REIZ TECH HOMEWORK ASSIGNMENT DOCUMENT FOR THE JAVA INTERN POSITION

Thanks for your interest and for applying to REIZ TECH. To help us get to know you better, we have created this small task to see how you approach objectives and will use it as a discussion point going forward.

May The Odds Be Ever in Your Favour!

Once done, please give us the link to your online storage space (GitHub or similar, make sure your online storage space is public so we can see your magical code). We would also like you to give us your thoughts on this assignment (don't worry, you can be honest ;)).

Please reply to the original email with an answer.

---

1 Please implement a supermarket store which:

- Allows users to select products.
- Accepts bills and coins.
- Returns the selected product and remaining change, if any.
- Allows users to change their mind and cancel the transaction at any time

Note: product prices and coin values can be integers or, for bonus points, decimal numbers

2 Implementation requirements

- SupermarketService – an **interface** that defines the public API.
- SupermarketServiceImpl – the concrete implementation, the business logic. Make it a **singleton**.
- CashRegister – a class that represents the cash register of the supermarket, where coins and bills are Stored.
- Product – a class that models an item sold by the supermarket.
- ProductStorage – a class that represents the storage area of the supermarket where the items are stored.
- NotEnoughChangeException – Supermarket throws this **exception** to indicate that it doesn't have sufficient change to complete this transaction.
- SoldOutException – Supermarket throws this **exception** if the user requests a product which is sold out or does not exist.
- PayNotAcceptedException – Supermarket throws this **exception** if the user pays with non-accepted bills or coins.

Your system is required to have at least the components mentioned above. Beside these components you should implement any other components that you think you will need.

3 Steps should be

- ProductStorage is filled with products and CashRegister is filled with a finite amount of bills and coins
- supermarket lists all available products for the user
- user writes/selects the name of the product
- user inserts coin/bill (writes the bill/coin value in the console; only one unit per input)
- repeat previous step until amount is reached or surpassed
- supermarket provides change from the CashRegister, if necessary
- supermarket provides user with product from the ProductStorage
- if, at any point, an exception occurs, a nice message is displayed to the user

4 Bonus points

- Use floating point values (decimal numbers) for product prices (e.g., 2.3, 3.7 etc.) and for coin values (e.g., 0.1 and 0.5 coins)

- Use Java 8 Stream API at least once

Example end result:

```
---------------
Initial Product Inventory:
SODA Quantity: 10
BREAD Quantity: 10
WINE Quantity: 10
Initial Cash Inventory:
Value: 2, quantity: 50
Value: 1, quantity: 50
Value: 0.1, quantity: 50
Value: 0.5, quantity: 50
---------------

What would you like to buy? Type in the name of the desired product.
SODA (price: 2.3) BREAD (price: 1.1) WINE (price: 2.7)
SODA
You are trying to buy SODA. You need to pay 2.3
Provide bill or coin (accepted values: 0.1, 0.5, 1, 2)
2
You paid 2 in total. You still need to pay 0.3
Provide bill or coin (accepted values: 0.1, 0.5, 1, 2)
1
You paid 3 in total. Your change will be 0.7
Here is your change:
Value: 0.1, quantity: 2
Value: 0.5, quantity: 1
Here is your product: SODA

---------------
Updated Product Inventory:
SODA Quantity: 9
BREAD Quantity: 10
WINE Quantity: 10
Updated Cash Inventory:
Value: 2, quantity: 51
Value: 1, quantity: 51
Value: 0.1, quantity: 48
Value: 0.5, quantity: 49
---------------


What would you like to buy? Type in the name of the desired product.
SODA (price: 2.3) BREAD (price: 1.1) WINE (price: 2.7)
```

Good Luck!

Don't forget to reply to the original email with an answer.