

# Hệ thống Quản lý Sản phẩm E-commerce

Tên: Ngô Quang Trường

MSSV: 24110142

---

## 1. Phân tích OOA (Object-Oriented Analysis)

### Bước 1: Xác định đối tượng (Objects)

- Sản phẩm (Product)
- Sản phẩm điện tử (Electronic)
- Giỏ hàng (ShoppingCart)
- Đơn hàng (Order)
- Danh sách kho (InventoryList)
- Khả năng giảm giá (Discountable – interface)

### Bước 2: Xác định thuộc tính (Attributes)

- **Product:** id, name, price, quantity
- **Electronic:** (kế thừa Product) warrantyMonths, shippingFee, powerUsage
- **ShoppingCart:** items (InventoryList<Product\*>), totalPrice, totalQuantity
- **Order:** items (InventoryList<Product\*>), id, customerName, totalPrice, totalQuantity, status
- **InventoryList<T>:** items (vector<T>)

### Bước 3: Xác định phương thức (Methods)

- **Product:** printInformation(), updateQuantity(), applyDiscount(), operator==, operator<
- **Electronic:** printInformation(), estimateMonthlyUsage(), applyDiscount() (giới hạn giảm tối đa 30%)
- **ShoppingCart:** operator+=, operator-=, showCart(), clearCart(), removeProduct(), checkout()
- **Order:** confirmOrder(), displayOrder(), applyDiscount(), setStatus()
- **InventoryList<T>:** addItem(), removeItem(), findItem(), printAll()
- **Discountable:** applyDiscount() – pure virtual

### Bước 4: Xác định quan hệ kế thừa (Inheritance & Relationships)

- Electronic kế thừa Product.

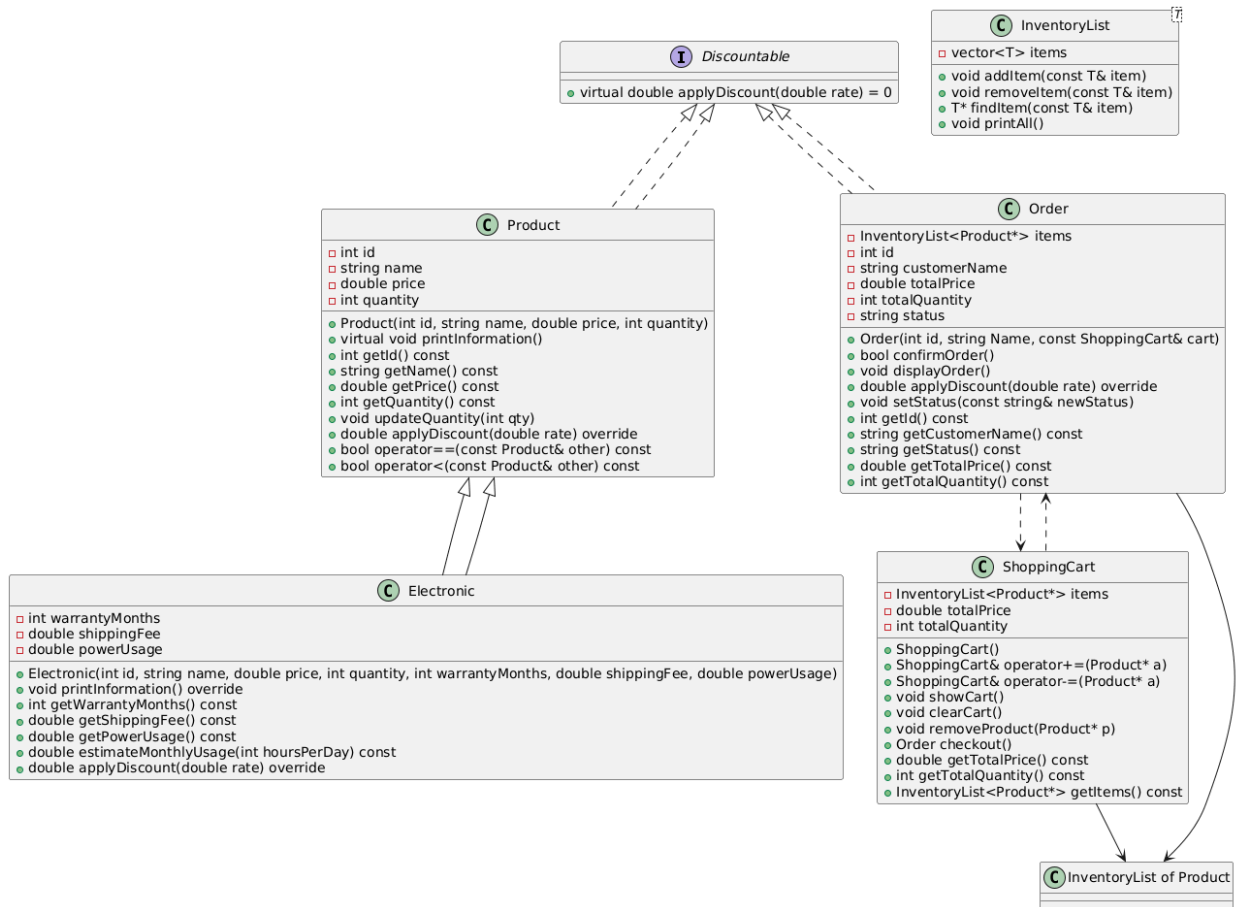
- Product và Order **thực thi interface** Discountable.
  - ShoppingCart và Order **chứa** InventoryList<Product\*> (composition).
  - ShoppingCart **tạo** Order (quan hệ tạo – creation).
- 

