Computer Science CSC 324S: Programming Languages

Assignment 1B, Java

Due Date: Friday, 5 February, 4pm exactly

Check the web page for where to drop off your assignment.
No grace days can be used on this assignment.

Introduction

The Gap enjoyed your efforts, but another branch of the chain managed to hire someone else to do the same job, and the decision has been made to go with that version. However, due to a poor choice of data structure, the program has some minor flaws. Naturally, you've been re-recruited to fix them.

Programming Style

Please follow the same programming style as described in the previous handout.

Part A

The same clothes descriptions are used in this assignment. Look at the starter code for the names and exact hierarchy.

You'll notice that `ClothingInventory` uses an array to store the items. As a result, there is a maximum number of items. You don't need to fix this.

But there's a small style point – `ClothingInventory.getItems()` returns a `Vector`, when it should really return an `Enumeration`; once you change this, class `Driver` no longer needs to know a thing about the `Clothes` class! (Once you've fixed this, you can remove the `import csc324.clothes.*;` statement.)

Task 1: create an `ArrayEnumeration` class that operates on arrays. The `ArrayEnumeration` constructor will need some information about the array it is operating on – it needs to know which array it is using, and the number of elements in the array. Since this is a nice freature for any array of objects, make the new class `public` and put it in the `csc324.util` package. Change the method `ClothingInventory.getItems()` to return an instance of this `Enumeration`.

Task 2: You'll need to modify the `Driver.sellClothes()` method to take this into account. Also, it may involve adding another method in `ClothingInventory` that takes an integer and sells the piece of clothing with that index.

Hand in: a printout of your new `ClothingInventory` and `Driver` classes, with your changes highlighted.

Electronically submit: your new `ClothingInventory` and `Driver` classes.

Part B

There seems to be a problem with prices. In particular, all price ranges are the same! It's clear in the code that all the clothing classes are setting `minPrice` and `maxPrice` – take a look at the static initializers. Find the source of this problem and fix it. Note that your solution may be not be stylistically satisfactory; the problem involves the difference between shadowing and overloading, and you may find that you need to declare a method as non-`static`, even though it feels like it should be `static`. Relevant methods include, but may not be limited to, `isValid()`, `getMinPrice()`, and `getMaxPrice()`.

Hand in: A typed explanation of the problem, and how you fixed it. A paragraph or two is enough; we don't want a book. Also include a diagram of an example of the problem, using the memory model discussed in class on 26 January. (To keep the diagram small, you may leave out all unrelated instance variables, methods, instances, and classes.) Also hand in printouts of all the classes that you modified to fix this, with your changes highlighted.

Electronically submit: Any classes you modified to fix this problem.