

Lab 06: Command

Áp dụng mẫu thiết kế Command

1. Giới thiệu bài toán

Bạn đang phát triển một ứng dụng quản lý điều khiển thiết bị điện tử trong một ngôi nhà thông minh. Ứng dụng này cho phép người dùng bật/tắt các thiết bị điện tử như đèn, quạt, và máy lạnh từ một giao diện điều khiển trung tâm. Để cải thiện tính linh hoạt và khả năng mở rộng, bạn quyết định áp dụng mẫu thiết kế Command để tách biệt các yêu cầu điều khiển (lệnh) khỏi các đối tượng thực hiện chúng (thiết bị điện tử).

2. Yêu cầu thiết kế và cài đặt

- Xây dựng lớp `Command`: Định nghĩa một interface hoặc abstract class `Command` với phương thức `execute()`.
- Tạo các lớp cụ thể cho các lệnh: Xây dựng các lớp cụ thể cho các lệnh điều khiển như `TurnOnLightCommand`, `TurnOffLightCommand`, `TurnOnFanCommand`, và `TurnOffFanCommand`, mỗi lớp sẽ triển khai giao diện `Command`.
- Xây dựng các lớp thiết bị: Tạo các lớp `Light`, `Fan`, mỗi lớp này nên có các phương thức như `turnOn()` và `turnOff()`.
- Tạo lớp `RemoteControl`: Xây dựng một lớp `RemoteControl` có thể lưu trữ và thực thi các lệnh. Lớp này nên có khả năng thêm lệnh vào danh sách và gọi phương thức `execute()` của lệnh.
- Giao diện người dùng: Tạo một lớp `Application` với giao diện người dùng đơn giản (console-based) cho phép người dùng gửi lệnh cho các thiết bị thông qua `RemoteControl`.

3. Gợi ý hướng dẫn giải quyết vấn đề

- Thiết kế Command: Xác định các lệnh và các hành động mà chúng sẽ thực hiện. Mỗi lệnh nên liên kết với một đối tượng thiết bị cụ thể và thực thi hành động trên đối tượng đó.
- Triển khai thiết bị: Mỗi thiết bị nên có các phương thức cụ thể cho hành động bật và tắt. Các lệnh sẽ gọi những phương thức này khi thực thi.
- Xây dựng RemoteControl: RemoteControl là lớp quản lý lệnh và thực thi lệnh. Bạn có thể sử dụng một danh sách để lưu trữ các lệnh và một phương thức để thực thi lệnh.
- Giao diện người dùng: Xây dựng một giao diện đơn giản để người dùng có thể chọn lệnh và thiết bị. Sử dụng `Scanner` trong Java để nhập lệnh từ bàn phím.

4. Kết quả cần đạt

- Các lớp lệnh cụ thể: Các lớp lệnh như `TurnOnLightCommand`, `TurnOffLightCommand`, `TurnOnFanCommand`, và `TurnOffFanCommand` phải hoạt động chính xác và thực hiện hành động trên các thiết bị tương ứng.

- Quản lý lệnh: Lớp `RemoteControl` phải cho phép lưu trữ và thực thi các lệnh. Nó phải có khả năng thêm lệnh vào danh sách và thực thi lệnh theo yêu cầu.
- Giao diện người dùng: Người dùng có thể gửi lệnh bật/tắt thiết bị qua giao diện điều khiển và thấy kết quả tương ứng (chẳng hạn như "Đèn đã bật" hoặc "Quạt đã tắt").

5. Hướng phát triển mở rộng bài toán

- Thêm tính năng undo/redo: Bạn có thể mở rộng ứng dụng để hỗ trợ tính năng undo/redo cho các lệnh đã thực thi.
- Thêm loại thiết bị khác: Mở rộng hệ thống để hỗ trợ thêm các thiết bị khác như máy điều hòa không khí hoặc hệ thống âm thanh.
- Lưu trữ lệnh: Triển khai khả năng lưu trữ các lệnh vào cơ sở dữ liệu hoặc file cấu hình để người dùng có thể xem lại lịch sử các lệnh đã thực hiện.

6. Hướng dẫn đánh giá

- Tính đúng đắn: Kiểm tra xem các lệnh có thực thi đúng hành động trên các thiết bị không.
- Tính linh hoạt: Đánh giá khả năng mở rộng của hệ thống khi thêm lệnh mới hoặc thiết bị mới.
- Giao diện người dùng: Đánh giá tính dễ sử dụng của giao diện người dùng và khả năng thực hiện các lệnh.
- Sự tuân thủ mẫu thiết kế: Kiểm tra xem mẫu thiết kế Command có được áp dụng đúng cách không, và mã nguồn có theo đúng cấu trúc của mẫu thiết kế hay không.