

PHẦN 1: SINGLETON PATTERN

Bài 1 – Singleton đơn giản (Easy)

Mô tả: Tạo lớp `AppConfig` lưu trữ cấu hình ứng dụng. Đảm bảo chỉ có một instance duy nhất.

Yêu cầu:

- Có private constructor
 - Sử dụng phương thức `getInstance()` để truy cập instance
-

Bài 2 – Thread-safe Singleton (Medium)

Mô tả: Tái thiết kế `Logger` singleton sao cho thread-safe, không tạo nhiều instance khi chạy đa luồng.

Yêu cầu:

- Sử dụng `synchronized`, `volatile`, hoặc `Bill Pugh Singleton`
 - Tạo hàm `log(String message)`
-

Bài 3 – Enum Singleton (Medium)

Mô tả: Viết lại lớp `DatabaseConnection` theo Singleton bằng `enum`.

Yêu cầu:

- Enum phải có method kết nối
 - Test thử gọi nhiều lần mà chỉ có 1 instance
-

PHẦN 2: FACTORY METHOD PATTERN

Bài 4 – Factory cơ bản (Easy)

Mô tả: Tạo interface `Shape` và các lớp `Circle`, `Square`. Dùng `ShapeFactory` để tạo đối tượng tương ứng.

Yêu cầu:

```
java
CopyEdit
Shape shape = ShapeFactory.getShape("circle");
```

Bài 5 – Factory với tham số đầu vào (Medium)

Mô tả: Viết `NotificationFactory` trả về `EmailNotification`, `SMSNotification`, `PushNotification`.

Yêu cầu:

- Interface: `Notification.send()`
 - Factory chọn đúng loại dựa trên tham số truyền vào
-

Bài 6 – Mở rộng Factory (Advanced)

Mô tả: Thêm khả năng đăng ký class động vào factory (registry).

Yêu cầu:

- `register(String type, Supplier<Product>)`
 - `create(String type)` tạo đối tượng đã đăng ký
-

PHẦN 3: ABSTRACT FACTORY PATTERN

Bài 7 – Abstract Factory cơ bản (Medium)

Mô tả: Tạo giao diện GUI cho 2 hệ điều hành: `WindowsFactory`, `MacFactory`.

Yêu cầu:

- Interface: `Button`, `Checkbox`
 - Abstract Factory: `GUIFactory`
 - `App` sử dụng `GUIFactory` mà không biết chi tiết cụ thể
-

Bài 8 – Thêm tính năng mở rộng (Advanced)

Mô tả: Mở rộng `GUIFactory` để thêm `Menu`, `Scrollbar`.

Yêu cầu:

- Đảm bảo mọi platform tạo ra được bộ giao diện đồng bộ (button, menu, scrollbar)