

Lab 03: Adapter, Bridge

Áp dụng mẫu thiết kế Adapter và Bridge trong phát triển phần mềm

Bài tập này giúp người học thực hành việc kết hợp nhiều mẫu thiết kế trong một tình huống phát triển phần mềm thực tế, qua đó nâng cao khả năng áp dụng lý thuyết vào thực tiễn.

1. Giới thiệu bài toán

Bạn được giao nhiệm vụ phát triển một hệ thống quản lý tài liệu trực tuyến cho một công ty lớn. Hệ thống này cho phép người dùng tải lên và truy cập các tài liệu ở nhiều định dạng khác nhau như PDF, DOCX, và TXT. Tuy nhiên, các tài liệu này có thể được lưu trữ trên các dịch vụ lưu trữ đám mây khác nhau như Google Drive, Dropbox, hoặc trên hệ thống tệp cục bộ.

Để đảm bảo tính linh hoạt và khả năng mở rộng của hệ thống, bạn cần áp dụng các mẫu thiết kế phần mềm để giải quyết bài toán này. Đặc biệt, bạn cần sử dụng kết hợp các mẫu thiết kế Adapter và Bridge.

2. Yêu cầu thiết kế và cài đặt

Yêu cầu chính:

- Thiết kế một giao diện chung ('Document') cho tất cả các loại tài liệu (PDF, DOCX, TXT).
- Sử dụng mẫu thiết kế Adapter để chuyển đổi giữa các định dạng tài liệu khác nhau.
- Sử dụng mẫu thiết kế Bridge để tách biệt việc truy cập và lưu trữ tài liệu khỏi logic xử lý cụ thể, cho phép hệ thống hỗ trợ nhiều loại dịch vụ lưu trữ đám mây khác nhau.
- Cài đặt các lớp cụ thể để xử lý các định dạng tài liệu và kết nối với các dịch vụ lưu trữ đám mây.

Chi tiết:

- Xây dựng các lớp 'PDFDocument', 'DOCXDocument', và 'TXTDocument' triển khai từ giao diện 'Document'.
- Sử dụng mẫu Adapter để tạo các lớp chuyển đổi giữa các định dạng tài liệu, ví dụ 'PDFToDOCXAdapter', 'DOCXToTXTAdapter', v.v.
- Áp dụng mẫu Bridge để tạo các lớp 'CloudStorage' (chứa một phương thức trừu tượng để lưu trữ và truy xuất tài liệu) và các lớp con như 'GoogleDriveStorage', 'DropboxStorage', và 'LocalStorage' để xử lý lưu trữ trên các nền tảng khác nhau.
- Xây dựng lớp quản lý ('DocumentManager') để điều phối việc lưu trữ và chuyển đổi tài liệu.

3. Gợi ý hướng dẫn giải quyết vấn đề

Adapter Pattern:

- Xác định các lớp hiện tại không tương thích với hệ thống mới.
- Tạo các lớp adapter để làm cầu nối giữa các định dạng tài liệu.
- Đảm bảo adapter có thể chuyển đổi định dạng một cách trơn tru và dễ dàng mở rộng.

Bridge Pattern:

- Phân tách giao diện lưu trữ khỏi các thực thể cụ thể.

- Áp dụng lớp cầu nối để thay thế trực tiếp các loại lưu trữ mà không làm ảnh hưởng đến logic xử lý tài liệu.
- Xác định các phần mà bạn có thể thay đổi độc lập (ví dụ: loại tài liệu và nơi lưu trữ).

4. Kết quả cần đạt

- Mã nguồn cài đặt mẫu thiết kế Adapter và Bridge đáp ứng các yêu cầu đã nêu.
- Một ứng dụng Java có thể quản lý và xử lý các tài liệu ở nhiều định dạng, hỗ trợ lưu trữ và truy xuất trên các nền tảng đám mây khác nhau.
- Hệ thống dễ dàng mở rộng để hỗ trợ thêm các định dạng tài liệu và dịch vụ lưu trữ khác.

5. Hướng phát triển mở rộng bài toán

- Mở rộng hỗ trợ định dạng tài liệu: Thêm các loại tài liệu khác như `ODT`, `RTF`, v.v.
- Thêm dịch vụ lưu trữ: Hỗ trợ thêm các dịch vụ lưu trữ đám mây khác như OneDrive, Amazon S3.
- Tối ưu hóa hiệu suất: Áp dụng các kỹ thuật caching hoặc lazy loading để cải thiện hiệu suất hệ thống.
- Bảo mật: Tích hợp cơ chế mã hóa cho các tài liệu trước khi lưu trữ trên đám mây.

6. Hướng dẫn đánh giá

- Đúng yêu cầu: Đánh giá dựa trên việc đáp ứng các yêu cầu thiết kế và cài đặt.
- Tính sáng tạo: Khuyến khích người học mở rộng bài toán và đề xuất các giải pháp sáng tạo.
- Chất lượng mã nguồn: Xem xét cấu trúc mã, khả năng tái sử dụng, và mức độ tuân thủ nguyên tắc SOLID.
- Hiệu năng: Đánh giá hiệu suất của hệ thống khi xử lý một số lượng lớn tài liệu và kết nối với nhiều dịch vụ lưu trữ.