Cập nhật tháng 8 năm 2024

# [Bài đọc] Nạp chồng phương thức - Overloading

1. **Khái niệm:**
Nạp chồng phương thức (Overloading) là một tính năng trong Java cho phép một lớp định nghĩa nhiều phương thức có cùng tên nhưng khác nhau về tham số. Điều này mang lại tính linh hoạt khi thiết kế chương trình, giúp người dùng gọi cùng một tên phương thức nhưng với các cách sử dụng khác nhau.

2. **Điều kiện để nạp chồng phương thức:**
Hai hoặc nhiều phương thức được coi là nạp chồng nếu:
- Có cùng tên.
- Khác nhau về **số lượng tham số** hoặc **kiểu dữ liệu của tham số** hoặc cả hai.

Lưu ý: Kiểu trả về (return type) **không phải** là tiêu chí để phân biệt phương thức nạp chồng.

3. **Cách nạp chồng phương thức:**
Các cách phổ biến để nạp chồng phương thức:
1. **Thay đổi số lượng tham số.**

```
class Calculator {
    int add(int a, int b) {
        return a + b;
    }
    int add(int a, int b, int c) {
        return a + b + c;
    }
}
```

2. **Thay đổi kiểu dữ liệu của tham số.**

```
class Calculator {
    int add(int a, int b) {
        return a + b;
    }
    double add(double a, double b) {
        return a + b;
    }
}
```

3. **Thay đổi thứ tự các tham số** (với các kiểu dữ liệu khác nhau).

```
class Printer {
    void print(String str, int number) {
        System.out.println(str + ": " + number);
    }
    void print(int number, String str) {
        System.out.println(number + " - " + str);
    }
}
```

4. **Ưu điểm của nạp chồng phương thức:**

- **Tính dễ đọc và bảo trì**: Cùng một tên phương thức với các phiên bản khác nhau giúp chương trình dễ hiểu hơn.
- **Tái sử dụng mã**: Giảm thiểu việc tạo ra các phương thức khác nhau với các tên khác nhau cho cùng một mục đích.
- **Tính linh hoạt**: Cung cấp cách xử lý đa dạng cho các tình huống khác nhau mà không cần thay đổi tên phương thức.

5. **Một số ví dụ thực tế:**

Ví dụ 1: Nạp chồng trong lớp Math (giả lập).

```java
class MathOperations {
    int multiply(int a, int b) {
        return a * b;
    }
    double multiply(double a, double b) {
        return a * b;
    }
}
```

Ví dụ 2: Xử lý in thông tin trong một lớp.

```java
class Display {
    void show(String message) {
        System.out.println("Message: " + message);
    }
    void show(int number) {
        System.out.println("Number: " + number);
    }
}
```

Link tài nguyên đọc thêm: https://www.geeksforgeeks.org/method-overloading-in-java/

**Danh sách các bài học**