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?

1.1. Static là gì ?

* **Static** trong lập trình là một từ khóa (keyword) được sử dụng để khai báo các **biến** (thuộc tính), phương thức hoặc **khối mã** mà không phụ thuộc vào đối tượng (instance) của lớp.
* Các thành phần static được chia sẻ chung giữa **tất cả các đối tượng** của lớp. Mọi đối tượng đều dùng chung một bản sao của thuộc tính static, không có bản sao riêng biệt cho từng đối tượng.

1.2. Phương thức, thuộc tính khai báo bằng từ khóa static được sử dụng khi nào?

* Sử dụng thược tính static khi :
  + Dùng khi cần một giá trị hoặc biến chung cho tất cả các đối tượng. Ví dụ: đếm số lượng đối tượng được tạo ra từ lớp (có thể dùng một thuộc tính static để lưu số lượng này).
  + Khi là hằng số mà không bao giờ thay đổi .
* Sử dụng phương thức static khi :
  + Phương thức static thường được sử dụng cho các **tiện ích** hoặc các chức năng **không phụ thuộc vào trạng thái của đối tượng**. Ví dụ: phương thức tính toán như Math.max() trong Java.
  + Sử dụng khi phương thức không cần truy cập hoặc thay đổi dữ liệu của các đối tượng.

1.3. Làm thế nào để truy cập được tới phương thức, thuộc tính static?

* Vì thuộc tính static thuộc về lớp chứ không phải đối tượng nên có thể truy cập trực tiếp mà không cần tạo đối tượng :
* **Truy cập thuộc tính static:**
  + Vì thuộc tính static thuộc về lớp, nên có thể truy cập trực tiếp bằng tên lớp mà không cần tạo đối tượng.
  + Cú pháp: ClassName.staticVariable
* **Truy cập phương thức static:**
  + Tương tự như thuộc tính static, phương thức static cũng có thể được gọi trực tiếp bằng tên lớp.
  + Cú pháp: ClassName.staticMethod()

1. 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.

2.1. Thế nào là final ?

* Trong Java từ khoá final dùng để định nghĩa hằng số hoặc ngăn chặn sự thây đổi của biến , phương thức và cả class .
* **final dùng để khai báo một biến:** Biến này chỉ có thể được gán giá trị một lần. Sau khi đã được gán giá trị, biến không thể thay đổi giá trị của nó nữa.
* **final dùng để khai báo một phương thức:** Phương thức final không thể bị ghi đè (override) bởi các lớp con.
* **final dùng để khai báo một lớp:** Lớp final không thể bị kế thừa bởi các lớp khác. Ví dụ: Lớp String trong Java là một lớp final.

2.2. Khai báo 1 biến final khác gì với static?

* Biến final :
  + Khi khai báo một biến với từ khóa final, giá trị của biến đó không thể thay đổi sau khi được khởi tạo.
  + Biến final thường được sử dụng cho các **hằng số**, tức là các giá trị không bao giờ thay đổi trong suốt quá trình chương trình chạy.
  + Biến final không nhất thiết phải là thuộc tính của lớp mà có thể được khai báo trong **phương thức** hoặc **block** và chỉ được khởi tạo một lần.
* Biến static:
  + Thuộc tính static có thể thay đổi trong suốt chương trình, nhưng nó **thuộc về lớp** và **chia sẻ** giữa tất cả các đối tượng của lớp đó.
  + Biến static được khởi tạo khi chương trình bắt đầu và được giữ nguyên trong suốt quá trình chạy của chương trình.
* Đặc biệt trong một số trường hợp, chúng ta có thể kết hợp cả hai từ khóa static và final. Một biến static final là một hằng số toàn cục của lớp, được khởi tạo một lần và không bao giờ thay đổi, nhưng nó cũng thuộc về lớp và có thể truy cập mà không cần tạo đối tượng.

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

* Một biến final không thể thay đổi giá trị một khi nó đã được khởi tạo. Nếu cố gắng thay đổi giá trị của biến final sau khi nó đã được gán, trình biên dịch sẽ báo lỗi.