Design for class "ProductDetailUI"

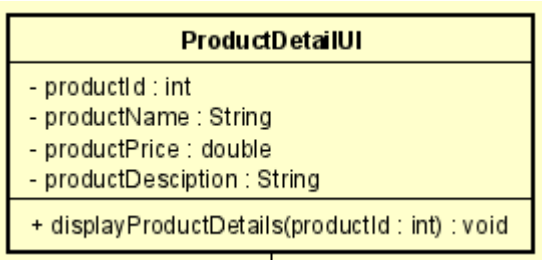

Table 1: Example of attribute design

#	Name	Data type	Default value	Desciption
1	productId	int	-1	ID sản phẩm
2	productName	String	“”	Tên sản phẩm
3	productPrice	double	0.0	Giá sản phẩm
4	productDesciption	String	“”	Mô tả sản phẩm

Table 2: Example of operation design

#	Name	Return type	Desciption(purpose)
1	displayProductDetails	void	Hiển thị chi tiết sản phẩm lên giao diện

Parameter:

- x: productId: int (ID của sản phẩm cần hiển thị)
- Default value, description: Không có giá trị mặc định, tham số này là bắt buộc để xác định sản phẩm cần hiển thị.

Exception:

- AException if ...: ProductNotFoundException nếu không tìm thấy sản phẩm với productId được cung cấp.
- BException if ...: InvalidInputException nếu productId là số âm hoặc không hợp lệ.

Method:

- **How to use parameters/attributes:**
  Phương thức displayProductDetails sử dụng tham số productId để gọi ProductController và lấy thông tin sản phẩm. Sau khi nhận được dữ liệu, phương thức cập nhật các thuộc tính productName, productPrice, và productDescription để hiển thị lên giao diện.
- **Flowchart/activity diagram/sequence diagram if the method has a complex/special algorithm:**
  Luồng hoạt động:
  1. Nhận productId từ người dùng.
  2. Gọi ProductController.getProductDetails(productId) để lấy thông tin sản phẩm.

3. Cập nhật các thuộc tính của ProductDetailUI (productName, productPrice, productDescription).
4. Hiển thị thông tin lên giao diện người dùng.
   Sequence diagram:
   ProductDetailUI -> ProductController: getProductDetails(productId)
   ProductController -> ProductService: fetchProductById(productId)
   ProductService -> ProductRepository: findById(productId)
   ProductRepository -> ProductService: return Product
   ProductService -> ProductController: return Product
   ProductController -> ProductDetailUI: return Product
   ProductDetailUI: Update attributes and display

State:

- **State diagram if any:**
  - Idle: Trạng thái chờ, khi chưa có sản phẩm nào được chọn.
  - Displaying: Trạng thái đang hiển thị chi tiết sản phẩm.
  - Chuyển đổi:
    - Idle -> Displaying: Khi displayProductDetails được gọi và dữ liệu tải thành công.
    - Displaying -> Idle: Khi người dùng đóng giao diện hoặc chọn sản phẩm khác.

Design for class "ProductController"

Table 1: Example of attribute design

Không có thuộc tính

Table 2: Example of operation design

#	Name	Return type	Desciption(purpose)
1	**getProductDetails**	**Product**	**Lấy chi tiết sản phẩm từ tầng nghiệp vụ**

Parameter:

- x: productId: int (ID của sản phẩm cần lấy)

- Default value, description: Không có giá trị mặc định, tham số này là bắt buộc.

Exception:

- AException if ...: ProductNotFoundException nếu không tìm thấy sản phẩm.
- BException if ...: InvalidInputException nếu productId không hợp lệ.

Method:

- **How to use parameters/attributes:**
  Phương thức getProductDetails nhận productId và gọi ProductService.fetchProductById(productId) để lấy thông tin sản phẩm. Trả về đối tượng Product cho ProductDetailUI.
- **Flowchart/activity diagram/sequence diagram if the method has a complex/special algorithm:**
  Luồng:
  1. Nhận productId.
  2. Gọi ProductService.fetchProductById(productId).
  3. Trả về kết quả.

State:

- **State diagram if any:** Không có (lớp này là stateless).

