1 Objectives

1. Practice using classes
2. Practice using subroutines

2 Discussion

Recall that Java provides access modifiers for class members: public and private. This lab will demonstrate writing and using “getter” and “setter” functions, which provide an interface with private class member variables. These form what is called an API: Application Programmable Interface.

This lab involves two classes: Inventory and MainClass. The Inventory class will be implemented in a way that is portable, and can be used in any other application that might need to model inventory items, along with their name, quantity, and price. The MainClass only provides an interface with the Inventory class, and is not portable in the same sense of the Inventory class – i.e. it could not be easily adapted for another program or project.

Finally, this will be the first component of a developing series of application modules which, by the end of the semester, will form a usable inventory management system!

3 Tasks

1. Start a new Java project called “Lab9”
2. Create a class called MainClass
3. Create a class called Inventory.

(a) Add private instance (non-static) variables to the Inventory class to store data for:
   i. name - a String that describes the product.
   ii. stock - the integer quantity of items in stock
   iii. price - the value of the item
(b) Add six (!) public functions inside the Inventory class – two for each of the private members. These will be “getters” and “setters”: If the private member variable is private double price, the setter should look like public void setPrice(double p) and it will take the argument p and assign it to the price attribute inside the class. The
“getter” for \texttt{price} would look like \texttt{public double getPrice()} and simply return the value contained in \texttt{price}. This is a common programming pattern so that the person designing the class can control which variables can be changed or accessed from outside that class.

4. In your \texttt{MainClass}, create an array of five (5) \texttt{Inventory} objects. These will each have to be instantiated \textit{individually} inside the array!

5. Use a \texttt{Scanner} to ask the user to enter the data for the five separate inventory items, inputting their names, quantities, and prices. Note that you will have to use your “setter” functions to modify the object attributes.

6. Write a \texttt{static void} function in \texttt{MainClass} called \texttt{showInventory}, and which takes a single array parameter of type \texttt{Inventory[]} . The full prototype of this function should look like \texttt{public static void showInventory(Inventory[] items)}. The function should neatly display the three attribute values stored inside the passed inventory object.

7. Write a similar function to \texttt{showInventory}, but called \texttt{printInventory} that prints the inventory report to a file.

8. Call the \texttt{printInventory} and \texttt{showInventory} functions with the array of \texttt{Inventory} objects that you read in from the keyboard.

9. Try modifying one of the inventory items, via a setter function, then print it again with your \texttt{printInventory} subroutine. For example, change the name of the one of the inventory items, call the \texttt{showInventory} function, and verify that the change is reflected in the corresponding item.

4 \textbf{Turn In}

1. Upload each “java” file to the Canvas assignment, with appropriate comments.

5 \textbf{Bonus}

1. \textit{(10 pts)} Allow the user to enter an arbitrary number of items into the array. Note that to do this, you will need to have a flow (functions can make this much easier) to \texttt{resize} the array by making a new, bigger one, copying the items out of the smaller one, and then replacing the array with the bigger one.