# Đánh giá vi phạm SOLID và đề xuất cải thiện

|  |  |  |  |
| --- | --- | --- | --- |
| Modules | Nguyên lý vi phạm | Lý do vi phạm | Đề xuất cải thiện |
| Cart | SRP | Lớp chịu nhiều trách nhiệm, bao gồm quản lý giỏ hàng, tính toán tổng số lượng, kiểm tra tính khả dụng, và xử lý ngoại lệ. | Tách trách nhiệm thành các lớp riêng như CartCalculator và ProductAvailabilityChecker. |
| Cart | OCP | Lớp không dễ mở rộng vì logic kiểm tra tính khả dụng được viết cứng trong phương thức checkAvailabilityOfProduct. | Sử dụng interface hoặc lớp trừu tượng để quản lý các quy tắc kiểm tra sản phẩm. |
| Cart | DIP | Lớp phụ thuộc trực tiếp vào Media và CartMedia, làm tăng sự phụ thuộc cứng vào các lớp chi tiết. | Sử dụng abstraction hoặc dependency injection để giảm phụ thuộc. |
| CartMedia | SRP | Lớp chịu nhiều trách nhiệm: vừa lưu trữ thông tin về media, số lượng, giá cả, vừa xử lý logic khởi tạo đối tượng. | Tách lớp thành các lớp chuyên biệt như MediaInfo và CartMediaManager để quản lý logic liên quan. |
| CartMedia | OCP | Khó mở rộng để thêm các thuộc tính mới hoặc logic xử lý khác mà không sửa đổi trực tiếp lớp. | Sử dụng các interface hoặc kế thừa từ lớp trừu tượng để hỗ trợ mở rộng. |
| CartMedia | DIP | Lớp phụ thuộc trực tiếp vào Media, làm tăng độ kết dính và giảm khả năng thay đổi hoặc thay thế. | Sử dụng dependency injection hoặc interface để quản lý Media. |
| Book | SRP | Lớp Book thực hiện các chức năng không liên quan đến nhau: lưu trữ dữ liệu sách và thao tác database (phương thức getMediaById. | Tách các logic làm việc với database sang một lớp DAO (Data Access Object), như lớp BookDAO. |
| Book | OCP | Lớp Book bị phụ thuộc chặt chẽ vào SQL query trong phương thức getMediaById, khi thay đổi dữ liệu hoặc query thì cần sửa lớp. | Sử dụng lớp DAO để quản lý truy vấn SQL, tách biệt logic database khỏi lớp Book. |
| Book | LSP | Việc kế thừa phương thức getMediaById từ lớp cha Media được triển khai khác biệt trong lớp con Book, gây hiểu nhầm cho người dùng. | Xem xét loại bỏ hoặc tách interface riêng cho các thao tác truy vấn database. |
| **CD** | SRP | Thực hiện logic truy vấn dữ liệu từ database thay vì thuộc về DAO. | Tách logic truy vấn SQL ra lớp DAO (Đề xuất: CDDAO) để quản lý toàn bộ truy vấn và thao tác dữ liệu. |
| DVD | SRP | Phương thức getMediaById và getAllMedia thực hiện truy vấn cơ sở dữ liệu, không phù hợp với trách nhiệm của lớp DVD, vốn chỉ đại diện cho một thực thể dữ liệu. | Tách logic truy vấn cơ sở dữ liệu ra một lớp DAO riêng như DVDDAO. Lớp này sẽ chịu trách nhiệm quản lý dữ liệu liên quan đến DVD. |
| Media | SRP | Các phương thức chứa truy vấn cơ sở dữ liệu làm cho lớp chịu trách nhiệm quá nhiều việc | Tách chức năng xử lý cơ sở dữ liệu ra một lớp DAO riêng |
| Order | SRP | Lớp Order chịu trách nhiệm quá nhiều: quản lý thông tin đơn hàng, tính toán phí và số tiền thanh toán. | Tách chức năng tính toán (getAmount()) ra một lớp dịch vụ riêng như OrderService. |
