

## Static & final

1. Thế nào là static ? Phương thức, thuộc tính khai báo bằng từ khóa static được sử dụng khi nào ? Làm thế nào để truy cập được tới phương thức, thuộc tính static

- **static** là một từ khóa trong Java dùng để tạo ra các thành viên (thuộc tính và phương thức) **thuộc về chính lớp đó**, chứ không thuộc về một đối tượng (instance) cụ thể nào. Có nghĩa là thành viên static được **dùng chung** cho tất cả các đối tượng của lớp đó và chỉ có **một bản sao duy nhất** trong bộ nhớ, bất kể bạn tạo ra bao nhiêu đối tượng.

- **Sử dụng thuộc tính static khi:**

- *Định nghĩa hằng số (Constants):* Đây là trường hợp sử dụng phổ biến nhất. Các giá trị không đổi và dùng chung cho toàn bộ chương trình. Ví dụ: Hằng số PI trong lớp Math. Bất kỳ đâu bạn cần số PI, bạn chỉ cần dùng Math.PI.

```
public class Math {  
    public static final double PI = 3.14159;  
}
```

- *Lưu trữ dữ liệu dùng chung cho tất cả đối tượng:* Khi bạn cần một biến được chia sẻ và cập nhật bởi tất cả các đối tượng. Ví dụ:

```
class NguoiDung {  
    public static int soLuongNguoiDung = 0;  
    public NguoiDung() {  
        soLuongNguoiDung++;  
    }  
}
```

- **Sử dụng phương thức static khi:**

- *Tạo các phương thức tiện ích:* Các phương thức chỉ thực hiện một chức năng dựa trên tham số đầu vào mà không cần truy cập đến bất kỳ thuộc tính riêng nào của đối tượng.

- *Phương thức chỉ làm việc với các thành viên static khác:* Nếu một phương thức bên trong nó chỉ truy cập đến các thuộc tính và phương thức static khác của lớp, nó cũng nên được khai báo là static.

Phương thức static **không thể** truy cập trực tiếp các thuộc tính và phương thức **non-static** vì nó không biết phải làm việc trên đối tượng cụ thể nào.

#### - Truy cập thuộc tính/ phương thức static:

Truy cập thuộc tính và phương thức static một cách trực tiếp thông qua **tên của lớp** mà không cần tạo đối tượng.

**Cú pháp:** TenLop.tenThuocTinhStatic hoặc TenLop.tenPhuongThucStatic()

2. Thế nào là final ? Khai báo 1 biến final khác gì với static, biến khai báo bằng final có thể chỉnh sửa được không ? Nếu được cho ví dụ minh họa.

- Từ khóa **final** trong Java được sử dụng để tạo ra một thực thể **không thể thay đổi**. Tùy thuộc vào ngữ cảnh, nó có thể được áp dụng cho biến, phương thức, và lớp với ý nghĩa khác nhau.

- Đối với một **biến**, final có nghĩa là giá trị của nó **chỉ có thể được gán một lần duy nhất**. Sau lần gán đầu tiên, nó không thể được sửa đổi nữa, biến đó trở thành một hằng số.

| Tiêu chí | final                                              | Static                                                                    |
|----------|----------------------------------------------------|---------------------------------------------------------------------------|
| Mục đích | Để tạo ra một hằng số (không thể thay đổi giá trị) | Để tạo ra một thành viên thuộc về lớp dùng chung cho mọi đối tượng.       |
| Bản sao  | Mỗi đối tượng có một bản sao riêng của biến final  | Chỉ có một bản sao duy nhất dùng cho cả lớp, dùng chung cho mọi đối tượng |
| Giá trị  | Giá trị có thể khác nhau giữa các đối tượng        | Giá trị chung cho tất cả các đối tượng                                    |

|       |                                                                 |                                                             |
|-------|-----------------------------------------------------------------|-------------------------------------------------------------|
| Ví dụ | final int MSV ( mỗi sinh viên sẽ có một mã sinh viên không đổi) | static String tenKhoa ( Tất cả sinh viên học cùng một khoa) |
|-------|-----------------------------------------------------------------|-------------------------------------------------------------|

### - Biến final có thể chỉnh sửa được không?

**Không** đối với bản thân biến đó, nhưng có một lưu ý quan trọng khi biến đó là một tham chiếu đối tượng.

*Kiểu dữ liệu nguyên thủy (primitive types)*: Giá trị của biến final kiểu nguyên thủy (như int, double, boolean) hoàn toàn không thể thay đổi sau khi được gán.

*Kiểu dữ liệu tham chiếu (reference types)*: Khi một biến tham chiếu đối tượng (ví dụ: StringBuilder, ArrayList, hoặc một đối tượng của lớp bạn tự tạo) được khai báo là final, điều đó có nghĩa là:

- Không thể gán lại cho biến đó để nó trỏ đến một đối tượng khác.
- Có thể thay đổi nội dung (trạng thái) bên trong của chính đối tượng mà nó đang trỏ đến.

Ví dụ:

```
public static void main(String[] args) {

    final StringBuilder sb = new StringBuilder("Hello");
    System.out.println("Ban dau" + sb.toString());
    sb.append(" Viet Nam!");
    System.out.println("Sau thay doi: " + sb.toString());
    sb = new StringBuilder("Hello Viet Nam !"); // Lỗi: Cannot assign a value to
    final variable 'sb'
}
```