

VNUHCM-UNIVERSITY OF SCIENCE

FACULTY OF INFORMATION TECHNOLOGY

CSC10003 – OBJECT-ORIENTED PROGRAMMING

---

## OOP-Lab: Assignment 06

---

**Lecturer**

Mr. Nguyễn Lê Hoàng Dũng

Mr. Hồ Tuấn Thanh

**Class**

**23CLC08**

**Students**

23127255 - Nguyễn Thọ Tài



October 18th, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>The application</b>	<b>3</b>
2.1	Project structure . . . . .	3
2.1.1	Product class . . . . .	3
2.1.2	Slot class . . . . .	3
2.1.3	Inventory class . . . . .	4
2.2	Project runtime . . . . .	6
2.2.1	Main menu . . . . .	6
2.2.2	Add product . . . . .	6
2.2.3	Display inventory and prices . . . . .	9
2.2.4	Change slot quantity . . . . .	10

# 1 Introduction

This report explains a simple console application in C++ to simulate the following shopping cart in Shopee as the image bellow.

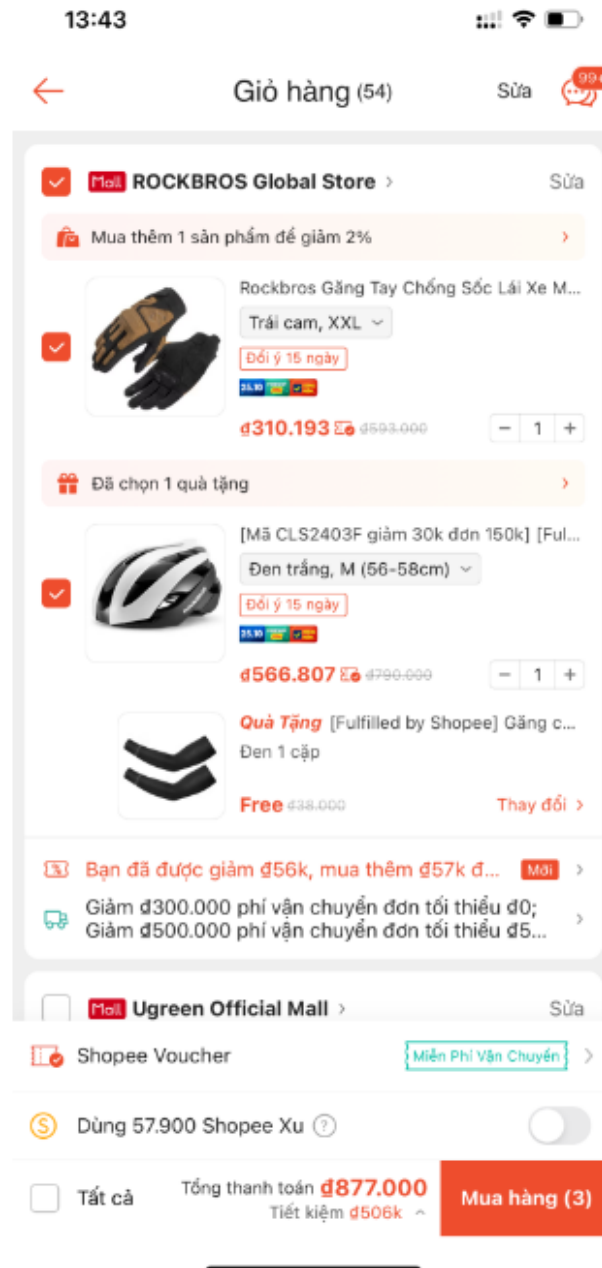


Figure 1: Shoppe's shopping cart

## 2 The application

### 2.1 Project structure

The project contains 5 files, with 3 type of class.

- Product(name, size, shop, price)
- Slot(product, quantity)
- Inventory (slots)

#### 2.1.1 Product class

The Product class contain information about the an item's name, size, shop, and its price as presented here as an UML.