| Order | SRP | Danh sách lstOrderMedia được quản lý trực tiếp trong lớp Order, dẫn đến trách nhiệm trùng lặp khi thêm/xóa OrderMedia. | Sử dụng lớp trung gian quản lý danh sách OrderMedia, ví dụ OrderMediaManager. |
| Order | ISP | HashMap không phải là cách tốt nhất để lưu trữ thông tin giao hàng, vì không có ràng buộc chặt chẽ về cấu trúc dữ liệu. | Tạo một lớp riêng, chẳng hạn DeliveryInfo, để lưu trữ thông tin giao hàng với các thuộc tính cụ thể ( tên, địa chỉ, số điện thoại) |
| Order | SRP | shippingFees có thể được tính toán dựa trên thông tin giao hàng hoặc cấu hình, không cần lưu trữ trực tiếp trong lớp. | Loại bỏ thuộc tính này và sử dụng một phương thức trong OrderService để tính toán shippingFees khi cần thiết. |
| BaseController | SRP | Lớp BaseController không chỉ chịu trách nhiệm điều hướng (controller logic), mà còn trực tiếp xử lý logic liên quan đến Cart. | Tách các phương thức thao tác Cart ra một service chuyên biệt (ví dụ: CartService) để lớp BaseController chỉ tập trung vào điều hướng và giao tiếp giữa các thành phần. |
| BaseController | OCP | Lớp phụ thuộc trực tiếp vào Cart, khiến việc mở rộng hoặc thay đổi logic của Cart yêu cầu sửa đổi trong BaseController. | Sử dụng một lớp trung gian như CartService với interface để trừu tượng hóa các thao tác liên quan đến Cart, giúp lớp BaseController không bị ràng buộc chặt chẽ. |
| BaseController | DIP | Lớp phụ thuộc trực tiếp vào một implementation cụ thể (Cart) thay vì một abstraction (interface hoặc lớp trừu tượng). | Tạo một interface (ví dụ: ICartService) và sử dụng lớp CartService để cung cấp các chức năng thao tác Cart. BaseController sẽ phụ thuộc vào interface này thay vì Cart. |
| HomeController | SRP | Lớp HomeController chịu trách nhiệm điều khiển luồng sự kiện trong màn hình chính, nhưng cũng trực tiếp gọi đến phương thức lấy dữ liệu Media từ cơ sở dữ liệu. | Tách logic truy vấn dữ liệu Media ra một lớp riêng (ví dụ: MediaService), giúp HomeController chỉ tập trung vào điều khiển luồng dữ liệu và sự kiện. |
| HomeController | OCP | Lớp phụ thuộc trực tiếp vào lớp Media để lấy dữ liệu, do đó nếu cần thay đổi cách truy vấn dữ liệu (ví dụ: đổi từ SQL sang NoSQL), lớp này sẽ cần sửa đổi. | Tạo một lớp service (như MediaService) để xử lý việc truy vấn, và HomeController chỉ gọi qua service này, giúp mở rộng dễ dàng mà không phải sửa đổi lớp controller. |
| HomeController | DIP | Lớp này phụ thuộc trực tiếp vào implementation cụ thể của Media thay vì một abstraction. Điều này dẫn đến sự phụ thuộc chặt chẽ vào cách thực thi hiện tại của lớp Media. | Tạo một interface (ví dụ: IMediaService) để trừu tượng hóa việc truy vấn dữ liệu Media, giúp giảm sự phụ thuộc của HomeController vào lớp cụ thể. |
| PaymentController | SRP | Lớp PaymentController vừa xử lý luồng thanh toán, vừa chịu trách nhiệm làm trống giỏ hàng bằng cách gọi trực tiếp Cart.emptyCart(). Điều này vi phạm SRP vì lớp đảm nhận quá nhiều trách nhiệm. | Tách logic liên quan đến giỏ hàng (xử lý việc làm trống giỏ hàng) vào một service riêng như CartService để tăng tính tập trung của PaymentController. |
