lại code
- Ví dụ sau là hợp lệ:



Chuẩn 3

- Ví dụ về lạm dụng việc kế thừa:
- Bảng điểm chứa thông tin về điểm, học lực và thông tin sinh viên. Ta có thể cho bảng điểm kế thừa từ sinh viên sau đó dễ dàng lấy được các thông tin sinh viên. Nhưng bản chất của bảng điểm không phải là(is a) sinh viên và ngược lại
- Do đó ta không lạm dụng kế thừa chỉ vì lười code

Chuẩn 4

Chỉ kế thừa khi mọi phương thức được kế thừa đều có ý nghĩa

- Ví dụ lớp người có các hành vi: ăn, uống, nghỉ ngơi, giao tiếp, làm việc.
- Lớp nhân viên kế thừa người và có thêm hành vi như chấm công, tính lương, nhận lương, đi công tác, nhận nhiệm vụ, thăng chức…
- Ta muốn tạo lớp sinh viên kế thừa 1 trong 2 lớp trên. Rõ rang ta không kế thừa từ nhân viên mà kế thừa từ người
- Sau đó bổ sung them các hành vi đặc trưng của sinh viên: làm bài thi, làm bài tập, đăng kí môn học

Chuẩn 5

Không thay đổi mục đích sử dụng của phương thức override

- Khi override ta cần giữ nguyên mục đích sử dụng của phương thức gốc trong lớp cha
- Không tự ý thay đổi phương thức đó để biểu diễn một hành vi khác, mục đích khác
- Để đảm bảo tính nhất quán của phương thức trong chuỗi kế thừa
- Ví dụ: không dùng phương thức move để thể hiện việc speak hay eat

Chuẩn 6

Sử dụng tính đa hình thay cho kiểu cụ thể

- Khi bạn gặp một vấn đề dạng:
  - Nếu x là đối tượng kiểu A thì thực hiện hành động 1
  - Nếu x là đối tượng kiểu B thì thực hiện hành động 2
- Hãy nghĩ đến việc sử dụng tính chất đa hình: dùng kiểu lớp cha để tham chiếu đến đối tượng của các lớp con
- Khi chương trình hoạt động nó sẽ dựa vào đối tượng được tham chiếu để quyết định thực hiện hành động tương ứng
- Code sử dụng tính đa hình dễ bảo trì, mở rộng

Ví dụ

```
public class Animal {
    public void move() {
        System.out.println("Animal is is moving...");
    }
}

class Fish extends Animal {
    @Override
    public void move() {
        System.out.println("Fish is moving by swimming...");
    }
}

class  Cat extends Animal {
    @Override
    public void move() {
        System.out.println("Cat is moving by running on the ground...");
    }
}

class Bird extends Animal {
    @Override
    public void move() {
        System.out.println("Bird is moving by flying...");
    }
}
```

Ví dụ

➢Tại nơi sử dụng:

```
public class Test {
    public static void main(String[] args) {
        Animal fish = new Fish();
        Animal cat = new Cat();
        Animal bird = new Bird();
        Animal animal = new Animal();

        // gọi phương thức override sử dụng tính đa hình
        animal.move();
        fish.move();
        cat.move();
        bird.move();
    }
}
```

Nội dung tiếp theo

Lớp trừu tượng