**姓名： \_\_\_\_\_庞晓宇\_\_\_\_\_\_\_**

**学号： \_\_\_\_2020118100\_\_\_\_**

**时间：\_2021年10月29日\_**

**编号：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Project: ShoppingCart**

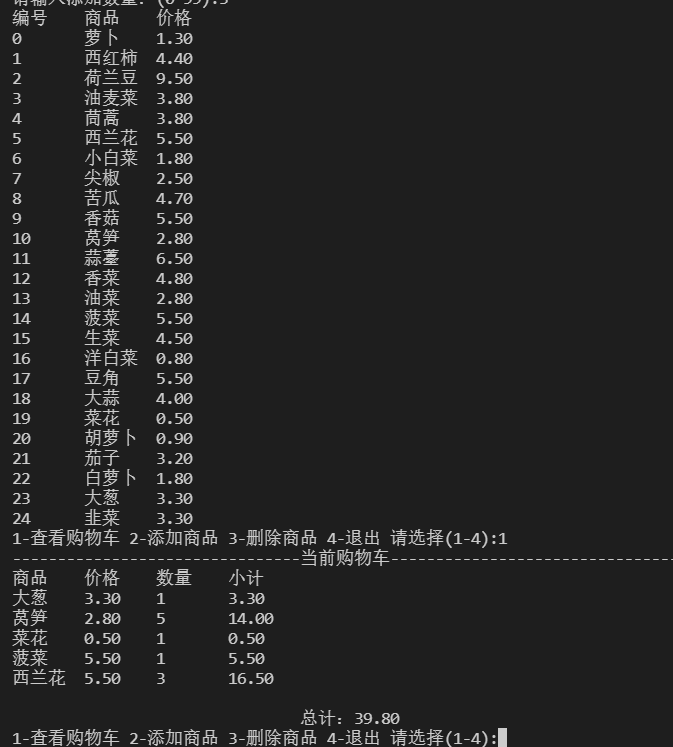
**问题描述：**

Description: Design and implement a shopping cart system that allows the user to enter in various products and display them to potential buyers in the store. The buyer can choose an item, place it in their shopping cart and go through checkout to calculate the subtotal, tax and final purchase total. Tips: Here we have another instance of Products being an object. The shopping cart will also be an object that will hold these product objects. To figure out the subtotals you would loop through the cart, add up the sum (by getting the price from the product object and multiply it by the quantity the buyer wants), calculate the tax based on this total and add it to the subtotal to find the order total. Depending on the language/medium you are using to implement this project, you may have to keep track of a session so that after putting an item in the cart, the user can continue to shop without losing those items. Then when they are ready to buy, the cart will contain all the products. Make sure the cart has the option of removing items and changing the quantity of a given item. Added Difficulty: Save the contents of the shopping cart between sessions (opening/closing of the program or between visits to a web site). Add in bundling where two items can be bundled together for a discounted price. Here bundles may be another object type which also has a discount property and contains a list of bundled items.

