

Lab 01: Singleton

Áp dụng mẫu thiết kế Singleton trong phát triển phần mềm

Bài tập này không chỉ giúp người học làm quen với mẫu thiết kế Singleton mà còn rèn luyện khả năng giải quyết các vấn đề thực tế trong phát triển phần mềm.

1. Giới thiệu bài toán

Bạn đang phát triển một ứng dụng quản lý cấu hình hệ thống cho một công ty lớn. Ứng dụng này sẽ quản lý các thiết lập cấu hình như kết nối cơ sở dữ liệu, cài đặt mạng, và thông tin đăng nhập. Các thiết lập này cần phải được truy xuất và cập nhật một cách nhất quán trong suốt quá trình chạy của ứng dụng. Bạn cần đảm bảo rằng chỉ có một đối tượng duy nhất quản lý các thiết lập này trong toàn bộ ứng dụng, bất kể có bao nhiêu lớp hay luồng truy cập vào.

2. Yêu cầu thiết kế và cài đặt

Thiết kế: Sử dụng mẫu thiết kế Singleton để đảm bảo chỉ có một đối tượng duy nhất quản lý cấu hình hệ thống. Đối tượng này phải có khả năng:

- Lưu trữ các thiết lập cấu hình.
- Cung cấp phương thức để lấy giá trị cấu hình theo tên.
- Cung cấp phương thức để cập nhật giá trị cấu hình.
- Cung cấp phương thức để kiểm tra tình trạng của cấu hình (ví dụ: kết nối cơ sở dữ liệu còn hoạt động hay không).

Cài đặt:

- Tạo một lớp `ConfigurationManager` sử dụng mẫu thiết kế Singleton.
- Đảm bảo rằng lớp này không thể bị khởi tạo trực tiếp từ bên ngoài.
- Cung cấp một phương thức tĩnh để lấy đối tượng duy nhất của lớp này.
- Cung cấp các phương thức cần thiết để quản lý cấu hình (như đã nêu ở phần yêu cầu thiết kế).

3. Gợi ý hướng dẫn giải quyết vấn đề

- Xác định rõ vai trò của lớp `ConfigurationManager` trong ứng dụng.
- Sử dụng từ khóa `private` để ẩn constructor của lớp, ngăn không cho khởi tạo đối tượng từ bên ngoài.
- Tạo một biến tĩnh trong lớp để lưu trữ đối tượng Singleton.
- Cung cấp một phương thức tĩnh để trả về đối tượng Singleton này.
- Đảm bảo rằng mọi phương thức trong lớp `ConfigurationManager` đều hoạt động chính xác khi chỉ có một đối tượng duy nhất quản lý cấu hình.

4. Kết quả cần đạt

- Người học cần cài đặt thành công một lớp sử dụng mẫu thiết kế Singleton.

- Đảm bảo rằng lớp này hoạt động chính xác trong môi trường đa luồng (có thể cần sử dụng `synchronized` hoặc các cơ chế khác để đảm bảo tính toàn vẹn dữ liệu).
- Người học phải viết các phương thức để thêm, cập nhật, và lấy các giá trị cấu hình một cách chính xác.

5. Hướng phát triển mở rộng bài toán

- Mở rộng bài tập bằng cách thêm khả năng tải cấu hình từ tệp tin bên ngoài hoặc cơ sở dữ liệu.
- Thử nghiệm với việc sử dụng mẫu thiết kế Singleton trong môi trường đa luồng và xử lý các tình huống tranh chấp tài nguyên.
- Tích hợp mẫu thiết kế Singleton với các mẫu thiết kế khác như Factory Method hoặc Dependency Injection để tăng tính linh hoạt của ứng dụng.

6. Hướng dẫn đánh giá

- Thiết kế: Đánh giá dựa trên việc áp dụng đúng mẫu thiết kế Singleton (cấu trúc lớp, phương thức truy cập đối tượng duy nhất).
- Cài đặt: Đánh giá mức độ chính xác và đầy đủ của việc cài đặt lớp `ConfigurationManager`.
- Hiệu năng: Kiểm tra xem lớp Singleton có xử lý đúng trong môi trường đa luồng và quản lý tài nguyên hiệu quả hay không.
- Mở rộng: Đánh giá cách người học mở rộng và tích hợp Singleton với các tính năng khác.