

Phân tích vấn đề của mã nguồn và thiết kế hiện tại khi triển khai Additional Requirements

1. Thêm mặt hàng Media mới:

AudioBook

Thông tin về loại mặt hàng mới như sau: AudioBook

+ String author, ví dụ: Paulo Coelho

+ String format, ví dụ: mp3

+ String language, ví dụ: English

+ String accent, ví dụ: Male - North America

+ int lengthInMinutes, ví dụ: 226

⇒ Việc thêm mặt hàng mới (AudioBook) không vi phạm các nguyên tắc thiết kế. Khi triển khai thêm loại sản phẩm AudioBook, mã nguồn cần thêm class AudioBook kế thừa từ class Media tương tự như Book, CD, DVD.

2. Thay đổi cách tính phí vận chuyển:

Phí vận chuyển sẽ được tính dựa theo khối lượng thực tế và độ cồng kềnh của kiện hàng.

Độ cồng kềnh của kiện hàng được tính theo kích thước của kiện hàng: chiều dài, chiều rộng, chiều cao, với công thức như sau:

Độ cồng kềnh (kg) = Chiều dài (cm) x Chiều rộng (cm) x Chiều cao (cm) / 6000. Khối lượng để tính phí vận chuyển = MAX (khối lượng thực tế, độ cồng kềnh).

VD: Kiện hàng nặng 2kg, có độ cồng kềnh $50 \times 60 \times 10 / 6000 = 5$ (kg). Vậy khối lượng để tính phí vận chuyển = MAX (2,5) = 5kg. Sau đó, phí vận chuyển được tính như trong mô tả của AIMS.

- ⇒ Khi thay đổi cách tính phí, class PlaceOrderController sẽ phải thay đổi do phương thức calculateShippingFee thuộc class phải chỉnh sửa. Để tránh PlaceOrderController đảm nhận quá nhiều trách nhiệm (vi phạm Single Responsibility principle), ta nên tách phương thức calculateShippingFee thành một class riêng phụ trách việc tính phí vận chuyển (ví dụ: ShippingFeeCalculator).
- ⇒ Ngoài ra, cần tách logic tính khối lượng đơn hàng thành phương thức riêng, khi muốn thay đổi cách tính phí vận chuyển thì chỉ cần thay đổi phương thức tính khối lượng đơn hàng mà các thành phần khác không bị ảnh hưởng.