**Tìm hiểu về Factory Pattern**

1. Factory Pattern là gì?

Factory Pattern là một trong những Pattern phổ biến trong lập trình hướng đối tượng, chúng thuộc nhóm Creational Design Pattern (mẫu khởi tạo).

Mẫu này giúp quản lý và trả về các đối tượng theo yêu cầu, cung cấp một trong những các tốt nhất để tạo một đối tượng. Giúp cho việc khởi tạo đối tượng linh hoạt hơn. Trong Factory Pattern, chúng ta tạo đối tượng mà không để lộ logic ở phía người dùng và tham chiếu đến đối tượng mới bằng cách sử dụng interface chung.

2. Lợi ích của Factory Design Pattern

- Giúp che giấu xử lý logic của việc khởi tạo Objects.

- Cho phép lớp con chọn kiểu đối tượng cần tạo

- Thúc đẩy sự liên kết lỏng lẻo bằng cách loại bỏ sự cần thiết phải ràng buộc các lớp cụ thể vào code

- Factory Pattern giúp giảm sự phụ thuộc giữa các module: cung cấp 1 hướng tiếp cận với interface thay vì các implement. Giúp các chương trình độc lập với những lớp cụ thể mà chúng ta cần tạo đối tượng, code ở phía client sẽ không bị ảnh hưởng khi thay đổi logic ở facetory hay sub class.

- Việc mở rộng code dễ dàng hơn: khi cần mở rộng, chỉ việc tạp ra các subclass và implement thêm vào facetory method

- Dễ dàng quản lý life cycle của các Object

- Thống nhất về mặt naming convention

3. Ưu điểm và nhược điểm

a. Ưu điểm

• Factory Method Pattern giúp gom các đoạn code tạo ra product vào một nơi trong chương trình, nhờ đó giúp dễ theo dõi và thao tác.  
• Với Pattern này, chúng ta có thể dễ dàng mở rộng, thêm những đoạn code mới vào chương trình mà không cần phá vỡ các đối tượng ban đầu .

Nhược điểm  
• Mã code có thể trở nên phức tạp hơn mức bình thường do đòi hỏi phải sử dụng nhiều class mới có thể cài đặt được pattern này.

4. Mục đích sử dụng  
Factory pattern đưa ra 1 ý tưởng mới cho việc khởi tạo các instance phù hợp với mỗi yêu cầu từ phía Client. Sử dụng Factory pattern sẽ có những ưu điểm sau:

Tạo ra 1 cách mới trong việc khởi tạo Object thông qua 1 interface chung.

Khởi tạo các Objects mà che giấu đi xử lí logic của việc khởi tạo đấy.

Giảm sự phụ thuộc giữa các module, các logic với các class cụ thể. Mà chỉ phụ thuộc vào interface hoặc abstract class.