**设计:**

UML类图：见./UML.pdf

UI



**代码:**

│ fileOut.txt

│ goods.txt

│ README.md

│ ShoppingCart.docx

│ UML.pdf

│

├───.vscode

│ settings.json

│

├───bin

│ └───com

│ └───xftxyz

│ └───shoppingcart

│ ├───domain

│ │ Goods.class

│ │

│ ├───service

│ │ GoodFactory.class

│ │ ShoppingCart.class

│ │ ShoppingException.class

│ │

│ └───view

│ SCUtility.class

│ ShopView.class

│

├───lib

└───src

└───com

└───xftxyz

└───shoppingcart

├───domain

│ Goods.java：商品

│

├───service

│ GoodFactory.java：商品工厂

│ ShoppingCart.java：购物车

│ ShoppingException.java：自定义异常类

│

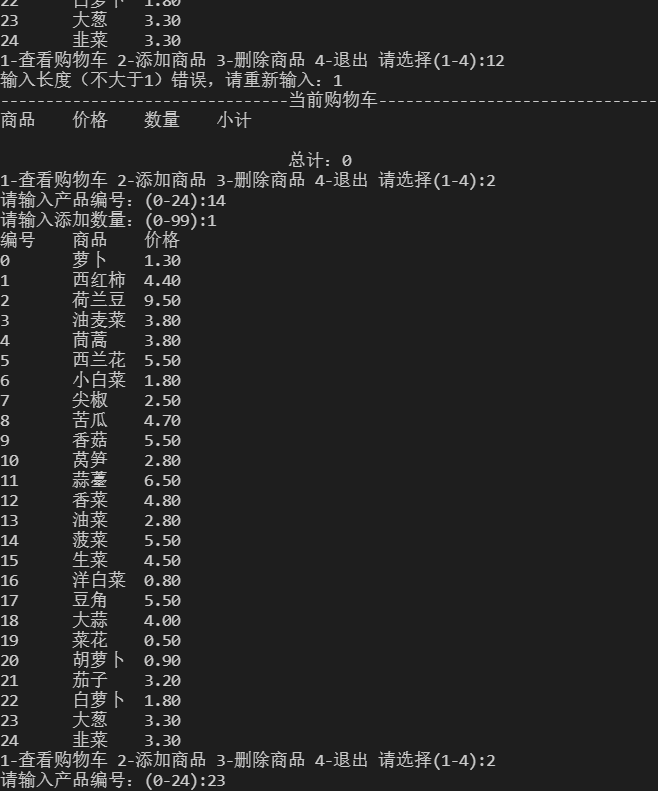
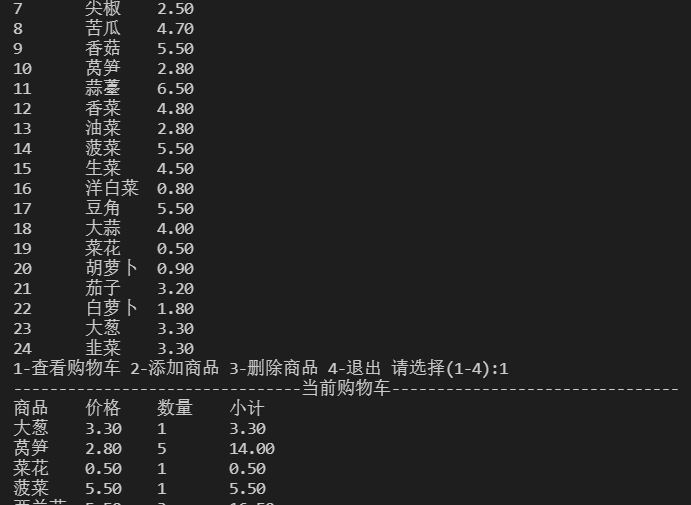
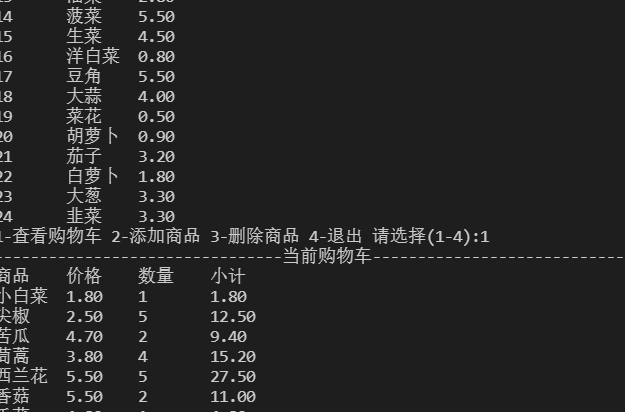
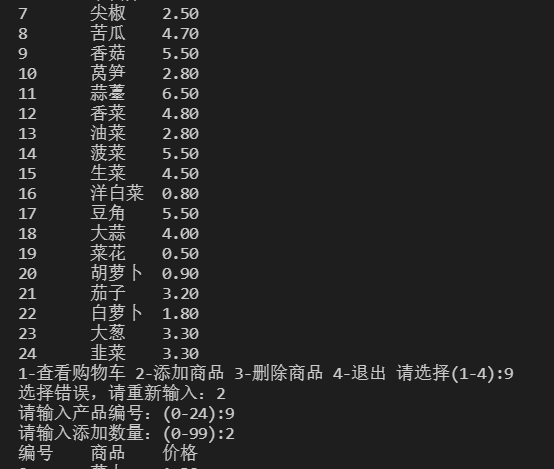
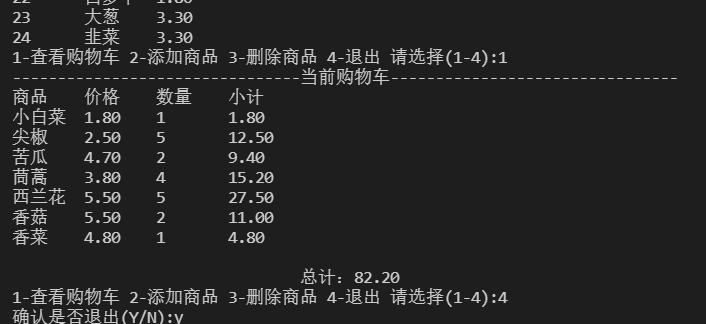
└───view

SCUtility.java：输入类

ShopView.java：主视图

另外以附件形式提交源代码

**运行程序的屏幕截图：**



**自我评价：**

测试结果；功能；界面；……

程序能够正常的添加、删除商品，查看购物车会详细的展示出每件商品的价格、数量、小结，以及购物车中所有商品的总价格。

程序ui实现的相对简陋，没有采用图形化界面来进行实现。另外商品由商品工厂保存，不能添加新的商品。