Bài thực hành số 5

Các khái niệm và nguyên lý thiết kế phần mềm

Họ và tên: Bùi Xuân Dương

MSSV: 20183901

I. Coupling và Cohesion

1. Coupling - UC Place Rush Order

a. Content coupling  
b. Stamp coupling

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| PlaceOrderController | Phương thức caculateShippingFee nhận tham số là đối tượng Order nhưng chỉ sử dụng thuộc tính amount của lớp Order | Chỉ cần chuyền vào tham số kiểu float để đại diện cho thuộc tính amount của đối tượng Order |
| PlaceRushOrderController | Phương thức caculateShippingFee nhận tham số là đối tượng Order nhưng chỉ sử dụng thuộc tính amount của lớp Order | Chỉ cần chuyền vào tham số kiểu float để đại diện cho thuộc tính amount của đối tượng Order |

2. Cohesion – UC Place Rush Order

1. Considental cohesion
2. Logical cohesion
3. Temporal
4. Procedural cohesion

II. Nguyên lý thiết kế SOLID

1. UC PlaceRushOrder
2. Single Responsibility Principle

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| PlaceRushOrderController | Lớp này đang thực hiện 2 nhiệm vụ : (1) là check validate của các trường thông tin; (2) Tính phí | Tách các phương thức validation và tính phí ra một lớp riêng |
| PlaceOrderController | Lớp này đang thực hiện 3 nhiệm vụ : (1) Thực hiện tạo đơn hàng; (2) là check validate của các trường thông tin; (3) Tính phí | Tách các phương thức validation và tính phí ra một lớp riêng |

1. Open/Closed Principle

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| PlaceRushOrderController (caculateShippingFee) | Nếu ta muốn tính phí ship theo kiểu khác, hoặc cần lưu nhiều kiểu tính phí ship khác nhau, thì chúng ta sẽ phải sửa đoạn code trên bằng một đoạn code với cách tính hoàn toàn khác | Tạo ra một interface ShippingFeeCalculator với phương thức trừu tượng là calculateShippingFee |
| PlaceOrderController (caculateShippingFee) | Nếu ta muốn tính phí ship theo kiểu khác, hoặc cần lưu nhiều kiểu tính phí ship khác nhau, thì chúng ta sẽ phải sửa đoạn code trên bằng một đoạn code với cách tính hoàn toàn khác | Tạo ra một interface ShippingFeeCalculator với phương thức trừu tượng là calculateRushShippingFee |

1. Liskov Substitution

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| Media | Phương thức Media.getAllMedia() được kì vọng trả về một List, tất cả các class con override lại phương thức này nhưng lại trả về null | Xoá đoạn code Override trong các lớp con vì không Override thì phương thức vẫn k bị ảnh hưởng |

1. Interface Segregation
2. Dependency Inversion