**Assignment 4 – BCS 345 Java Programming**

**Due: 11/6/2019 @ 10:50am**

SUBMIT ALL PROJECTS NECESSARY TO RUN THIS PROGRAM. When you submit the assignment on Blackboard make sure your zip file is attached to the submission.

IMPORTANT – Make sure you ***properly comment*** AND ***properly indent*** your program. The commenting and indenting documents are on Blackboard. ***If you fail to properly comment or properly indent I will deduct points.***

***VERY IMPORTANT – IF THE PROGRAM DOES NOT COMPILE THERE WILL BE MAJOR POINTS TAKEN OFF.***

***Overview***

You will be writing the PurchaseCollection class. There are two main parts to this assignment.

1. Write the PurchaseCollection class.
2. Write a new console UI class.
3. Modify main so the user can choose the UI they want to use.

You will NOT need to create any new projects or packages for this assignment. You should just add to the existing business and presentation projects.

Check the “BCS 345 Eclipse Create Project And Package” document on Blackboard for instructions on how to create projects and packages and how to import them.

**Class – PurchaseCollection**

Store in project/package: <your last name>.bcs345.purchases.business

***Member Variables (all private)***

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Description** |
| Customer | Customer | The customer who is making the purchases. |
| Purchase | Purchase[] | Array of Purchase. |

***Member Method Signatures and Descriptions (all public)***

|  |  |
| --- | --- |
| **Signature** | **Description** |
| Default Constructor | Default constructor. Initializes member variables. |
| void SetCustomer(Customer c) | Sets the customer member variable. |
| Customer GetCustomer() | Gets the customer member variable. |
| Purchase GetMaxPurchase() | Searches the array for the purchase that costs the most and returns it from the method. A purchase’s cost is calculated by multiplying the product price by the quantity. It should return the instance in the array that has the highest calculated cost. It should return null if there are no purchases in the array. |
| Purchase GetByIndex(int index) | Returns the purchase at the given index.  Should use the **throw** keyword to throw an exception if the index is invalid or the array is null.  Specifically, this method should throw an ArrayIndexOutOfBounds exception if the index is not valid. No value will be returned if the index is invalid, it just throws an exception. |
| void Report(PrintStream ps) | Writes a report on the given PrintStream. Use the elements of the array as the data for the report. The report should be the exact same format as you used in assignment 1. |
| void Write(PrintStream ps) | Write the contents of the PurchaseCollection instance to the given instance of PrintStream. Use the format described at the end of this assignment. Assume the PrintStream is already open and ready to use. See file format section below for how data should be written out.  DO NOT ADD ANY DESCRIPTIVE TEXT IN THE OUTPUT. JUST PRINT THE VALUES.  IMPORTANT - Whatever data is written out should be readable by the Read method of this class. If descriptive text is added then Read will not work.  Hint: You can use the Write methods of other classes to help with this. |
| void Read(Scanner s) | Read the contents of all member variables from the given instance of Scanner. Expect data to be in the format described at the end of this assignment. Assume that the Scanner is already open.    Hint: You can use the Read method of another class to help with this. |
| void WriteJSON(PrintStream ps) | Write the member variables in JSON format to the given PrintStream. |
| void ReadJSON(FileReader fr) | Read the contents of all member variables from the given instance of FileReader as JSON. Assume the following:   1. FileReader is already open. 2. Member variable values are stored in JSON format. |
| @Override  String toString() | This method should show descriptive text and data. It can be used to display data to the user. For example:  **Jane Doe**  **10 Broadway**  **New York, NY 1001**  **2, Galaxy s10, $199.99**  **1, Samsung Bluetooth, $29.99**  **2, Slim Fit Hard Plastic Case, $2.99**  **3, Charger, $17.99** |

**Class – PurchaseCollectionConsoleUI**

Store in project/package: <your last name>.bcs345.purchases.presentation

***Member Variables (all private)***

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Description** |
| No member variables |  |  |

***Member Method Signatures and Descirptions (all public)***

|  |  |
| --- | --- |
| Signature | Description |
| ShowUI() | Shows the user interface. When this method is called it should do the following:   1. Display the menu to the user. 2. Process the user selections   There should be no display or processing code in main. See the Menu Description section below for details. |

**Class – Main (revise from previous assignment)**

Store in project/package: <your last name>.bcs345.purchases.presentation

***Member Variables (all private)***

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Description** |
| No member variables |  |  |

***Member Method Signatures and Descirptions (all public)***

|  |  |
| --- | --- |
| Signature | Description |
| public static void main(String args[]) | Update the existing main method so that the user is given a choice of which user interface to use (create a menu and display to the user).  The choices are CustomerPurchaseUI (from previous assignment) and PurchaseCollectionConsoleUI.  If the user chooses CustomerPurchaseUI then create an instance of CustomerPurchaseConsoleUI and call the ShowUI method on it.  If the user chooses PurchaseCollectionConsoleUI then create an instance of PurchaseCollectionConsoleUI and call the ShowUI method on it.  Look in the “Menu that Main Shows“ section below for the menu to display. The menu should keep showing until the user chooses to exit. |

