

Bài 48: Overloading vs Overriding

- ✓ Khái niệm
- ✓ Sự khác nhau
- ✓ Ví dụ minh họa

Khái niệm

- Overloading hay nạp chồng là kỹ thuật trong Java cho phép hai hay nhiều phương thức có cùng tên nhưng phải khác tham số
- Overriding hay ghi đè là kỹ thuật trong Java cho phép lớp con định nghĩa lại phương thức của lớp cha theo cách của lớp con

Sự khác nhau

Nạp chồng	Ghi đè
Thuộc loại đa hình tại thời điểm biên dịch chương trình	Thuộc loại đa hình tại thời điểm chạy chương trình
Cung cấp cùng chức năng cho nhiều kiểu tham số khác nhau	Tăng mức độ tương thích của phương thức lớp cha trong ngữ cảnh sử dụng của lớp con
Thường xảy ra trong một lớp độc lập hoặc trong kế thừa	Chỉ xảy ra tại lớp con trong mối quan hệ kế thừa
Bắt buộc cùng tên nhưng phải khác tham số	Phải giữ nguyên bản dấu hiệu nhận biết như phương thức gốc trong lớp cha

Sự khác nhau

Nạp chồng	Ghi đề
Kiểu trả về có thể giống hoặc khác nhau nhưng tham số phải khác nhau(về kiểu, số lượng, thứ tự)	Kiểu trả về giữ nguyên như trong phương thức gốc
Có thể nạp chồng mọi phương thức không bị giới hạn bởi access hay non-access modifier	Chỉ xảy ra ở các phương thức được kế thừa lại của lớp cha và có thể truy cập một cách trực tiếp từ lớp con. Trừ các phương thức private, final, static

Ví dụ overloading

```
public class MyCalculator {  
    // phương thức overload với tham số là int  
    public static int add(int a, int b) {  
        return a + b;  
    }  
  
    // phương thức overload với tham số là long  
    public static long add(long a, long b) {  
        return a + b;  
    }  
  
    // phương thức overload với tham số là double  
    public static double add(double a, double b) {  
        return a + b;  
    }  
  
    public static void main(String[] args) {  
        // sử dụng phương thức overload  
        double sumDouble = add(3.25, 7.85);  
        long sumLong = add(15536L, 15454L);  
        int sumInt = add(1314, 578);  
        //...  
    }  
}
```

Ví dụ overriding

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



Minh họa cụ thể

➤ Thực hiện trong công cụ lập trình

Nội dung tiếp theo

**Lớp Object và các phương
thức quan trọng**