Assignment #2: Revise a Java program

Manage Inventory for an Online Store

CS 4354 Summer II 2014
Instructor: Jill Seaman

Due: before class **Wednesday, 7/16/2014** (upload electronic copy by 9:30am)

Problem:

The video store has expanded and now also sells used books and toys in addition to DVDs. Modify your Java program from assignment 1 to handle these different types of products in the inventory.

The inventory for the store will contain the following information for each product in the inventory:

SKU	(stock-keeping unit, an integer, must be unique)
quantity	(number of copies in inventory, greater than or equal to 0)
price	(dollars and cents, greater than 0)
title	(may contain spaces in it)

In addition to these attributes, the following are stored for each product type:
For **movies** (dvds) a upc (universal product code) is stored
For **books**, an isbn (international standard book number) and author name are stored.
For **toys**, the weight (in total ounces!!) is stored

The program should offer the user a menu with the following options (changes from the previous version in **bold**):

1. Add a product to the inventory (user should enter a letter (M or B or T) to indicate the **product category** and then input corresponding values).
2. Remove a product from the inventory (by sku).
3. Display the information for a product (given the sku).
4. Display the inventory in a table, **sorted by sku.**
5. **Process a sale.**
6. Quit

For #3, display **all** the attributes for the product item (this will differ for each category).

For #4, display **only** the product category (Movie, Book, or Toy), sku, price, quantity, and title for each product item.

The details of #5 Process a sale are given below.

The program should perform the operation selected by number and then re-display the menu. If the operation fails (i.e. attempt to remove a product not in the inventory) your program should display a message.

Your program should store the inventory in a file between executions of the program, so that when the program is run again it will start up with the same inventory contents as when it last terminated.

5. Process a sale

The products are offered for sale on Amazon, which charges a commission on each sold item, and refunds a shipping credit to the seller for each sold item (the seller must ship the items offline and pay their own shipping costs).

The process sale operation will (A) update the quantity of sold items in the inventory, and (B) calculate and output the sellers profit for the sold items.

To process the sale of a certain product, the user must input the following:
sku
quantity sold
shipping cost (the amount paid by the seller to ship these item(s))

Your program should update the quantity attribute for that item in the inventory, if there are enough items in the inventory. If not, issue an error message and abort the operation. Do NOT remove the product from the inventory if the quantity is 0.

The shipping credit and commission that Amazon charges for each sold item are shown below. Note that for toys, the shipping credit depends on the weight (in pounds, rounded up) of the item. So for a toy weighing 18 ounces, the shipping credit is $4.49 + (.50*2) = $5.49. (Note: 1 pound is 16 ounces)

Product Type	Shipping credit per item	Commission
Movie (dvd)	$2.98	12% of price
Book	$3.99	15% of price
Toys	$4.49 + .50/lb	15% of price

To complete the process, compute and output the following values:

Total price
Total shipping credit
Total commission
Profit

These values should reflect the total quantity of items sold. So the Total price is the price times the quantity sold. The same for the Shipping credit and Commission.

The Profit is calculated as follows:

Total price + Total shipping credit - (Total commission + Shipping cost)

(Note that Shipping cost is input by the user, and is the cost to ship ALL of the items).

NOTES:

- You may use an IDE (Eclipse, netbeans, etc) or just an editor and command line operations (javac, java) in unix or windows/dos to develop your program.
- This assignment is to be done with your partner (in groups of 2).
- Use inheritance to implement the different product types. Use polymorphic functions to compute the shipping credit and display ALL of each product's attributes. Note: a method can just return a single value.
- Use good design. Try to move the common attributes and methods into the superclass as much as possible. Use abstract classes when you can. Make the instance variables private (or protected). Avoid duplicate code in subclasses.
- Implement the Comparator in order to do the sorting (do NOT write your own sort function!)
- Use a package for your classes and put your files in the appropriate directory structure.
- Follow the style guidelines from the class website. **Use javadoc comments for all of your public elements.**

Logistics:

Please submit your files in a single zip file (assign2_xxxxxx_yyyyyy.zip). The xxxxxx and yyyyyy are your TX State NetIDs (mine is js236, you have two, one for each partner).

Submit: an electronic copy only, using the Assignments tool on the TRACS website for this class. Submit using the TRACS account of just ONE member of your partnership.