Online Store

Student Workshop 3w

Version 7.0 Y

Project Description

In this workshop you will work as a team. One team member should create a public GitHub repo for this workshop project.

Ensure that all team members are granted admin access to this GitHub repo, since all team members will need to contribute to the code.

You will build a simple command line Online Store application. The application will have a CLI user interface that is the store front for users to shop at your store.

Create a new GitHub project named `Online-Store` and clone it to your `workshops` directory.

Data Files

`Products.csv`

```
SKU|Product Name|Price|Department
AV1051|JBL Bluetooth Speaker|89.95|Audio Video
AV1312|Mini 1000 Lumens Projector|149.95|Audio Video
GM1148|Retro Handheld Arcade|24.45|Games
PW1001|Solar Powered Battery Charger|19.99|Electronics
...
```

Requirements

- Use the provided `products.csv` file to load the store's product inventory into your application.
- Create a `Product` class that stores all of the properties defined in the csv file
- Customers should be able to view all products
  - They should also be able to search by `Product Name, Price or Department`
- Customers should be able to add products to their cart
- Customers should be able to remove products from their cart

<u>Screens</u>

- **The Store Home Screen** - The home screen should display a list of options that a user can choose from.
  - Display Products
  - Display Cart
  - Exit - closes out of the application
- **Display Products** - Displays a list of products that your store sells.
  - On this screen the customer should be able to
    - `Search` or filter the list of products
    - `Add` a product to their cart
    - `Go Back` to the home page
- **Display Cart** - This displays a list of line items that are in the customer's cart. It should also display the total sales amount of the cart.
  - The customer should be able to:
    - `Check Out`
    - `Remove Product` from the cart
    - `Go Back` to the home screen
  - If the customer chooses to remove a product need to prompt them for the product to remove

BONUS (optional)

- **Check Out** - Here you should display the total sales amount owed for this order and prompt the user for payment.
    - Assume that the user will pay in cash
    - When the customer enters their payment amount, verify that the amount is sufficient to cover the cost of the cart
        - If the customer added enough money for the sale,
            - calculate the change that is owed to the customer and display the calculated change
            - print a sales receipt to the screen
                - Order Date
                - All Line items
                - Sales Total
                - Amount Paid
                - Change Given
            - clear the shopping cart
            - Return to the home screen
- When a customer adds multiple items to a cart
    - Instead of showing the item multiple times, your cart should only display each item once, but display the quantity for each item
- Create Sales Receipt File
    - After checkout, in addition to displaying the sales receipt to the screen, create a file with the sale information
        - Save the file in a `Receipts` folder
        - The file name should be the date and time stamp of the sale
            - `202303151148.txt`
        - All of the information printed to the screen should be saved to the file

What Makes a Good Workshop Project?

- **You should:**
  - Have a clean and intuitive user interface (give the user clear instructions on each screen)
  - Implement the ability for a customer to add/remove items to a cart and also to purchase the items in the cart

- **You should adhere to best practices such as:**
  - Create a Java Project that follows the Maven folder structure
  - Create appropriate Java packages and classes
  - Class names should be meaningful and follow proper naming conventions (PascalCase)
  - Use good variable naming conventions (camelCasing, meaningful variable names)
  - Your code should be properly formatted easy to understand
  - use Java comments effectively

- **Make sure that:**
  - Your code is free of errors and that it compiles and runs

- **The GitHub Repo for your project should be public**
  - Include a README.md file that describes your project and includes screen shots of
    - ∗ your home screen
    - ∗ your products display screen
    - ∗ one calculator page that shows erroneous inputs and an error message.
  - ALSO make sure to include one interesting piece of code and a description of WHY it is interesting.