**MẪU THIẾT KẾ - CTK 44**

**LAB 1: TÌM HIỂU VÀ CÀI ĐẶT NHÓM MẪU CREATIONAL (CREATION PATTERNS)**

Mẫu Creational là nhóm các mẫu thiết kế (design patterns) tập trung vào việc tạo lập đối tượng theo cách tối ưu, linh hoạt, giảm phụ thuộc vào các lớp cụ thể. Nhóm này giúp cải thiện tính mở rộng và bảo trì của phần mềm. Trong Lab 1, chúng ta tìm hiểu và cài đặt 5 mẫu chính:

1. Abstract Factory (Nhà máy trừu tượng)

* Mục đích: Cung cấp một giao diện để tạo ra họ các đối tượng liên quan hoặc phụ thuộc nhau mà không cần chỉ rõ lớp cụ thể.
* Thành phần: AbstractFactory, ConcreteFactory, AbstractProduct, ConcreteProduct, Client.
* Ứng dụng: Trò chơi động vật ở châu Phi, châu Mỹ.

1. Builder (Người xây dựng)

* Mục đích: Tách biệt quá trình xây dựng đối tượng phức tạp khỏi biểu diễn của nó, cho phép cùng một quá trình xây dựng tạo ra các biểu diễn khác nhau.
* Thành phần: Builder, ConcreteBuilder, Director, Product.
* Ứng dụng: Tạo các đối tượng phức tạp theo từng bước (xe máy, ô tô, xe tay ga).

1. Factory Method (Phương thức nhà máy)

* Mục đích: Định nghĩa giao diện tạo đối tượng, nhưng để lớp con quyết định lớp nào sẽ được khởi tạo.
* Thành phần: Product, ConcreteProduct, Creator, ConcreteCreator.
* Ứng dụng: Tạo tài liệu như báo cáo, sơ yếu lý lịch.

1. Prototype (Nguyên mẫu)

* Mục đích: Dùng một đối tượng mẫu để tạo các đối tượng mới bằng cách sao chép.
* Thành phần: Prototype, Concrete Prototype, Client.
* Ứng dụng: Clone màu sắc, đối tượng phức tạp cần sao chép nhanh.

1. Singleton (Đơn nhất)

* Mục đích: Đảm bảo một lớp chỉ có duy nhất một thể hiện và cung cấp điểm truy cập toàn cục.
* Thành phần: Singleton.
* Ứng dụng: Load Balancer phân phối tải hệ thống.

Ý Nghĩa & Ứng Dụng Chung:

* Giảm sự phụ thuộc giữa các lớp, dễ bảo trì, mở rộng.
* Dễ dàng thay đổi cách khởi tạo đối tượng mà không ảnh hưởng mã nguồn chính.
* Tăng tính linh hoạt của hệ thống khi yêu cầu thay đổi hoặc thêm loại sản phẩm mới.