Assignment

Overview

The MCC Bookstores need to keep track of inventory of its products on hands. You have been tasked with writing an application that reads in the text file of each product, allows the person to add inventory, make sales(subtracting from inventory) and adding in new items(if they are in fact new).

A lot of the code is written for you including an Item class, a system file that just has some test add, subtract and new item calls. What you need to do is first code the file to read in the data from the provided file, save each line as an Item and add that item to the Arraylist. Second you must write the method code to write out to the same file updating the data stored.

Directions

To Get Started:

1. Create a new project in IntelliJ(or your preferred IDE) called YOUR_MCC_Name-Assignment1

2. Download and extract the files needed for the assignment: **1531Assignment1Starters.zip**

- The java files, Item.java, InventoryDB.java and InventorySystem.java should go in your **src** folder
- The storeproducts.txt file is our data and should go in your project folder

3. Look through the the Item, InventoryDB and InventorySystem files. The InventoryDB is where you will be doing your coding. See that it has instance variables of an ArrayList and a String for the filename to open. The InventorySystem creates an InventoryDB

object sending the name of the file. So you shouldn't need to hardcode the file name when opening but rather use the variables provided.

Open up the storeproducts.txt file. Note the first lines are headings. Often times headings are in the file to help let you know what the type of data is. You will need to skip past this first line. Basically read it then read past it with another read. Content is divided up via **tabs** in the file

Part 1: Read in the content

Open the InventoryDB.java file. Make sure to reference the Javadoc comment under the author part, put your name there. Put your resource statement in there as well.

Navigate down to the readInProducts() method that is defined. This is your first task to complete. You will need to open the storeproducts.txt file, read in each line, split the data up, create an Item for each record and add it to the ArrayList.

- Use the correct delimiter for splitting- tabs
- Parse text data to numbers for inventory and prices
- Use a basic IOException catching here and print out the exception
- Make note that the first line in the text file is the headers not the data to convert, skip it with another read, otherwise you will get errors on the parsing

Testing:

How do you know if the reading works? In the InventorySystem.java file I have provided you a way to do it. On line 21 there is a db.displayProducts() command that can be uncommented that will print out the list. When you run the file it should print out the following:

```
Name: College ID Holder
Manufacturer: LXG
Product Code: CG-8052
Inventory: 8
Price: $11.95
Name: Key Strap
Manufacturer: Spirit
Product Code: KT269
Inventory: 12
Price: $0.49
Name: MCC T-Shirt - XL
Manufacturer: Champion
Product Code: 030631
Inventory: 8
Price: $20.00
```

```
Name: MCC T-Shirt - L
Manufacturer: Champion
Product Code: 030632
Inventory: 10
Price: $20.00
Name: MCC T-Shirt - M
Manufacturer: Champion
Product Code: 030633
Inventory: 13
Price: $20.00
Name: MCC T-Shirt - S
Manufacturer: Champion
Product Code: 030634
Inventory: 5
Price: $20.00
Name: MCC Long Sleeve - XL
Manufacturer: Champion
Product Code: 040631
Inventory: 5
Price: $26.00
Name: MCC Long Sleeve  - L
Manufacturer: Champion
Product Code: 040632
Inventory: 4
Price: $26.00
Name: MCC Long Sleeve  - M
Manufacturer: Champion
Product Code: 040633
Inventory: 7
Price: $26.00
Name: MCC Long Sleeve  - S
Manufacturer: Champion
Product Code: 040634
Inventory: 15
Price: $26.00
Name: MCC Trucker Cap
Manufacturer: Legacy
Product Code: 067101
Inventory: 23
Price: $28.00
Name: MCC Hooded Sweatshirt - XL
Manufacturer: Champion
Product Code: 070631
Inventory: 10
Price: $44.00
Name: MCC Hooded Sweatshirt - L
Manufacturer: Champion
Product Code: 070632
Inventory: 8
```

```
Price: $44.00
Name: MCC Hooded Sweatshirt - M
Manufacturer: Champion
Product Code: 070633
Inventory: 7
Price: $44.00
Name: MCC Hooded Sweatshirt - S
Manufacturer: Champion
Product Code: 070634
Inventory: 7
Price: $44.00
Name: Snickers
Manufacturer: Mars
Product Code: 401020
Inventory: 35
Price: $1.43
Name: Milky Way
Manufacturer: Mars
Product Code: 006947
Inventory: 25
Price: $1.62
Name: 3 Musketeers
Manufacturer: Mars
Product Code: 246039
Inventory: 32
Price: $1.55
Name: Reese's Peanut Butter Cups
Manufacturer: Hershey
Product Code: 004409
Inventory: 28
Price: $1.25
Name: Hershey's Chocolate Bar
Manufacturer: Hershey
Product Code: 240005
Inventory: 18
Price: $1.09
Name: PayDay
Manufacturer: Hershey
Product Code: 807229
Inventory: 72
Price: $1.65
Name: Pepsi
Manufacturer: Pepsi
Product Code: 182952
Inventory: 48
Price: $2.25
Name: Pepsi Zero
Manufacturer: Pepsi
Product Code: 182955
```

```
Inventory: 36
Price: $2.25
Name: Mountain Dew
Manufacturer: Pepsi
Product Code: 182476
Inventory: 24
Price: $2.25
Name: Diet Mountain Dew
Manufacturer: Pepsi
Product Code: 182478
Inventory: 32
Price: $2.25
Name: Starry
Manufacturer: Pepsi
Product Code: 182322
Inventory: 51
Price: $2.25
Name: Dr. Pepper
Manufacturer: Dr Pepper Snapple Group
Product Code: 237834
Inventory: 158
Price: $2.25
Name: Diet Dr. Pepper
Manufacturer: Dr Pepper Snapple Group
Product Code: 237835
Inventory: 25
Price: $2.25
Name: M&M
Manufacturer: Mars
Product Code: 359685
Inventory: 26
Price: $1.38
Name: Peanut M&M
Manufacturer: Mars
Product Code: 359852
Inventory: 26
Price: $1.43
Name: Peanut Butter M&M
Manufacturer: Mars
Product Code: 359998
Inventory: 26
Price: $1.63
```

