

Lab 04: Composite, Decorator

Áp dụng mẫu thiết kế Composite và Decorator

Bài tập này sẽ giúp người học hiểu rõ hơn về cách kết hợp hai mẫu thiết kế Composite và Decorator để giải quyết các bài toán thực tế trong phát triển phần mềm.

1. Giới thiệu bài toán

Trong một hệ thống quản lý đồ họa cho một phần mềm vẽ hình (Graphics Editor), người dùng có thể tạo ra các hình học cơ bản như hình tròn, hình vuông, đường thẳng. Người dùng cũng có thể nhóm các hình này lại với nhau để tạo thành các hình phức hợp. Ngoài ra, người dùng còn muốn áp dụng các hiệu ứng trang trí như viền, bóng đổ lên các hình đơn lẻ hoặc cả nhóm hình phức hợp.

Bạn được yêu cầu thiết kế và triển khai hệ thống này bằng cách áp dụng kết hợp mẫu thiết kế Composite để xử lý cấu trúc cây của các hình, và Decorator để áp dụng các hiệu ứng trang trí.

2. Yêu cầu thiết kế và cài đặt

Cấu trúc Composite:

- Tạo một lớp trừu tượng `Graphic` đại diện cho một thành phần trong cây (có thể là một hình đơn lẻ hoặc một nhóm hình).
- Cài đặt các lớp con `Circle`, `Square`, `Line` cho các hình cơ bản, kế thừa từ `Graphic`.
- Tạo lớp `CompositeGraphic` để đại diện cho một nhóm hình (cũng kế thừa từ `Graphic`), trong đó chứa danh sách các đối tượng `Graphic` và có khả năng thêm/xóa các thành phần.

Cấu trúc Decorator:

- Tạo một lớp trừu tượng `GraphicDecorator` kế thừa từ `Graphic` và chứa một đối tượng `Graphic`.
- Cài đặt các lớp con như `BorderDecorator`, `ShadowDecorator` để thêm hiệu ứng viền, bóng đổ.

Giao diện và chức năng chính:

- Hệ thống phải cho phép tạo ra các hình đơn lẻ và nhóm hình.
- Hệ thống phải cho phép thêm/xóa các hình vào/ra khỏi nhóm.
- Hệ thống phải cho phép áp dụng các hiệu ứng trang trí lên từng hình hoặc cả nhóm.

Cài đặt giao diện điều khiển đơn giản để thực hiện các thao tác trên.

3. Gợi ý hướng dẫn giải quyết vấn đề

- Phân tích yêu cầu: Xác định các đối tượng và hành vi cần thiết từ bài toán. Lưu ý cấu trúc cây tự nhiên của các hình và tính chất có thể trang trí thêm.
- Thiết kế lớp Composite: Xác định mối quan hệ giữa `Graphic` và các lớp con, đồng thời cách quản lý danh sách các đối tượng trong `CompositeGraphic`.
- Thiết kế lớp Decorator: Cần chú ý việc trang trí phải linh hoạt, cho phép áp dụng nhiều trang trí cùng lúc mà không làm thay đổi lớp đối tượng gốc.

- Triển khai: Bắt đầu từ các lớp cơ bản, sau đó xây dựng các lớp phức tạp hơn như `CompositeGraphic` và các `Decorator`.

4. Kết quả cần đạt

- Cài đặt thành công các lớp Composite và Decorator theo yêu cầu bài toán.
- Chương trình có thể tạo và thao tác với các đối tượng hình đơn lẻ và nhóm hình.
- Chương trình có thể áp dụng các hiệu ứng trang trí lên đối tượng một cách linh hoạt.
- Mã nguồn rõ ràng, dễ bảo trì và mở rộng.

5. Hướng phát triển mở rộng bài toán

- Hỗ trợ nhiều loại hình khác: Thêm các loại hình học khác như tam giác, đa giác.
- Thêm các hiệu ứng trang trí: Triển khai thêm các hiệu ứng như làm mờ (blur), đổi màu (color change).
- Tích hợp giao diện đồ họa người dùng (GUI): Thay vì giao diện điều khiển, có thể phát triển ứng dụng với giao diện đồ họa, cho phép người dùng vẽ và trang trí trực tiếp trên canvas.

6. Hướng dẫn đánh giá

- Tính đầy đủ: Kiểm tra xem người học đã triển khai đầy đủ các yêu cầu cơ bản của bài toán chưa.
- Tính đúng đắn: Kiểm tra tính chính xác của cấu trúc Composite và Decorator khi áp dụng trong hệ thống.
- Tính linh hoạt: Đánh giá khả năng mở rộng và tái sử dụng mã nguồn, đặc biệt là với các Decorator.
- Chất lượng mã nguồn: Đánh giá theo tiêu chuẩn lập trình sạch (clean code), bao gồm việc đặt tên biến, tách chức năng, và bình luận mã.
- Thử nghiệm: người học cần chuẩn bị các trường hợp thử nghiệm để kiểm chứng các tính năng chính của hệ thống.