Design for class "ProductService"

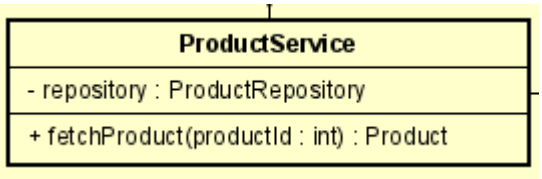

Table 1: Example of attribute design

#	Name	Data type	Default value	Desciption
1	**repository**	**ProductRepository**	**null**	**Tham chiến đến kho dữ liệu**

Table 2: Example of operation design

#	Name	Return type	Desciption(purpose)
1	**fetchProduct**	**Product**	**Lấy thông tin sản phẩm theo ID**

Parameter:

- x: productId: int (ID của sản phẩm cần lấy)

- Default value, description: Không có giá trị mặc định, tham số này là bắt buộc.

Exception:

- AException if ...: ProductNotFoundException nếu không tìm thấy sản phẩm.
- BException if ...: InvalidInputException nếu productId không hợp lệ.

Method:

- **How to use parameters/attributes:**
  Phương thức fetchProductById sử dụng thuộc tính repository để gọi ProductRepository.findById(productId). Trả về đối tượng Product cho ProductController.
- **Flowchart/activity diagram/sequence diagram if the method has a complex/special algorithm:**
  Luồng:
  1. Nhận productId.
  2. Gọi repository.findById(productId).
  3. Trả về kết quả.

State:

- **State diagram if any:** Không có (lớp này là stateless).

Design for class "ProductRepository"

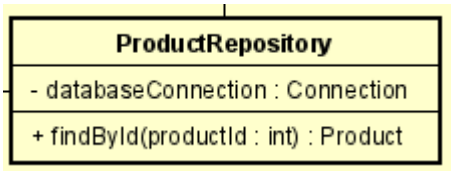

Table 1: Example of attribute design

#	Name	Data type	Default value	Desciption
1	**databaseConnection**	**Connection**	**null**	**Kết nối đến cơ sở dữ liệu**

Table 2: Example of operation design

#	Name	Return type	Desciption(purpose)
1	**findById**	**Product**	**Tìm sản phẩm theo ID trong cơ sở dữ liệu**

Parameter:

- x: productId: int (ID của sản phẩm cần tìm)

- Default value, description: Không có giá trị mặc định, tham số này là bắt buộc.

Exception:

- AException if ...: ProductNotFoundException nếu không tìm thấy sản phẩm.
- BException if ...: DatabaseException nếu có lỗi kết nối cơ sở dữ liệu.

Method:

- **How to use parameters/attributes:**
  Phương thức findById sử dụng databaseConnection để thực hiện truy vấn SQL (SELECT * FROM products WHERE id = productId). Tạo và trả về đối tượng Product từ kết quả truy vấn.
- **Flowchart/activity diagram/sequence diagram if the method has a complex/special algorithm:**
  Luồng:
  1. Nhận productId.
  2. Sử dụng databaseConnection để truy vấn cơ sở dữ liệu.
  3. Nếu tìm thấy, tạo đối tượng Product và trả về.
  4. Nếu không tìm thấy, ném ProductNotFoundException.

State:

- **State diagram if any:** Không có (lớp này là stateless).

Design for class "Product"

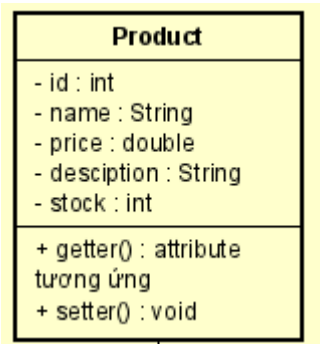

Table 1: Example of attribute design

#	Name	Data type	Default value	Desciption
1	id	int	-1	ID sản phẩm
2	name	String	“”	Tên sản phẩm
3	price	double	0.0	Giá sản phẩm
4	desciption	String	“”	Mô tả sản phẩm
5	stock	int	0	Số lượng sản phẩm trong kho

Table 2: Example of operation design

#	Name	Return type	Desciption(purpose)
1	**Getter**	**Thuộc tính ương ứng**	**Lấy thông tin thuộc tính sản phẩm tương ứng**
2	**Setter**	**void**	**Sủa thông tin thuộc tính sản phẩm tương ứng**

Parameter:

Exception:

AException if ...: InvalidInputException nếu các giá trị nhập vào Invalid.

Method:

- **How to use parameters/attributes:**
    - Getter: Trả về giá trị của thuộc tính.
    - Setter: Cập nhật giá trị của thuộc tính với tham số được cung cấp.
- **Flowchart/activity diagram/sequence diagram if the method has a complex/special algorithm:**
  Phương thức đơn giản (getter/setter), không cần biểu đồ.

State:

- **State diagram if any:** Không có.