Lab 2: Queues

Copyright ©2022 – Howard Community College All rights reserved; Unauthorized duplication prohibited.

This lab will continue with our TV store software that was started in Lab 1. As such, the first thing that must be done is to fix any issues that were found in Lab 1.

In this code, you are adding the ability of customers to buy TVs and then have the store check them out with payment via the adding and removing from a queue.

There must be a javadoc at the beginning of the code with your name, the due date, a description of the program and a version number. The description is NOT the title of the lab. It needs to explain what is being done in the program. These are comments only.

Follow the sample output screen shots to determine how your program must run and then display the data. The first three lines of output must be the name of the Lab, the copyright statement, and the title of the program. This must be done via a call to a static method.

The code needs to add a new class called Customer. Customer must have the following instance variables:

1. Customer Name
2. Customer Account Number
3. The number of TVs purchased
4. The cost of the TVs purchased
5. An ArrayList that holds the id numbers of the specific TVs purchased

The Customer class must have:

1. A default constructor
2. A constructor with parameters for the name, account number, the number of TVs purchased, and the ArrayList of the TV id numbers (note: the cost will be calculated)
3. The appropriate Get and Set methods
4. A toString method that displays the receipt of the sales as displayed in the screen.
5. A method to calculate the amount due (including sales tax). Each TV costs $199.95. The sales tax is 6%. All costs are in US dollars and must display to two decimal places

In the driver program:

1. Create a Customer queue as a LinkedList.
2. Add two menu options
   1. Customer Purchase
   2. Customer Checkout
3. Customer purchase should:
   1. Determine if there are any TVs to purchase; if the inventory is empty, the code must display an appropriate error message
   2. Allow the user to enter in the customer’s name, account number and TVs purchased
      1. The code must assure that the number of TVs purchased is > 0 and less than or equal to the number of available TVs
   3. Once a valid number of TVs are purchased, the code must
      1. Create an ArrayList
      2. Remove the appropriate number of TV id numbers from the inventory stack
      3. Add the TV id number to the ArrayList
   4. Create a Customer object using the data just input
   5. Add the Customer object to the queue
   6. Display the number of TVs left in the inventory stack
4. Customer checkout should:
   1. Determine if there are any customers in the queue; if the queue is empty, the code must display the appropriate error message
   2. Remove the first element of the queue
   3. Display the receipt as shown in the screen shot below; the receipt has the customer’s name, the customer’s account number, the number of TVs purchased, the total cost, including tax, in US dollars and the list of the TV id numbers
   4. The code should then display the number of customers left that need to be checked out
5. The Exit Program must check for the existence of any customers still in the queue
   1. If any customers are still in the queue, the code must display an appropriate error message and redisplay the main menu
   2. If there are no customers left in the queue, the code must display the final inventory and a goodbye message

Additional Requirements:

1. There should be NO hard coded values in the code. All final variables should reside in the interface file.
2. The code must check for the valid input of all numeric input entries. This includes not only checking for validity but checking that an integer is entered.
3. The code must use a stack for the TVs in inventory and a queue for the customers. The code must use the Java stack and queue collections; do not use an array.

Submit your .java code files (the driver class, the TV class, the Customer class, and the interface) via Canvas. Please use a .zip file for the submissions.

Screen Shots:

The new menu items**:**

**Graphical user interface, text, application

Description automatically generated**

Inputting in a customer:

**Text

Description automatically generated**

Inputting in a second customer:

**Text

Description automatically generated**

Checking out a customer:

**Text

Description automatically generated**

Exiting the program with customers left in the queue:

**Graphical user interface, text, application

Description automatically generated**

Exiting the program with no customers left in the queue:

**Text

Description automatically generated**