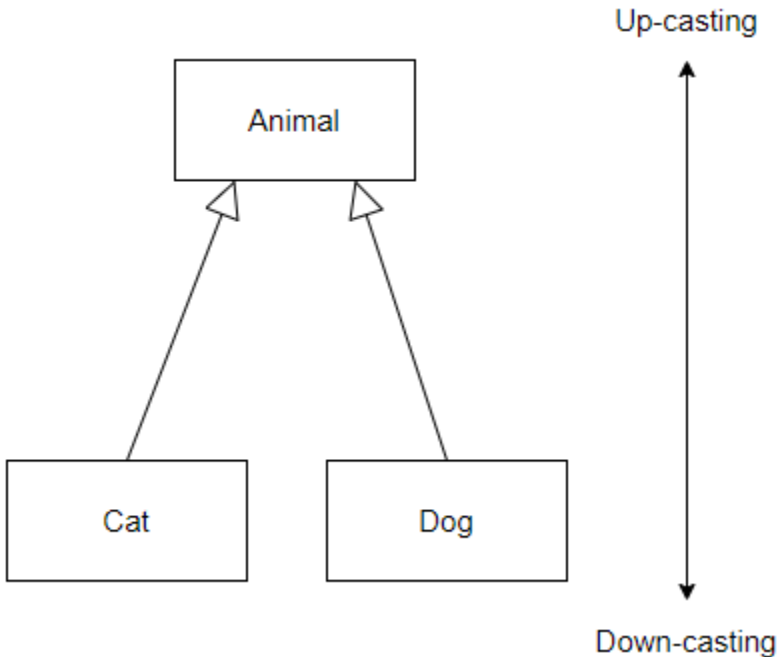


Lý thuyết

Để làm được bài này bạn cần hiểu về khái niệm **Up-casting** và **Down-casting** trong Java.



Up-casting

Khi biến của lớp cha tham chiếu tới đối tượng của lớp con (đối tượng của lớp con bị chuyển kiểu dữ liệu về lớp cha), thì đó được gọi là up-casting. Ví dụ:

```
class Animal {
    public void sound() {
        System.out.println("some sound");
    }
}

class Cat extends Animal {
    public void sound() {
        System.out.println("meow meow");
    }
}

class Dog extends Animal {
```

```

        public void sound() {
            System.out.println("woof woof");
        }
    }

    public class Entry {
        public static void main(String[] args) {
            // Up-casting
            Animal animal1 = new Cat();
            animal1.sound();
            // Up-casting
            Animal animal2 = new Dog();
            animal2.sound();
        }
    }
}

```

Kết quả khi chạy chương trình:

```

meow meow
woof woof

```

Có thể thấy đối tượng thuộc lớp `Cat` và lớp `Dog` đã được gán cho biến thuộc lớp `Animal` (chuyển kiểu về lớp `Animal`), đó chính là up-casting.

Lưu ý: khi biến của lớp cha tham chiếu tới đối tượng của lớp con thì biến này chỉ có thể gọi tới các thuộc tính và phương thức **có ở lớp cha** và nếu lớp con ghi đè thì phương thức được gọi sẽ ở lớp con. Ví dụ:

```
Entry.java
1 package OOP;
2
3 class Animal {
4     public void sound() {
5         System.out.println("some sound");
6     }
7 }
8
9 class Dog extends Animal {
10    public void sound() {
11        System.out.println("woof woof");
12    }
13    public void play() {
14        System.out.println("The dog is playing");
15    }
16 }
17
18 public class Entry {
19    public static void main(String[] args) {
20        Animal animal = new Dog();
21        animal.play();
22    }
23 }
24
```

The method play() is undefined for the type Animal

2 quick fixes available:

- Create method 'play()' in type 'Animal'
- Add cast to 'animal'

Chương trình này sẽ báo lỗi do phương thức `play()` không có ở lớp cha và nếu thay `animal.play()` thành `animal.sound()` thì chương trình sẽ hiển thị ra "woof woof" thay vì "some sound" (do lớp con ghi đè phương thức `sound()` từ lớp cha).

Quay lại với lỗi trong chương trình trên, biến `animal` trở tới được đối tượng thuộc lớp `Dog` nhưng lại không thể gọi tới phương thức `play()` là vì khi chương trình biên dịch code, trình biên dịch sẽ chỉ biết về kiểu dữ liệu được khai báo chứ không biết về đối tượng mà nó đang trỏ tới, và rõ ràng `animal` không có phương thức `play()` nên sẽ không thể biên dịch được. Để gọi tới được phương thức `play()`, bạn cần thực hiện down-casting.

Down-casting

Bạn hãy hiểu đơn giản rằng down-casting ngược với up-casting, up-casting là chuyển từ lớp con sang lớp cha thì down-casting là chuyển từ lớp cha sang lớp con. Như ở ví dụ trên thì bạn có thể gọi tới phương thức `play()` từ biến `animal` giống như sau:

```
class Animal {
    public void sound() {
        System.out.println("some sound");
    }
}

class Dog extends Animal {
    public void sound() {
        System.out.println("woof woof");
    }
    public void play() {
        System.out.println("The dog is playing");
    }
}

public class Entry {
    public static void main(String[] args) {
        Animal animal = new Dog();
        // Down-casting
        ((Dog)animal).play();
    }
}
```

Kết quả khi chạy chương trình:

```
The dog is playing
```

Hoặc bạn cũng có thể thực hiện down-casting giống như sau:

```
Animal animal = new Dog();
Dog dog = (Dog)animal;
```

```
dog.play()
```