

## 1. Single responsibility principle

- Class `ApplicationProgrammingInterface`: nên tách việc thiết lập connection với các method `get`, `post`
- Class `Cart` (package `entity.cart`): class vừa quản lý giỏ hàng vừa thực hiện tính toán
- Class `DeliveryInfo` (package `entity.shipping`): là class entity nên không cần thực hiện tính toán trực tiếp phí ship tại đây
- Class `PaymentController` (package `controller`): việc làm rỗng giỏ hàng (hàm `emptyCart`) nên để vào class quản lý giỏ hàng
- Class `PlaceOrderController` (package `controller`): nên tách các hàm `validate` vào class riêng

## 2. Open close principle

- Class `CreditCard` (package `entity.payment`): khó mở rộng nếu thay đổi/bổ sung phương thức thanh toán
- Class `HomeScreenHandler` (package `views.screen.home`): class này vừa thực hiện việc xử lý màn hình (hiển thị, điều hướng), vừa có chức năng liên quan đến giỏ hàng
- Class `InterbankPayloadConverter` (package `subsystem.interbank`): đang cố định xử lý với `CreditCard`, nếu có loại thẻ khác sẽ phải sửa class
- Class `InterbankSubsystemController` (package `subsystem.interbank`): đang cố định xử lý với `CreditCard`, nếu có loại thẻ khác sẽ phải sửa class
- Class `PaymentController` (package `controller`): nếu có thêm loại thẻ hay phương thức thanh toán thì sẽ cần sửa class
- Class `PlaceOrderController` (package `controller`): phụ thuộc vào `distanceCalculator`
- Class `PaymentTransaction` (package `entity.payment`): khó mở rộng nếu thay đổi/bổ sung phương thức thanh toán (do gán với `CreditCard`)