Note: Comment that line of code when you are done so this doesn’t print every time.

Part 2: Write out the file

Now we need to write our modifications from the system file back to the text file (as if we got new inventory, sold products and got brand new products of items).

Under the InventoryDB.java file you will find the writeOutProducts() method that you need to complete. This method should open the storeproducts.txt and overwrite the file completly with the ArrayList content

- The file started with headers, it should have those written in first before content
- Remember to add the same delimiters back when writing.
- Basic IOException should be caught
- Note that the file is stored in fileName variable

Look at the InventorySystem file and note on lines 26 - 44 what is happening. These are adding, selling and creating new items. We are doing this via IDs. To make sure your program overwrites correctly find each of those items in the text file and make sure that the quantities are different.

Note: each time you run the program it adds/subtracts more, that is fine just make sure it is doing it correctly.

Add to Inventory

ID	Name	Description
040634	MCC Long Sleeve - S	add 3
MMS02168	Hydroflask	add 10
237834	Dr. Pepper	add 28
807229	PayDay Candy Bar	add 12

Sold from Inventory

ID	Name	Description
182322	Starry	Sold 3
182476	Mountain Dew	Sold 10
044251	MCC Bistro Mug	Sold 5
040632	MCC Long Sleeve - L	Sold 1

Added new items of M&M, Peanut M&M and Peanut Butter M&M

Check the output: (User input bolded)

```
Inventory Updated
Inventory Updated
Inventory Updated
Item(s) Sold
Item(s) Sold
Item(s) Sold
New Item Added
New Item Added
New Item Added
```

Remember that once you run it you will add 3 new items of M&Ms to the list...so you will get the last three lines as different after the first run. This is fine if you want the original fun, delete the last 3 lines in the text file. You may also need to recopy the text file into your project for a fresh file to work with.

```
Inventory Updated
Inventory Updated
Inventory Updated
Item(s) Sold
Item(s) Sold
Item already in products.
Item already in products.
Item already in products.
```

Submission

1. Compress the IntelliJ project folder and submit it to this assignment.

to compress: On Windows, right click -> send to -> compressed .zip file

on Mac, right-click -> Compress

Hints/Tips (Before Submitting):

- *Note the comments in the InventorySystem file talk about the start values and what we are adding. Every time you run this file the same values will add more to the original. I am not grading against it if you run/test your code before submission. I will just be checking that values are updated. If needed grab a new copy of the storeproduct.txt file to help with your check.*
- *Don't forget to have a header at the top of your file and include a **Resource statement**.*
- *Use comments throughout for full points.*
- *Follow all Java Styling Guides as covered*