Static & final

A. Thế nào là static?

Định nghĩa:

Từ khóa static trong Java được sử dụng để khai báo thuộc tính (biến) hoặc phương thức mà thuộc về lớp chứ không phải về một thể hiện (instance) của lớp đó. Điều này có nghĩa là tất cả các đối tượng của lớp đều chia sẻ cùng một thuộc tính hoặc phương thức static.

Khi nào sử dụng?:

Sử dụng static khi bạn muốn có một thuộc tính hoặc phương thức chung cho tất cả các đối tượng của lớp.

Ví dụ: Các hằng số hoặc các phương thức tiện ích (utility methods) mà không cần một thể hiện cụ thể.

Cách truy cập:

Để truy cập thuộc tính hoặc phương thức static, bạn có thể sử dụng tên lớp theo cú pháp:

ClassName.staticMethod();

ClassName.staticVariable;

Vd:

public class StaticExample {

static int staticCount = 0; // Thuộc tính static

static void incrementCount() { // Phương thức static

staticCount++;

}

public static void main(String[] args) {

StaticExample.incrementCount(); // Gọi phương thức static

System.out.println("Giá trị của staticCount: " + StaticExample.staticCount); // Kết quả: 1

}

}

B. Thế nào là final?

Định nghĩa:

Từ khóa final trong Java được sử dụng để khai báo biến, phương thức hoặc lớp không thể thay đổi sau khi đã được khởi tạo hoặc định nghĩa.

Sự khác biệt giữa final và static:

final không cho phép thay đổi giá trị của biến sau khi đã khởi tạo, trong khi static chỉ định rằng thuộc tính hoặc phương thức thuộc về lớp chứ không thuộc về thể hiện.

Biến được khai báo là final có thể là static, nhưng không nhất thiết. Nếu là static, nó có thể được truy cập qua lớp mà không cần một thể hiện.

Biến khai báo bằng final có thể chỉnh sửa được không?:

Không, một khi biến được khai báo là final, giá trị của nó không thể thay đổi.

Vd:

public class FinalExample {

final int finalValue = 10; // Biến final

void changeValue() {

// finalValue = 20; // Lỗi biên không thể thay đổi giá trị

}

public static void main(String[] args) {

FinalExample example = new FinalExample();

System.out.println("Giá trị của finalValue: " + example.finalValue); // Kết quả: 10

}

}