| PaymentController | OCP | Vi phạm nhẹ. Nếu cần thay đổi cách làm trống giỏ hàng (ví dụ: chuyển từ xử lý cục bộ sang xử lý trên server), cần sửa đổi trực tiếp trong PaymentController. | Sử dụng abstraction (interface) để tách biệt logic xử lý giỏ hàng. PaymentController sẽ chỉ gọi đến các phương thức trong interface của CartService. |
| PaymentController | DIP | Lớp này phụ thuộc trực tiếp vào lớp Cart (implementation cụ thể) để làm trống giỏ hàng, dẫn đến phụ thuộc chặt chẽ. | Thay vì sử dụng trực tiếp Cart, hãy sử dụng một interface (ví dụ: ICartService) để quản lý các thao tác liên quan đến giỏ hàng. |
| PaymentController | ISP | Không vi phạm rõ rệt, nhưng TransactionResultListener có thể gây khó khăn trong việc tái sử dụng nếu giao diện này mở rộng thêm nhiều chức năng không liên quan đến xử lý kết quả giao dịch. | Đảm bảo giao diện TransactionResultListener chỉ chứa các phương thức liên quan đến xử lý giao dịch. Nếu cần thêm chức năng khác, hãy tạo một giao diện riêng. |
| PlaceOrderController | SRP | Lớp này chịu trách nhiệm xử lý luồng sự kiện đặt hàng, kiểm tra tính khả dụng, tạo hóa đơn, tính phí vận chuyển và validate thông tin giao hàng, dẫn đến đảm nhận quá nhiều trách nhiệm. | Tách các trách nhiệm này thành các lớp hoặc service riêng, chẳng hạn: ValidationService (validate thông tin), ShippingService (tính phí giao hàng), và OrderService (xử lý đặt hàng). |
| PlaceOrderController | OCP | Không vi phạm rõ ràng. Nhưng các phương thức như validatePhoneNumber, validateName và calculateShippingFee khó mở rộng vì chúng được triển khai trực tiếp trong controller. | Sử dụng chiến lược hoặc dịch vụ bên ngoài để hỗ trợ việc validate và tính phí, giúp mở rộng hoặc thay đổi logic dễ dàng hơn mà không cần chỉnh sửa controller. |
| PlaceOrderController | DIP | Lớp này phụ thuộc trực tiếp vào Cart và các chi tiết implementation của nó. Điều này làm giảm khả năng thay đổi hoặc thay thế Cart. | Tạo interface trừu tượng (ví dụ: ICartService) để quản lý các thao tác trên giỏ hàng. Controller sẽ phụ thuộc vào abstraction này thay vì chi tiết cụ thể của lớp Cart. |
| ViewCartController | SRP | Không vi phạm rõ ràng. Lớp này tập trung vào việc xử lý luồng sự kiện liên quan đến giỏ hàng, như kiểm tra tính khả dụng và tính tổng giá trị. | Tiếp tục giữ trách nhiệm tập trung vào luồng điều khiển giỏ hàng. Có thể tách các thao tác xử lý logic của giỏ hàng vào một service riêng nếu cần. |
| ViewCartController | OCP | Có thể vi phạm nhẹ. Hiện tại phụ thuộc vào Cart trực tiếp và không có cách mở rộng hoặc thay thế hành vi mà không chỉnh sửa controller. | Sử dụng abstraction như một interface (ICartService) để thay thế việc gọi trực tiếp Cart, giúp mở rộng dễ dàng hơn. |
| ViewCartController | DIP | Vi phạm nhẹ. Lớp này phụ thuộc trực tiếp vào Cart, làm giảm tính linh hoạt khi muốn thay đổi logic xử lý giỏ hàng. | Đưa ra một abstraction (CartService) làm trung gian giữa controller và Cart để giảm sự phụ thuộc cụ thể. |