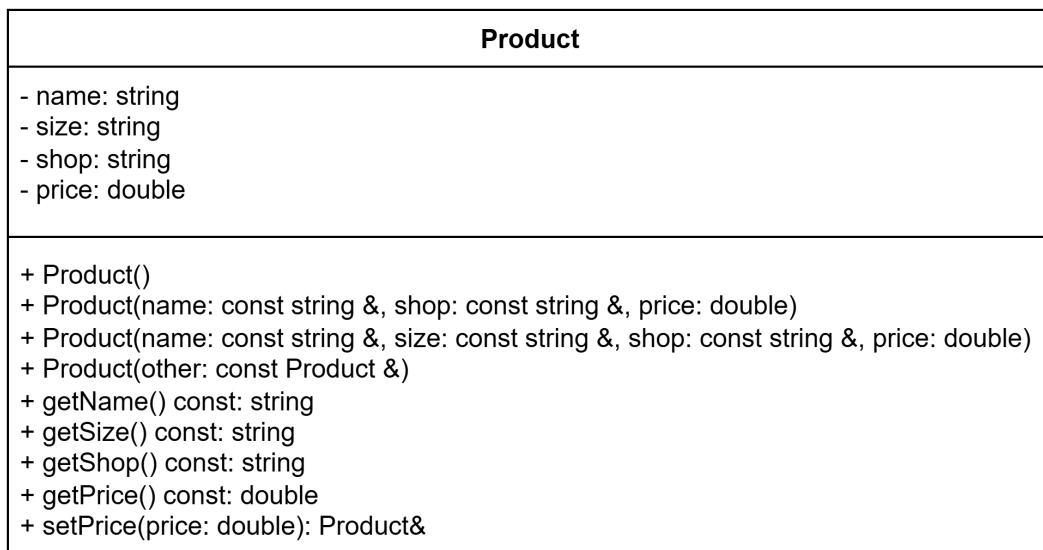


Figure 2: Product UML

Its role mainly is saving the information with no specical functions.

#### 2.1.2 Slot class

Similar to the Product class, its ownly role is to save informations. It likes a bundle which contain the products and how much is the products.

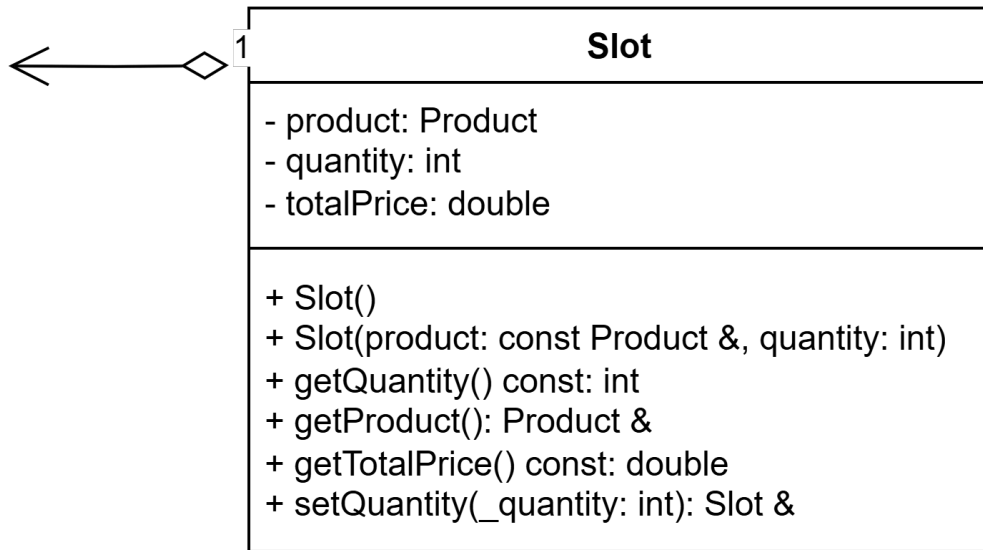


Figure 3: Slot UML

### 2.1.3 Inventory class

The inventory is on otherhand, not only saving all the Slots but also do the calculation and output. It also validate the slot and manage the products and slots inside it.