## 2. Giải thích thiết kế

- **Tính đóng gói (Encapsulation):**  
Tất cả thuộc tính được khai báo private/protected và truy cập qua getter/setter. Điều này đảm bảo tính an toàn và dễ bảo trì.
- **Kế thừa (Inheritance):**  
Electronic mở rộng từ Product để tái sử dụng thuộc tính cơ bản của sản phẩm, đồng thời thêm các đặc điểm riêng (bảo hành, phí vận chuyển, công suất điện).
- **Đa hình & interface:**  
Interface Discountable buộc mọi lớp cần giảm giá phải định nghĩa applyDiscount(). Nhờ đó, các lớp Product, Electronic và Order có thể được xử lý đa hình thông qua con trỏ Discountable\*.
- **Operator Overloading:**
  - ShoppingCart overload += và -= để thêm/xóa sản phẩm trực quan như cộng trừ.
  - Product overload == và < để so sánh sản phẩm theo tên/giá.  
Điều này làm cho code ngắn gọn và gần ngôn ngữ tự nhiên hơn.
- **Template Class:**  
InventoryList<T> là lớp khuôn mẫu dùng chung cho mọi kiểu dữ liệu, ở đây là Product\*. Nó cho phép quản lý danh sách sản phẩm linh hoạt, không phụ thuộc kiểu cụ thể.
- **Kiến trúc tổng thể:**
  - ShoppingCart tập hợp nhiều Product, khi checkout tạo ra Order.
  - Order giữ thông tin trạng thái đơn hàng và cho phép áp dụng giảm giá toàn đơn.

### 3. UML

#### 3.1 Class Diagram



#### 3.2 Sequence Diagram (Quy trình Checkout)

Mô tả tiến trình khi người dùng thanh toán giỏ hàng:

```
@startuml
actor Customer
Customer -> ShoppingCart : operator+= (Product)
Customer -> ShoppingCart : checkout()
ShoppingCart -> Order : create Order with items
Order -> Order : set status = "Pending"
ShoppingCart -> ShoppingCart : clearCart()
Order -> Customer : return Order
@enduml
```

#### 4. Mô tả kiểm thử

Quá trình kiểm thử được thực hiện thông qua bốn nhóm test case, phản ánh toàn bộ luồng nghiệp vụ của hệ thống:

##### Test Case 1 – Product Info:

```
===== TEST CASE 1: PRODUCT INFO =====  
Product information: Book  
- Id: 1  
- Price: 50000  
- Quantity available: 10  
  
Electronic Product information: Laptop  
- Id: 2  
- Price: 1.5e+07  
- Quantity available: 5  
Warranty: 24monts.  
Shipping Fee: 200000VND  
Power Usage: 65 W
```

Chương trình tạo và hiển thị thông tin hai loại sản phẩm: Book (thường) và Laptop (Electronic).

Kết quả in ra đầy đủ: id, giá, số lượng, cùng các thuộc tính mở rộng của Laptop (bảo hành 24 tháng, phí ship 200.000 VND, công suất 65 W).

