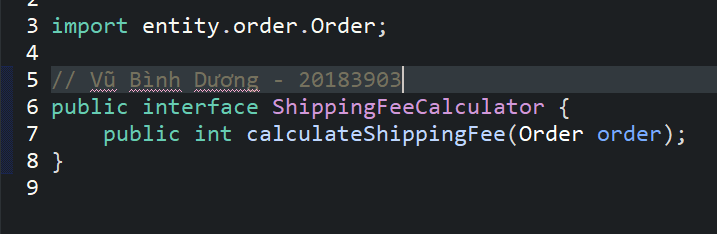
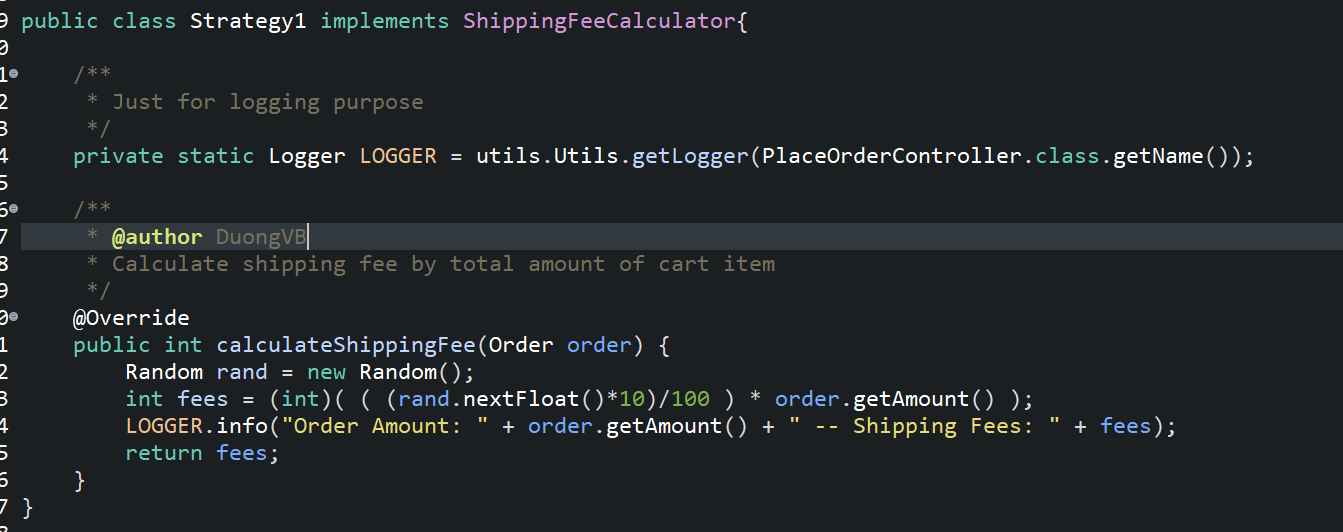
## **I. Yêu cầu mở rộng**

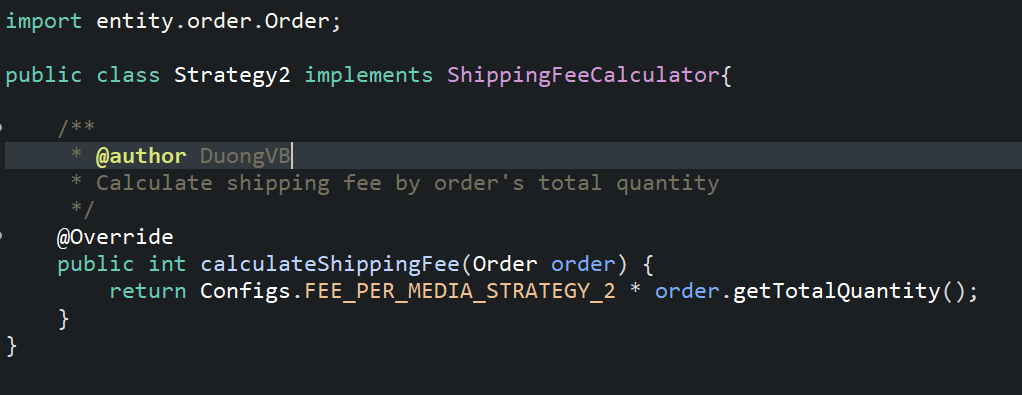
***1. Thay đổi phí ship***

Tạo interface calculateShippingFee và phương thức abstract calculateShippingFee:

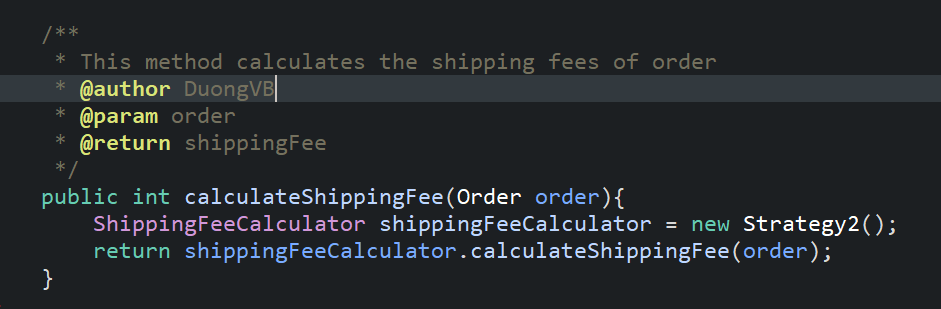


Tạo 2 class Strategy1 và Strategy2 kế thừa interface trên để tạo 2 cách xử lí shipping fee:



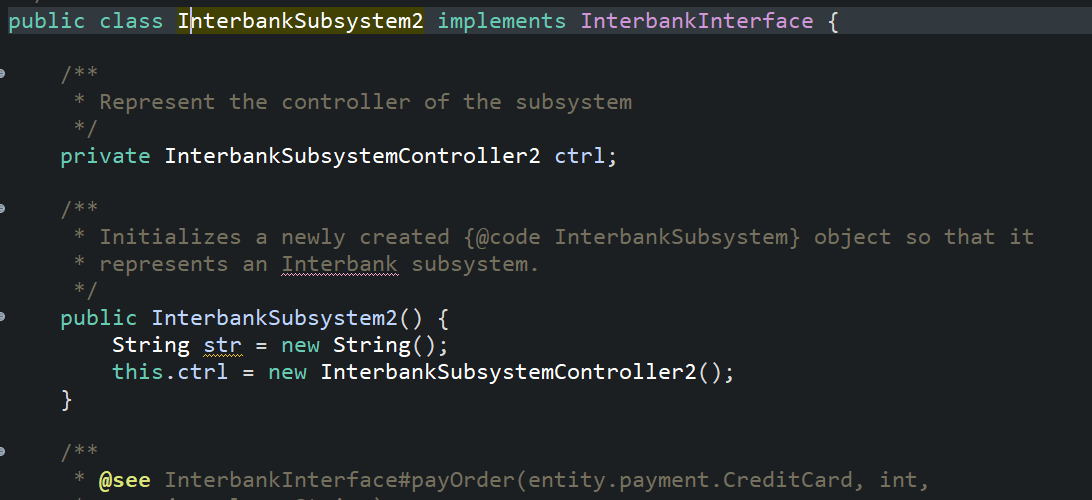


Define và sử dụng trong controller PlaceOrderController:

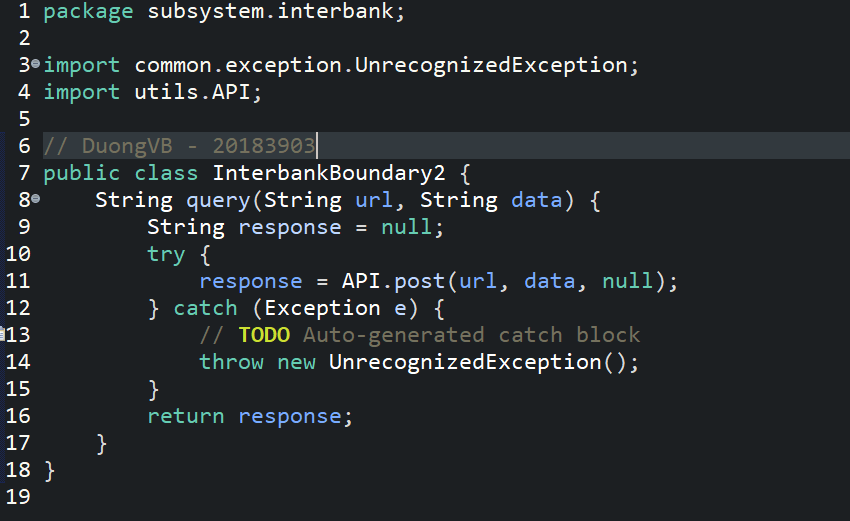


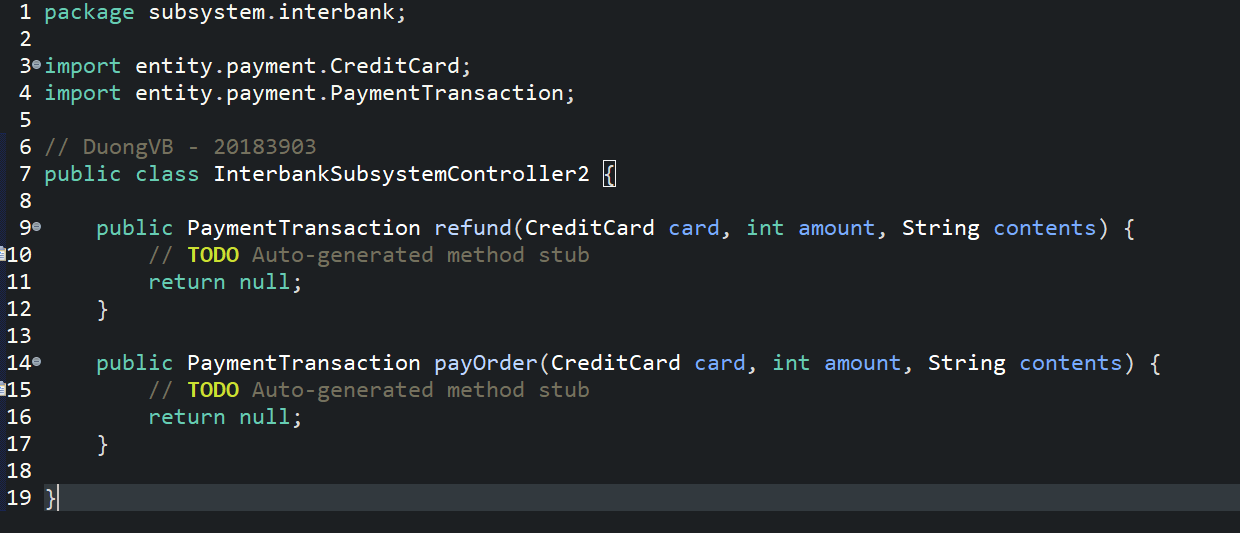
***2. Tạo interbank khác***

Tạo class InterbankSubsystem2 implement InterbankInterface:



Cài đặt thêm subsystem tương ứng bằng cách tạo thêm 2 class: InterbankBoundary2 và InterbankSubsystemController2:





Vì chưa có API thứ 2 nên ta sẽ để tạm như vậy.

## **II. Báo cáo thiết kế theo nguyên lý SOLID**

### ***Single Responsibility***

* Tạo controller PlaceRushOrderController riêng lẻ với PlaceOrderController để nó đảm nhận 1 nhiệm vụ (usecase)

### ***Open/Closed***

* Thêm interbank thứ 2 nhưng không ảnh hưởng đến interbank cũ

### ***Liskov Substitution***

### ***Interface Segregation***

* Tạo được các interface và các class kế thừa tương ứng (CalculateShippingFee, InterbankInterface)

### ***Dependency Inversion***

* Những thành phần trong chương trình phụ thuộc vào những cái trừu tượng(các interface class). Các thành phần sẽ giao tiếp thông qua các abstract method của interface và phần xử lý sẽ được thiết kế riêng dưới dạng các class con kế thừa.