Software Engineer Smart Shopping Cart Problem
ADP

Overview

Engineers at ADP spend the majority of their time developing production code. As such, code quality and design fundamentals are critical to success here. The following problem is intended to serve as the basis for the remainder of the interview process. Please devote some time to providing us with an implementation which reflects your ideals of code quality and proper unittesting.

During the on-site interview we will make reference to this problem and your solution. Please be prepared to conduct a more in-depth analysis when you arrive. This will include discussing implications for more real life scenarios; addressing issues such as scaling, configurability, validation, and varying inputs.

If you have any questions concerning the problem we would be happy to clarify the requirements. Also, we have provided a sample Maven project. The attached project includes dummy classes and sample input to give you a starting point. You may use this sample project if you wish, but are not required to do so if you are not familiar with Maven.

Problem Description

Please write a program that can dynamically apply discounts on shopping cart items, based on their category. Besides, based on the total cart value, the program should be able to apply a final slab-wise discount.

For simplicity, and to work around database portability, assume the shopping cart data to be available in XML/ JSON format.

For every item added in the shopping cart, apply the discount based on the category of the item being added. Likewise, based on all added items, (& their associated discounts applied), calculate the grand total, based on 3 parameters: (Unit Price, Quantity, & its applicable discount) – for each item. Apply a final slab based discount on the grand total, to generate the net bill value.

Inputs, to the program:

1. XML/ JSON – List of Shopping Item categories, and their applicable discount %
2. XML/ JSON – List of items in shopping cart, their associated information, including their shopping category
3. XML/ JSON – Grand total slabs, & their applicable discount %

Outputs from the program:

1. Itemized bill – listing each purchase item, its quantity, unit price, discount, and net purchase amount for that item, (& quantity)
2. Grand total, applicable discount, and Net Bill Amount.

The program should be written in Java. Please provide the source code and unit tests in a runnable state. Feel free to use the accompanying Maven based project, but its use is not required if you are not familiar with Maven. Your unit tests should be written in either JUnit or TestNG.

You can spend as much time as you like on the project, but typically, we expect you to spend about 2-3 hours to come up with a solution and a backlog of ideas on how to improve the system.

Sample XMLs to define shopping categories, & discount slabs

```
<Categories>
	<Category>
	  <id>001</id>
	  <name>ConsumerGoods</name>
	  <discPerc>5</discPerc>
	</Category>
	<Category>
	  <id>002</id>
	  <name>OrganicFood</name>
	  <discPerc>7</discPerc>
	</Category>
	<Category>
	  <id>003</id>
	  <name>Grocery</name>
	  <discPerc>2</discPerc>
	</Category>
	<Category>
	  <id>004</id>
	  <name>BabyProducts</name>
	  <discPerc>10</discPerc>
	</Category>
	<Category>
	  <id>005</id>
	  <name>Apparrel</name>
	  <discPerc>15</discPerc>
	</Category>
</Categories>

<FlatDiscountSlabs>
	<Slab>
		<RangeMin>0</RangeMin>
		<RangeMax>3000</RangeMax>
		<discPerc>2</discPerc>
	</Slab>
	<Slab>
		<RangeMin>3001</RangeMin>
		<RangeMax>7000</RangeMax>
		<discPerc>4</discPerc>
	</Slab>
	<Slab>
		<RangeMin>7000</RangeMin>
		<RangeMax></RangeMax>
		<discPerc>5</discPerc>
	</Slab>
<FlatDiscountSlabs>
```

Sample Shopping Cart Items List

```
<ShoppingCart>
        <itemID>000001</itemID>
        <itemCategoryID>003</itemCategoryID>
        <itemName>Muesli</itemName>
        <unitPrice>100</unitPrice>
        <quantity>2</quantity>

        <itemID>000002</itemID>
        <itemCategoryID>005</itemCategoryID>
        <itemName>Mens Tshirt Arrow 3463</itemName>
        <unitPrice>1536</unitPrice>
        <quantity>1</quantity>

        <itemID>000003</itemID>
        <itemCategoryID>002</itemCategoryID>
        <itemName>Organic Tomatoes</itemName>
        <unitPrice>10</unitPrice>
        <quantity>2</quantity>

        <itemID>000004</itemID>
        <itemCategoryID>001</itemCategoryID>
        <itemName>Wipro CFL</itemName>
        <unitPrice>120</unitPrice>
        <quantity>1</quantity>
</ShoppingCart>
```