Bài thực hành 05 – Tuần 12, ngày 21/12/2021

Mục lục

[I. Các yêu cầu mở rộng 3](#_Toc91260823)

[1. Thay đổi phí ship 3](#_Toc91260824)

[B1: Tạo Interface ShippingFeesInterface 3](#_Toc91260825)

[B2: Tạo Class ShippingFeesController implement ShippingFeesInterface: 4](#_Toc91260826)

[B3: Để gọi phương thức tính phí 4](#_Toc91260827)

[2. Thêm một cách thức thanh toán mới 5](#_Toc91260828)

[B1: Tạo Entity DebitCard với các thuộc tính như trên 5](#_Toc91260829)

[B2: Thêm các phương thức thanh toán mới vào InterbankInterface 5](#_Toc91260830)

[B3: Cài đặt phương thức đã thêm ở Interface vào lớp InterbankSubsystem 6](#_Toc91260831)

[3. Sử dụng interbank khác 6](#_Toc91260832)

[B1: Cài đặt lớp NationalBankSubsystem implement InterbankInterface 6](#_Toc91260833)

[B2: Cài đặt các phương thức cần thiết 7](#_Toc91260834)

[II. Bài tập 7](#_Toc91260835)

[1. Coupling và Cohesion 7](#_Toc91260836)

[1.1. Coupling 7](#_Toc91260837)

[1.1.1. Content coupling 7](#_Toc91260838)

[1.1.2. Control coupling 7](#_Toc91260839)

[1.1.3. Common coupling 7](#_Toc91260840)

[1.1.4. Stamp coupling 8](#_Toc91260841)

[1.1.5. Data coupling 8](#_Toc91260842)

[1.2. Cohesion 8](#_Toc91260843)

[1.2.1. Coincidental Cohesion 8](#_Toc91260844)

[1.2.2. Logical Cohesion 8](#_Toc91260845)

[1.2.3. Temporal Cohesion 8](#_Toc91260846)

[1.2.4. Procedural Cohesion 8](#_Toc91260847)

[1.2.5. Communicational Cohesion 9](#_Toc91260848)

[1.2.6. Coincidental Cohesion 9](#_Toc91260849)

[1.2.7. Coincidental Cohesion 9](#_Toc91260850)

[2. Nguyên lí thiết kế SOLID 9](#_Toc91260851)

[2.1. Single Responsibility Principle 9](#_Toc91260852)

[2.2. Open/Closed Principle 10](#_Toc91260853)

[2.3. Liskov Substitution 10](#_Toc91260854)

[2.4. Interface Segregation 10](#_Toc91260855)

[2.5. Dependency Inversion 11](#_Toc91260856)

# Các yêu cầu mở rộng

## Thay đổi phí ship

Graphical user interface, text, application

Description automatically generatedĐể thay đổi cách tính phí ship mà không phải sửa phương thức đã viết, giải pháp là tạo một Interface làm cầu nối giữa các lớp cài đặt phương thức calculateShippingFees với lớp PlaceOrderController, chứ không cài đặt trực tiếp nữa. Trong package controller, tạo thêm giao diện ShippingFeesInterface. Sau đó tạo lớp ShippingFeesController để implement Interface này.

### Graphical user interface, text, application, email Description automatically generatedB1: Tạo Interface ShippingFeesInterface

### B2: Tạo Class ShippingFeesController implement ShippingFeesInterface:

Graphical user interface, text, application

Description automatically generatedPhương thức giả định, tính phí ship bằng 10% giá trị đơn hàng)

### Graphical user interface, text, application Description automatically generatedB3: Để gọi phương thức tính phí

Khai báo một đối tượng thuộc Interface ShippingFeesInterface để sử dụng

## Thêm một cách thức thanh toán mới

Để thêm cách thức thanh toán mới, ta thêm phương thức mới vào Interface InterbankInterface, rồi cài đặt ở lớp InterbankSubsystem. Cụ thể: Hiện tại đang muốn thêm cách thức thanh toán bằng thẻ Debit Card, với các thông tin:

* Issuing bank, e.g., VietinBank
* Card number: 16 digits
* Valid-from date, e.g., 12/33
* Cardholder’s name, e.g., VU DUY MANH

### Graphical user interface, text, application, email Description automatically generatedB1: Tạo Entity DebitCard với các thuộc tính như trên

### B2: Thêm các phương thức thanh toán mới vào InterbankInterface

Text

Description automatically generated

### B3: Cài đặt phương thức đã thêm ở Interface vào lớp InterbankSubsystemText Description automatically generated

## Sử dụng interbank khác

Với yêu cầu này, ta thay thế InterbankSubsystem bằng một lớp khác. Giả sử ta có ngân hàng NationalBank, thì cần thiết kế lớp NationalBankSubsystem, và implement các phương thức cần thiết trong Interface InterbankInterface:

### B1: Cài đặt lớp NationalBankSubsystem implement InterbankInterface

Graphical user interface, text, application

Description automatically generated

### B2: Cài đặt các phương thức cần thiết

Graphical user interface, text, application, email

Description automatically generated Ở đây em cài đặt demo phương thức payOrder

# Bài tập

## Coupling và Cohesion

### Coupling

#### Content coupling

*Không có*

#### Control coupling

*Không có*

#### Common coupling

*Không có*

#### Stamp coupling

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 1 | RushInfoScreenHandler | Hàm calculateAdditionalFees nhận tham số là Order. Điều này là k cần thiết vì nó chỉ dùng phương thức setAdditionalFees và setRushInfo | Có thể cài đặt Interface cho những phương thức cần dùng và implement ở một lớp control khác |

#### Data coupling

*Các coupling về data là chấp nhận được*

### Cohesion

#### Coincidental Cohesion

*Không có*

#### Logical Cohesion

*Không có*

#### Temporal Cohesion

*Không có*

#### Procedural Cohesion

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 1 | PlaceOrderController | Các phương thức như validateAddress, validateName, validatePhone được gọi lần lượt chứ không liên quan về mặt chứ năng | Đưa các phương thức này vào một module nghiệp vụ chung (Một lớp hoặc một package) có nhiệm vụ validate các thông tin trên |
| 2 | PlaceRushOrderController | Các phương thức như validateDeliveryDate và validateProvince, cũng gặp vấn đề tương tự như trên | Đưa các phương thức này vào một module nghiệp vụ chung (Một lớp hoặc một package) có nhiệm vụ validate các thông tin trên |

#### Communicational Cohesion

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 1 | InterbankSubsystemController | Phương thức payOrder và refund được đặt trong cùng module vì trả về cùng kiểu dữ liệu PaymentTransaction chứ chức năng của chúng tương đối khác nhau | Tách module này thành các module nhỏ hơn để cung cấp phương thức payOrder và refund riêng cho lớp InterbankSubsystem |

#### Coincidental Cohesion

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 1 | Các lớp Entity như Cart, Invoice, Media, CartMedia và Order, cùng các ScreenHandler có liên quan | Các ScreenHandler cùng thao tác trên các attribute của các lớp | Không có |

#### Coincidental Cohesion

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 1 | API | Phương thức setUpConnection chỉ làm nhiệm vụ tạo connection đến server để gọi sau đó trong get và post | Có thể tạo một module riêng, làm nhiệm vụ kết nối tới API server |

## Nguyên lí thiết kế SOLID

### Single Responsibility Principle

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 1 | InterbankSubsystemController | Thực hiện cả công việc mapping thông tin từ thẻ, gọi API giao dịch và trả về kết quả | Tách ra 2 module con, thực hiện việc mapping thông tin thẻ và trả kết quả, còn controller chỉ làm nhiệm vụ gọi API giao dịch |

### Open/Closed Principle

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 1 | DebitCard, NationalBankSubsystem, InterbankInterface | Sau khi cài đặt yêu cầu mở rộng thì đã làm chưa tốt khi thêm cứng Entity DebitCard và thiết kế Interface chưa hợp lí | Tạo Entity thẻ thanh toán chung, để làm tham số trong các phương thức trong InterbankInterface, còn DebitCard, CreditCard thì cho kế thừa loại thẻ chung này |
| 2 | PlaceRushOrderController | Ban đầu cài đặt riêng lẻ, nên khi dùng phải gọi lại nhiều phương thức, và sử dụng chung attribute đã cài đặt trong PlaceOrderController | Kế thừa lại lớp PlaceOrderController để tận dụng các hàm có sẵn trong luồng chạy |

### Liskov Substitution

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 1 | Các lớp Controller | Đã kế thừa BaseController và hoạt động tốt, không gây ra lỗi | Không có |
| 2 | Các lớp CD, DVD, Book | Kế thừa lớp Media, tận dụng được các thuộc tính chung và tính đa hình | Không có |
| 3 | PlaceRushOrderController | Kế thừa PlaceOrderController nhưng không thể hoạt động riêng lẻ được mà vẫn phụ thuộc vào lớp cha | Có thể xem xét PlaceRushOrder là một use case riêng hoàn toàn, để xây dựng module điều khiển riêng, không cho extend từ PlaceOrderController nữa |

### Interface Segregation

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 1 | Các Interface | Đều nhỏ gọn, thực hiện đúng đủ chức năng và không gây dư thừa | Không có |

### Dependency Inversion

|  |  |  |  |
| --- | --- | --- | --- |
| # | Related modules | Description | Improvement |
| 1 | Các Interface | Đều nhỏ gọn, thực hiện đúng đủ chức năng và không gây dư thừa | Không có |
| 2 | Các lớp ScreenHandler liên quan đến Order | Các lớp này đang phụ thuộc chặt chẽ vào Entity Order, nếu có thay đổi về dữ liệu của Order thì các lớp này phải sửa theo | Tạo lớp OrderAbstract hoặc OrderInterface, và các lớp ScreenHandler chỉ tương tác với lớp này để tạo sự phụ thuộc ngược lại của Entity vào lớp giao diện |