

Đánh giá Cohesion, Coupling và SOLID của CartController

Cohesion

Phương thức/Lớp	Cohesion	Giải thích	Khuyến nghị
Constructor (CartController)	Functional Cohesion	Khởi tạo giỏ hàng cart và danh sách listCartMedia từ giỏ hàng chung.	Tách trạng thái tĩnh (static) để tăng khả năng hỗ trợ đa giỏ hàng.
requestToPlaceOrder	Functional Cohesion	Lọc danh sách sản phẩm được đánh dấu (checked) từ danh sách CartMedia.	Duy trì trạng thái hiện tại.
getAmount	Functional Cohesion	Tính tổng tiền của các sản phẩm được chọn dựa trên số lượng và giá.	Cân nhắc đưa phương thức này vào lớp Cart để giảm trách nhiệm của CartController.
checkAvailabilityOfSpecificProduct	Functional Cohesion	Kiểm tra tính khả dụng của sản phẩm trong kho so với yêu cầu.	Cân nhắc chuyển logic này sang CartMedia hoặc một class riêng biệt.
addMediaToCart	Logical Cohesion	Kiểm tra các điều kiện liên quan (sản phẩm đã có trong giỏ, số lượng khả dụng) rồi thêm sản phẩm vào danh sách.	Chuyển một phần logic kiểm tra này sang lớp Cart hoặc sử dụng các phương thức riêng trong CartMedia.
updateQuantityOfMedia	Functional Cohesion	Cập nhật số lượng của một sản phẩm đã có trong giỏ.	Duy trì trạng thái hiện tại.
updateMediaCheck	Logical Cohesion	Đổi trạng thái "checked" của sản phẩm trong danh sách dựa trên trạng thái hiện tại.	Sử dụng các phương thức phụ trợ trong CartMedia để tăng tính rõ ràng.

Coupling

Lớp	Coupling với	Loại Coupling	Giải thích	Khuyến nghị
CartController	Cart	Data Coupling	Phụ thuộc trực tiếp vào Cart để lấy danh sách CartMedia và quản lý giỏ hàng.	Sử dụng Dependency Injection để giảm phụ thuộc trực tiếp.
CartController	CartMedia	Data Coupling	Gọi phương thức và thuộc tính của CartMedia (như isChecked, getQuantity, getPrice).	Duy trì trạng thái hiện tại, nhưng đảm bảo CartMedia được kiểm soát rõ ràng hơn thông qua các interface cụ thể.
CartMedia	Media	Data Coupling	CartMedia chứa thuộc tính Media để truy cập thông tin sản phẩm trong giỏ.	Duy trì trạng thái hiện tại.

SOLID

Nguyên tắc	Tuân thủ/Vi phạm	Giải thích	Khuyến nghị
SRP	Vi phạm	CartController quản lý cả trạng thái giỏ hàng, kiểm tra sản phẩm, xử lý logic đặt hàng.	Phân chia thành các lớp riêng biệt, ví dụ: CartManager (quản lý giỏ hàng), CartValidator (xác minh sản phẩm).
OCP	Vi phạm	Nếu thêm loại giỏ hàng hoặc logic xử lý mới, cần thay đổi code của CartController.	Sử dụng Strategy Pattern cho các logic xử lý, giúp mở rộng mà không cần sửa đổi lớp.
LSP	Tuân thủ	CartMedia hoặc các lớp con tiềm năng có thể thay thế nhau mà không phá vỡ logic trong hệ thống.	Duy trì trạng thái hiện tại.

Nguyên tắc	Tuân thủ/Vi phạm	Giải thích	Khuyến nghị
ISP	Không áp dụng	Không có interface hoặc phân chia thành phần cụ thể nào áp dụng trong thiết kế hiện tại.	Đề xuất trừu tượng hóa hành vi thông qua interface, ví dụ: CartService cho các hành động chung.
DIP	Vi phạm	CartController phụ thuộc trực tiếp vào Cart và CartMedia.	Sử dụng Dependency Injection để giảm phụ thuộc, ví dụ inject một CartService thay vì trực tiếp truy cập Cart.