→ Xác nhận lớp Product và lớp kế thừa Electronic hoạt động đúng, hàm printInformation() đã ghi đè thành công.

## Test Case 2 – Shopping Cart Add/Remove:

```
===== TEST CASE 2: SHOPPING CART ADD/REMOVE =====
Items in Shopping Cart:
Product information: Book
- Id: 1
- Price: 50000
- Quantity available: 9

Product information: Laptop
- Id: 2
- Price: 1.5e+07
- Quantity available: 4

Total Price: 1.505e+07
Total Quantity: 2

After removing Book:
Items in Shopping Cart:
Product information: Laptop
- Id: 2
- Price: 1.5e+07
- Quantity available: 4

Total Price: 1.5e+07
Total Quantity: 1
```

Thêm Book và Laptop vào giỏ, tổng giá trị và số lượng hiển thị chính xác (1.505e+07 VND – 2 sản phẩm).

Sau khi xóa Book, giỏ còn lại 1 Laptop, giá trị cập nhật về 1.5e+07 và số lượng về 1.

→ Kiểm chứng các toán tử +=, -= của ShoppingCart cùng hàm tính tổng giá, tổng số lượng vận hành đúng.

## Test Case 3 – Discount:

```
===== TEST CASE 3: DISCOUNT =====
Book price before discount: 50000
Book price after 20% discount: 50000
Laptop price with 50% discount (but max 30% allowed): 1.07e+07
```

Áp dụng giảm giá 20% cho Book nhưng giá không đổi (50.000) vì Book không cho phép giảm.

Laptop yêu cầu giảm 50% nhưng bị giới hạn 30%, kết quả còn 1.07e+07.

→ Xác nhận logic đa hình của applyDiscount() và giới hạn giảm giá riêng cho Electronic.

## Test Case 4 – Checkout:

```
===== TEST CASE 4: CHECKOUT =====
Items in Shopping Cart:
Product information: Laptop
- Id: 2
- Price: 1.5e+07
- Quantity available: 3

Product information: Book
- Id: 1
- Price: 50000
- Quantity available: 8

Product information: Laptop
- Id: 2
- Price: 1.5e+07
- Quantity available: 3

Total Price: 3.005e+07
Total Quantity: 3
Checking out...
Order Information:
- Order ID: 1
- Customer Name: Customer
- Status: Pending
- Total Quantity: 3
- Total Price: 3.005e+07
- Items in Order:
Product information: Laptop
- Id: 2
- Price: 1.5e+07
- Quantity available: 3

Product information: Book
- Id: 1
```

Giỏ hàng chứa nhiều sản phẩm (Laptop + Book), tổng giá trị 3.005e+07 với 3 sản phẩm.

Khi gọi checkout(), đơn hàng Order được tạo, trạng thái ban đầu là “Pending”.

Áp dụng giảm giá 10% cho toàn đơn, tổng giá trị giảm còn 2.7045e+07.

→ Kiểm tra thành công quan hệ giữa ShoppingCart và Order, khả năng tạo đơn và giảm giá tổng.

### Kết luận kiểm thử:

Tất cả chức năng chính—quản lý sản phẩm, thêm/xóa trong giỏ, áp dụng giảm giá, và tạo đơn hàng—đều hoạt động chính xác theo yêu cầu thiết kế. Việc hiển thị số liệu (giá trị tiền, số lượng, trạng thái) trùng khớp với mong đợi và không phát sinh lỗi biên dịch hay runtime.

## 5. Sử dụng LLM (Large Language Model)

Trong quá trình thực hiện, tôi đã sử dụng **ChatGPT** để:

- Tham khảo ý tưởng về thiết kế template class InventoryList.
- Hỏi về cú pháp chuẩn khi overload toán tử trong C++.

- Nhờ ChatGPT viết code của UML sau khi đã tạo xong các file header chứa các class
- Soạn file report mẫu và viết lại theo cách của tôi

## **6. Kết luận**

Hệ thống đáp ứng đầy đủ yêu cầu của đề tài: sử dụng kế thừa, đa hình qua interface, template class, và operator overloading. Thiết kế này linh hoạt, dễ mở rộng và phản ánh một quy trình thương mại điện tử cơ bản.