Its design is a singleton, which mean there is only one of it for all instances. This would help easier to manage the inventory on everywhere without the need of having to save a reference to the inventory.

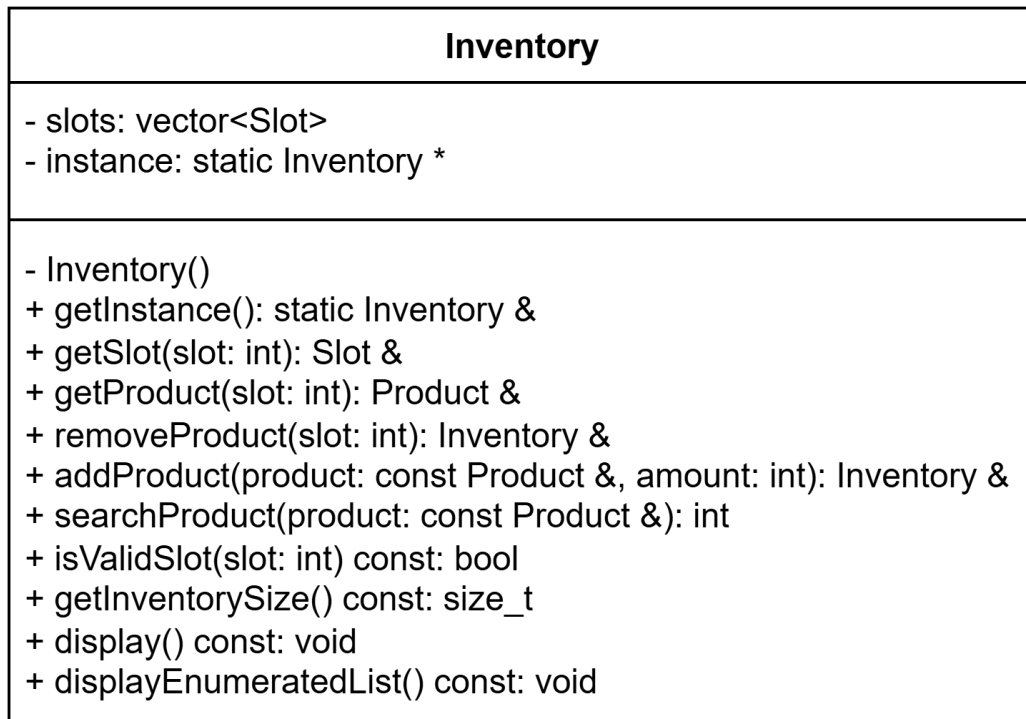


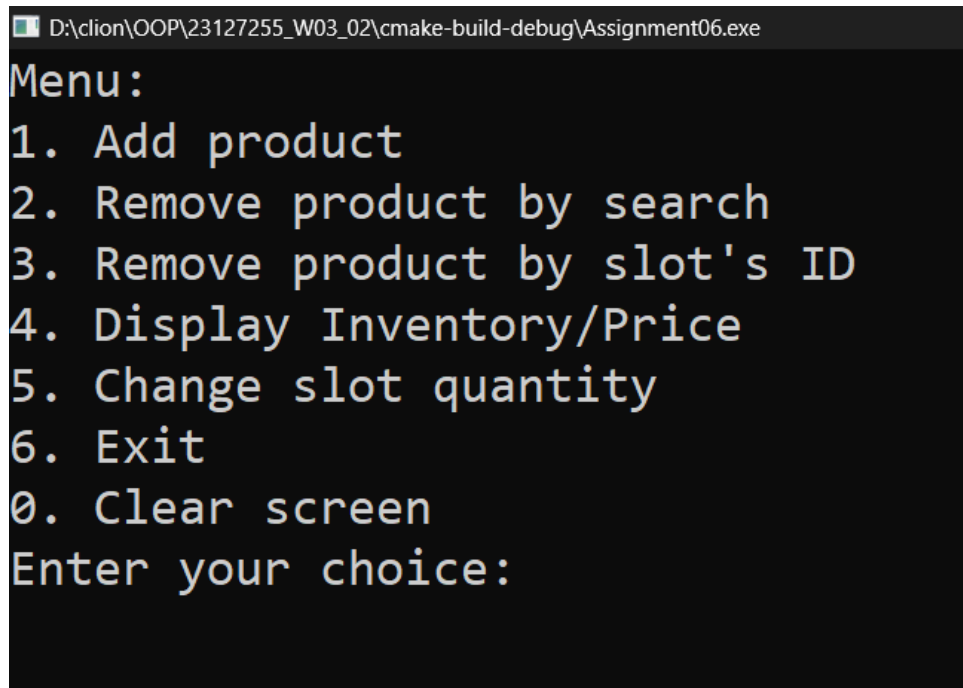
Figure 4: Inventory UML

Methods detail:

- addProduct(): add a product to the inventory
- removeProduct(): remove the product at slot position
- searchProduct(): search for the product if it exist in the inv. If yes, then return the position of the product in the inventory. If not return -1.
- display(): display the shopping cart with total price calculated.
- displayEnumeratedList(): display the inventory as a list for minimal views and without the price calculating.

## 2.2 Project runtime

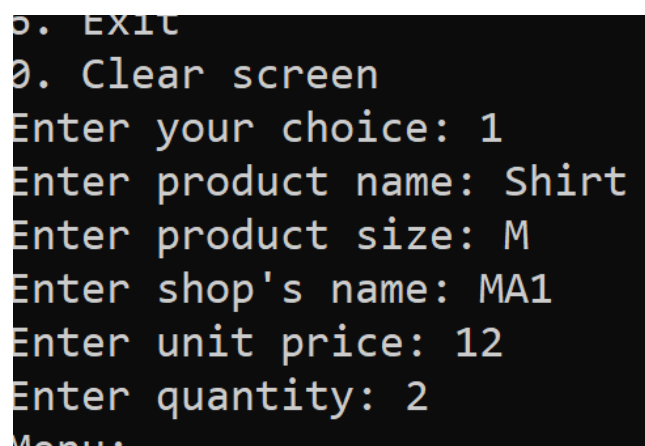
### 2.2.1 Main menu



```
D:\clion\OOP\23127255_W03_02\cmake-build-debug\Assignment06.exe
Menu:
1. Add product
2. Remove product by search
3. Remove product by slot's ID
4. Display Inventory/Price
5. Change slot quantity
6. Exit
0. Clear screen
Enter your choice:
```

Figure 5: Main menu

### 2.2.2 Add product



```
5. EXIT
0. Clear screen
Enter your choice: 1
