# Singleton Pattern

1. **Mục đích**

* Đảm bảo chỉ có một thể hiện (instance) của một class được tạo ra.
* Cung cấp một điểm truy cập toàn cục đến thể hiện (instance) đó, thường dùng cho các tài nguyên như kết nối cơ sở dữ liệu.

1. **Cách thức triển khai**

* Tạo 1 class với private constructor.
* Tạo một phương thức tĩnh (static method) để trả về instance duy nhất của class.
* Đảm bảo rằng instance duy nhất được tạo ra trong phương thức tĩnh (static method).

1. **Ưu điểm**

* Đảm bảo có duy nhất một instance của class.
* Giảm tải cho hệ thống khi quản lý tài nguyên.
* Cung cấp một điểm truy cập toàn cục.

# Factory Method Pattern

1. **Mục đích**

* Cung cấp một interface để tạo ra object, cho phép các class con quyết định class nào sẽ được khởi tạo.
* Thường được sử dụng khi không biết rõ trước loại object nào sẽ được tạo ra.

1. **Cách thức triển khai**

* Định nghĩa một interface hoặc class trừu tượng với phương thức tạo (factory method).
* Cài đặt class con sẽ triển khai phương thức tạo để tạo ra các object cụ thể.

1. **Ưu điểm**

* Tách biệt việc tạo object khỏi việc sử dụng chúng.
* Dễ dàng mở rộng cho các loại object mới mà không cần thay đổi mã hiện tại.