PGCert IT: Programming for Industry

Challenges 03 - Supermarket Inventory Systems

This challenge requires you to develop a simple application called Supermarket Inventory System.

Before attempting the challenge, read through and gain an understanding of the requirements.

Notes:

- This challenge is worth **3%** of your final Programming for Industry grade.
- You should have the following source files for the challenge:
    - `Keyboard.java`
    - `productList.csv`
- After completing the challenge, you should have a functional Java program.
- Zip your IntelliJ project for the challenge, including all the source files and submit the single Zip file before the due date.
- **IMPORTANT**: Read the instructions carefully before attempting each task.

Introduction

This exercise is adapted from McAllister, W., & Fritz, Jane. (2015). *Programming fundamentals using JAVA : A game application approach*.

Create an application for a supermarket inventory system. This application allows the user to input the basic information about a supermarket, such as name, location and inventory size. Then, the user is able to add various products to the supermarket inventory. The application provides the users the ability to view the information for all of the products in store as well as the information for the products under a particular category. The program also gives the user an alternative way to add products in the inventory list from a pre-filled csv file. The program should check and handle any exceptions.

Task One. Inventory Design

Design and create appropriate classes to fulfill the following requirements for the supermarket.

A supermarket has five categories in its inventory. These categories are: Produce, Meat, Deli, Bakery, and Cleaning. Every product in the supermarket has a name, a base price, and a quantity in stock. Each product under Produce, Meat, Deli or Bakery category has an expiry date and a weight in kgs. Each product under the Deli or Bakery category also has an indication of whether the product is suitable for vegan or not.

Task Two. Local Store Design

The supermarket inventory system has another class called LocalStore. The LocalStore class contains information about the name, the location of the supermarket and the maximum number of products that the supermarket can store. It also has a list of inventory. Design and implement the LocalStore class that can do the following:

- Set the supermarket name
- Set the location of the supermarket
- Specify the maximum number of products that the supermarket can store
- Add a new product to the supermarket's list of inventory. The system should print a message to notify if the item has been successfully entered. The system should also print a message when the inventory is full.
- Print information for all of the products stored in the supermarket
    - The information should include the product name, base price, and quantity in stock
- Print information for the products under a particular category stored in the supermarket
    - The information should include the product name, base price, and quantity in stock

- The information should also include the category name, and any additional attributes such as expiry date

Task Three. Main Class Design

Create a start class which initialises an instance of the LocalStore class. This class allows the user to enter the initial inventory, and then repeatedly outputs the inventory:

a) What is the name of the supermarket?
b) What is the location of the supermarket?
c) What is the maximum number of products that the supermarket can store?
d) Repeatedly present a menu to allow the user to select the generic category of a product and then request the information for that product until all information has been entered. Note that you should also provide an option to allow the user to see the list of products added, i.e. the menu described in item e.
e) Repeatedly present the user with the following menu until "q" is entered:
    - Enter 1 to output all the information for all of the products in the inventory
    - Enter 2 to output all the information for all of the products under a particular category
    - Enter 3 to add more products to the inventory
    - Enter q to quit the program

    Note that if the user enters 2, the prompt should display more options for the user to choose the category of products. If the user enters 3, the prompt should return to item d.

Task Four. Import Inventory

Modify your program so that it can add products to the supermarket's list of inventory from a pre-filled csv file. Take a look at the `productList.csv` file to see the expected input format.

Task Five. Exception Handling

Check and modify your program so that the users can use the program smoothly without crashing. For example, what happens if someone enters a non-integer value when the program expects an integer value?

Marking Guide

This challenge will be marked based on the correctness of your program and your programming style. Here are some questions to consider for the programming style:

- Is the code well-structured?
- Is the code self-explanatory?
- Can you understand the code easily?
- Are all variables properly defined with meaningful and clear code?

- Is there any commented out code?

Marks will be deducted if your code cannot be compiled or has a bad programming style.

All required functions implemented correctly including proper exception handling; Good software design and programming style (appropriate classes, code reuse, good structure, variable names, comments, etc).	*5 marks*
All required functions implemented correctly; Adequate software design and programming style.	*3 marks*
All basic functions implemented correctly; Poor software design and programming style.	*1 marks*
Code cannot be compiled; Most functions not implemented;	*0 marks*