**STATIC**

**1. Static Variable (Biến tĩnh)**

* **Đặc điểm**: Biến static là biến dùng chung cho tất cả các đối tượng của lớp. Chỉ có một bản sao của biến static trong bộ nhớ.
* **Sử dụng**: Thường dùng để lưu trữ thông tin mà tất cả các đối tượng của lớp có chung (ví dụ: bộ đếm số lượng đối tượng).
* **Ví dụ**:

public class MyClass {

static int count = 0; // Biến tĩnh

public MyClass() {

count++;

}

public static void showCount() {

System.out.println("Count: " + count);

}

}

public class Main {

public static void main(String[] args) {

MyClass obj1 = new MyClass();

MyClass obj2 = new MyClass();

MyClass.showCount(); // Output: Count: 2

}

}

**2. Static Method (Phương thức tĩnh)**

* **Đặc điểm**: Phương thức static thuộc về lớp, không phụ thuộc vào đối tượng. Phương thức static chỉ có thể truy cập các biến và phương thức static khác, không thể truy cập biến hoặc phương thức không phải là static của lớp.
* **Sử dụng**: Thường dùng cho các phương thức tiện ích (utility methods) hoặc các phương thức không yêu cầu dữ liệu của đối tượng.
* **Ví dụ**:

public class MathUtils {

public static int add(int a, int b) {

return a + b;

}

}

public class Main {

public static void main(String[] args) {

int result = MathUtils.add(5, 3); // Gọi phương thức tĩnh mà không cần tạo đối tượng

System.out.println("Result: " + result); // Output: Result: 8

}

}

**3. Static Block (Khối mã tĩnh)**

* **Đặc điểm**: Khối mã tĩnh được chạy chỉ một lần khi lớp được nạp vào bộ nhớ, trước khi bất kỳ đối tượng nào được tạo ra. Khối tĩnh thường dùng để khởi tạo các biến static.
* **Sử dụng**: Dùng để thực hiện các thao tác khởi tạo phức tạp mà không cần đối tượng.
* **Ví dụ**:

public class MyClass {

static int number;

static {

number = 10; // Khối tĩnh để khởi tạo giá trị ban đầu

System.out.println("Static block executed");

}

public static void main(String[] args) {

System.out.println("Number: " + number); // Output: Number: 10

}

}

**4. Static Import (Nhập tĩnh)**

* **Đặc điểm**: static import cho phép nhập trực tiếp các thành phần static của một lớp khác mà không cần phải viết tên lớp mỗi khi sử dụng.
* **Ví dụ**:

import static java.lang.Math.\*; // Nhập tĩnh tất cả các phương thức tĩnh từ lớp Math

public class Main {

public static void main(String[] args) {

double result = sqrt(16); // Không cần viết Math.sqrt()

System.out.println("Result: " + result); // Output: Result: 4.0

}

}

**Tóm tắt**

* **static** được dùng khi một thành phần thuộc về lớp chứ không phải đối tượng.
* Biến và phương thức static có thể được truy cập mà không cần tạo đối tượng của lớp.
* Các biến static dùng chung giữa tất cả các đối tượng.
* Các phương thức static không thể truy cập biến hoặc phương thức không phải là static của lớp.