

 Cập nhật tháng 8 năm 2024

[Bài đọc] Ghi đè phương thức - Overriding

1. Khái niệm:

Ghi đè phương thức (Method Overriding) là một tính năng trong Java cho phép lớp con cung cấp cách triển khai riêng của nó cho một phương thức đã được định nghĩa trong lớp cha. Khi ghi đè, phương thức trong lớp con có cùng tên, cùng tham số và cùng kiểu trả về như phương thức trong lớp cha.

2. Điều kiện để ghi đè phương thức

- **Tên phương thức:** Phải giống với tên phương thức trong lớp cha.
- **Tham số:** Phải giống hệt về số lượng, kiểu dữ liệu và thứ tự.
- **Kiểu trả về:** Phải giống hoặc là kiểu con (covariant return type) của kiểu trả về phương thức lớp cha.
- **Từ khóa:** Phương thức trong lớp cha không được khai báo với từ khóa `final` hoặc `static`.
- **Mức truy cập:** Không được giảm cấp độ truy cập (có thể mở rộng nhưng không được thu hẹp). Ví dụ:
 - Lớp cha: `protected`, lớp con: `public` (hợp lệ).
 - Lớp cha: `public`, lớp con: `protected` (không hợp lệ).

3. Lợi ích của ghi đè phương thức

- **Đa hình (Polymorphism):** Cho phép một phương thức có hành vi khác nhau dựa trên lớp thực thi.
- **Tái sử dụng mã:** Lớp con có thể sử dụng lại phương thức lớp cha nhưng cung cấp cách triển khai cụ thể.
- **Tăng tính linh hoạt:** Dễ dàng mở rộng hoặc sửa đổi hành vi của phương thức mà không cần thay đổi lớp cha.

4. Sử dụng từ khóa `@Override`

Từ khóa `@Override` được sử dụng trước phương thức ghi đè để:

- Đảm bảo rằng phương thức trong lớp con thực sự ghi đè phương thức trong lớp cha.
- Tránh lỗi khi sai cú pháp hoặc tên phương thức không khớp.

5. Ví dụ minh họa

Ví dụ 1: Ghi đè cơ bản

```
class Parent {  
    void display() {  
        System.out.println("Phương thức trong lớp cha");  
    }  
}  
  
class Child extends Parent {  
    @Override  
    void display() {  
        System.out.println("Phương thức được ghi đè trong lớp con");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Parent obj = new Child();  
        obj.display(); // Kết quả: Phương thức được ghi đè trong lớp con  
    }  
}
```

Ví dụ 2: Covariant Return Type

```
class Parent {  
    Parent getInstance() {  
        return this;  
    }  
}  
  
class Child extends Parent {  
    @Override  
    Child getInstance() {  
        return this;  
    }  
}
```

6. Sự khác biệt giữa Ghi đè (Overriding) và Nạp chồng (Overloading)

Tiêu chí	Overriding (Ghi đè)	Overloading (Nạp chồng)
Lớp liên quan	Xảy ra giữa các lớp (lớp cha và lớp con).	Xảy ra trong cùng một lớp.

Tên phương thức	Giống nhau.	Giống nhau.
Tham số	Phải giống hệt.	Phải khác nhau về số lượng, kiểu hoặc thứ tự.
Thời gian quyết định	Thời gian chạy (Run-time).	Thời gian biên dịch (Compile-time).
Kiểu trả về	Phải giống hoặc kiểu con (Covariant).	Không yêu cầu phụ thuộc vào kiểu trả về.

7. Các quy tắc quan trọng

8. **Không thể ghi đè phương thức static:** Vì phương thức static liên kết với lớp, không phải đối tượng. Thay vào đó, có thể sử dụng phương pháp *hiding*.
9. **Không thể ghi đè phương thức final:** Do phương thức final không thể bị thay đổi.
10. **Không thể ghi đè phương thức private:** Vì phương thức private không được kế thừa.

8. Ứng dụng trong thực tế

- **Tùy chỉnh hành vi lớp con:** Ví dụ, trong một ứng dụng giao thông, lớp con có thể ghi đè phương thức để chỉ định loại phương tiện (xe hơi, xe đạp, v.v.).
- **Xây dựng API linh hoạt:** Ví dụ, trong Java Collection Framework, lớp ArrayList ghi đè nhiều phương thức từ lớp cha AbstractList.

Link tài nguyên đọc thêm: <https://www.geeksforgeeks.org/overriding-in-java/>