Enter product name: Shirt
Enter product size: M
Enter shop's name: MA1
Enter unit price: 12
Enter quantity: 2
Menu:
```

Figure 6: Add product

When you re-add a same product, it will add up the quantity but not add the new product as its separated slot.





### 2.2.3 Display inventory and prices

```
0. Clear screen
Enter your choice: 4
SHOP: MA1
Name:      Shirt
Size:      M
Unit Price: 12
Quantity:  2
Price:     24

Name:      Dress
Size:      L
Unit Price: 134
Quantity:  3
Price:     402

      - = -
SHOP: MA2
Name:      Bo
Size:      123
Unit Price: 12
Quantity:  3
Price:     36

      - = -

-----
TOTAL PRICE: 462
Menu:
```

Figure 7: View inventory and prices

### 2.2.4 Change slot quantity

You can change a slot quantity

```
Menu:
1. Add product
2. Remove product by search
3. Remove product by slot's ID
4. Display Inventory/Price
5. Change slot quantity
6. Exit
0. Clear screen
Enter your choice: 5

ENUMERATED INVENTORY:
Idx | Product Name | Size | Shop | Price
0   | Shirt        | M    | MA1  | 12.00
1   | Dress        | L    | MA1  | 134.00
2   | Bo           | 123  | MA2  | 12.00
Enter idx (enter -1 to return to main menu):
```

Figure 8: Change slot quantity