***PurchaseCollection UI Menu Description***

This program will present a menu to the user and then perform an action depending on what the user chooses to do. You should create an instance of PurchaseCollection inside of ShowUI. When the program runs it should display the menu to the user and give them a chance to input a choice. An action should be taken depending on what choice the user makes. Here is the user menu:

**PurchaseCollection UI**

**---------------------**

**1 – Read PurchaseCollection from file**

**2 – Read PurchaseCollection from file as JSON**

**3 – Write PurchaseCollection to file**

**4 – Write PurchaseCollection to file as JSON**

**5 – Show purchase by index**

**6 – Show maximum purchase**

**7 - Show PurchaseCollection report on screen**

**8 - Show PurchaseCollection toString on screen**

**9 - Exit**

**Enter Choice:**

THE PROGRAM SHOULD KEEP SHOWING THE MENU AND PERFORMING AN ACTION UNTIL THE USER CHOOSES TO EXIT.

***Actions***

|  |  |
| --- | --- |
| Choice | Action |
| 1 | Reads in data for the instance of PurchaseCollection. This menu option expects data to come in according to the PurchaseCollection file format specified at the end of the assignment. The user should be prompted to enter a filename to read from.  Hint: You can use a function on PurchaseCollection to help out with this. |
| 2 | Similar to option 1 except the input file must be in JSON format. |
| 3 | Writes all data from the PurchaseCollection instance to a file. The user should be prompted to enter a filename to write the data to. Data should be written out according to the PurchaseCollection file format specified at the end of this assignment.  Hint: You can use a method on PurchaseCollection to help out with this. |
| 4 | Similar to option 3 except the output file must be in JSON format. |
| 5 | Show purchase by index. The user should be prompted to enter an index. All data for the purchase at that index should be displayed on the screen. If the index is invalid display a message to the user giving that information.  This menu option should contain a try/catch block for an ArrayIndexOutOfBounds exception. If the user enters an invalid index an ArrayIndexOutOfBounds method should be thrown by GetByIndex and that exception should be caught here.  Hint: You will need to use the GetByIndex method of PurchaseCollection to code this menu option. |
| 6 | Show maximum purchase.  Hint: You can use a method on PurchaseCollection to help out with this. |
| 7 | Show the PurchaseCollection report on the screen. This report should have the same format as the report from assignment 1 (see below for sample report). |
| 8 | Show what PurchaseCollection toString returns on screen. |
| 9 | Exit |

***Menu That Main Shows***

**Choose UI**

**---------**

**1 – CustomerPurchaseConsoleUI**

**2 – PurchaseCollectionConsoleUI**

**3 - Exit**

**Enter Choice:**

***Deliverables***

***One*** winzip file containing all Eclipse projects necessary to run the program.

***PurchaseCollection File Format (data on separate lines)***

First

Last

Number

Street

City

State

Zip

PurchaseCount

Description

Price

Quantity

…

Description

Price

Quantity

***Sample PurchaseCollection File Data***

**Jane**

**Doe**

**10**

**Broadway**

**New York**

**NY**

**10001**

**4**

**Galaxy S10**

**199.99**

**2**

**Samsung Bluetooth**

**29.99**

**1**

**Slim Fit Hard Plastic Case**

**2.99**

**2**

**Charger**

**17.99**

**3**

***Sample PurchaseCollection Report***

**Purchase Report**

**---------------**

**Jane Doe**

**10 Broadway**

**New York, NY 10001**

**Description Price Quantity Cost**

**----------- ----- -------- ----**

**Galaxy S10 199.99 2 399.98**

**Samsung Bluetooth 29.99 1 29.99**

**Slim Fit Hard Plastic Case 2.99 2 5.98**

**Charger 17.99 3 53.97**

**----------- ----- -------- ----**

**Total 8 489.92**

***Sample JSON***

**{**

**"customer" :**

**{**

**"first" : "Jane",**

**"last" : "Doe",**

**"address" :**

**{**

**Customer**

**"number" : "10",**

**"street" : "Broadway",**

**"city" : "New York",**

**"state" : "NY",**

**"zip" : "10001"**

**}**

**},**

**Purchase Array**

**"purchases" :**

**[**

**{**

**"product" :**

**{**

**"description" : "Galaxy s10",**

**"price" : 199.99**

**},**

**"quantity" : 2**

**},**

**{**

**"product" :**

**{**

**"description" : "Samsung Bluetooth",**

**"price" : 29.99**

**},**

**"quantity" : 1**

**},**

**{**

**"product" :**

**{**

**"description" : "Slim Fit Hard Plastic Case",**

**"price" : 2.99**

**},**

**"quantity" : 2**

**},**

**{**

**"product" :**

**{**

**"description" : "Charger",**

**"price" : 17.99**

**},**

**"quantity" : 3**

**}**

**]**

**}**