# Static & Final

# 1. Biến static

* **Chia sẻ giữa tất cả các đối tượng**: Biến static được chia sẻ giữa tất cả các đối tượng của lớp. Nếu một đối tượng thay đổi giá trị của biến static, tất cả các đối tượng khác sẽ thấy giá trị thay đổi đó.
* **Có thể truy cập mà không cần tạo đối tượng**: Bạn có thể truy cập biến static thông qua tên lớp mà không cần tạo một đối tượng.
* **Khởi tạo**: Biến static được khởi tạo khi lớp được nạp vào bộ nhớ. Chúng tồn tại trong suốt vòng đời của chương trình.

# 2. Phương thức Static

* **Truy cập mà không cần đối tượng**: Giống như biến static, bạn có thể gọi phương thức static mà không cần tạo một đối tượng của lớp.
* **Không thể truy cập thành viên không phải static**: Phương thức static không thể truy cập trực tiếp vào các thành viên không phải static (biến hoặc phương thức không phải static) của lớp.

# 3. Khối Static

* **Khối khởi tạo static**: Bạn có thể khai báo một khối static để khởi tạo các biến static. Khối này sẽ được thực thi khi lớp được nạp vào bộ nhớ.
* **Chỉ chạy một lần**: Khối static chỉ chạy một lần khi lớp được nạp vào bộ nhớ, giúp bạn thiết lập giá trị ban đầu cho các biến static.

# 4. Tính chất của static

* **Tăng hiệu suất**: Sử dụng biến và phương thức static có thể cải thiện hiệu suất vì chúng không cần tạo đối tượng để truy cập.
* **Tiết kiệm bộ nhớ**: Biến static tiết kiệm bộ nhớ vì chỉ có một bản sao của biến này cho tất cả các đối tượng.
* **Không thể bị ghi đè**: Nếu một phương thức static được khai báo trong một lớp cha, nó không thể bị ghi đè (overridden) bởi lớp con, mà chỉ có thể bị ẩn (hidden).

# 5. Biến final

* **Không thể thay đổi giá trị**: Khi một biến được khai báo là **final**, nó không thể được gán giá trị mới sau khi đã được khởi tạo.
* **Có thể khởi tạo một lần**: Biến final có thể được khởi tạo trong một khối khởi tạo (initializer) hoặc trong constructor của lớp.
* **Final với tham chiếu đối tượng**: Nếu biến final là một tham chiếu đối tượng, bạn không thể thay đổi tham chiếu đó, nhưng có thể thay đổi nội dung của đối tượng mà nó tham chiếu.

# 6. Phương thức final

* **Không thể ghi đè**: Nếu một phương thức được khai báo là **final**, các lớp con không thể ghi đè (override) phương thức đó.

# 7. Final Class (Lớp final)

* **Không thể kế thừa**: Nếu một lớp được khai báo là **final**, các lớp khác không thể kế thừa lớp đó.

# 8. Final với Tính chất

* **Bảo vệ tính toàn vẹn**: Sử dụng **final** giúp bảo vệ các thành phần của lớp khỏi bị thay đổi sau khi đã được khởi tạo, giúp tăng cường tính toàn vẹn và an toàn cho dữ liệu.
* **Cải thiện hiệu suất**: Biến, phương thức và lớp **final** có thể cải thiện hiệu suất, vì trình biên dịch có thể tối ưu hóa mã khi biết rằng chúng không thay đổi.