**CMSC203 Assignment 6 Implementation**

Class: CMSC203 CRN 36406

 Program: Assignment #6

Instructor: Professor Eivazi

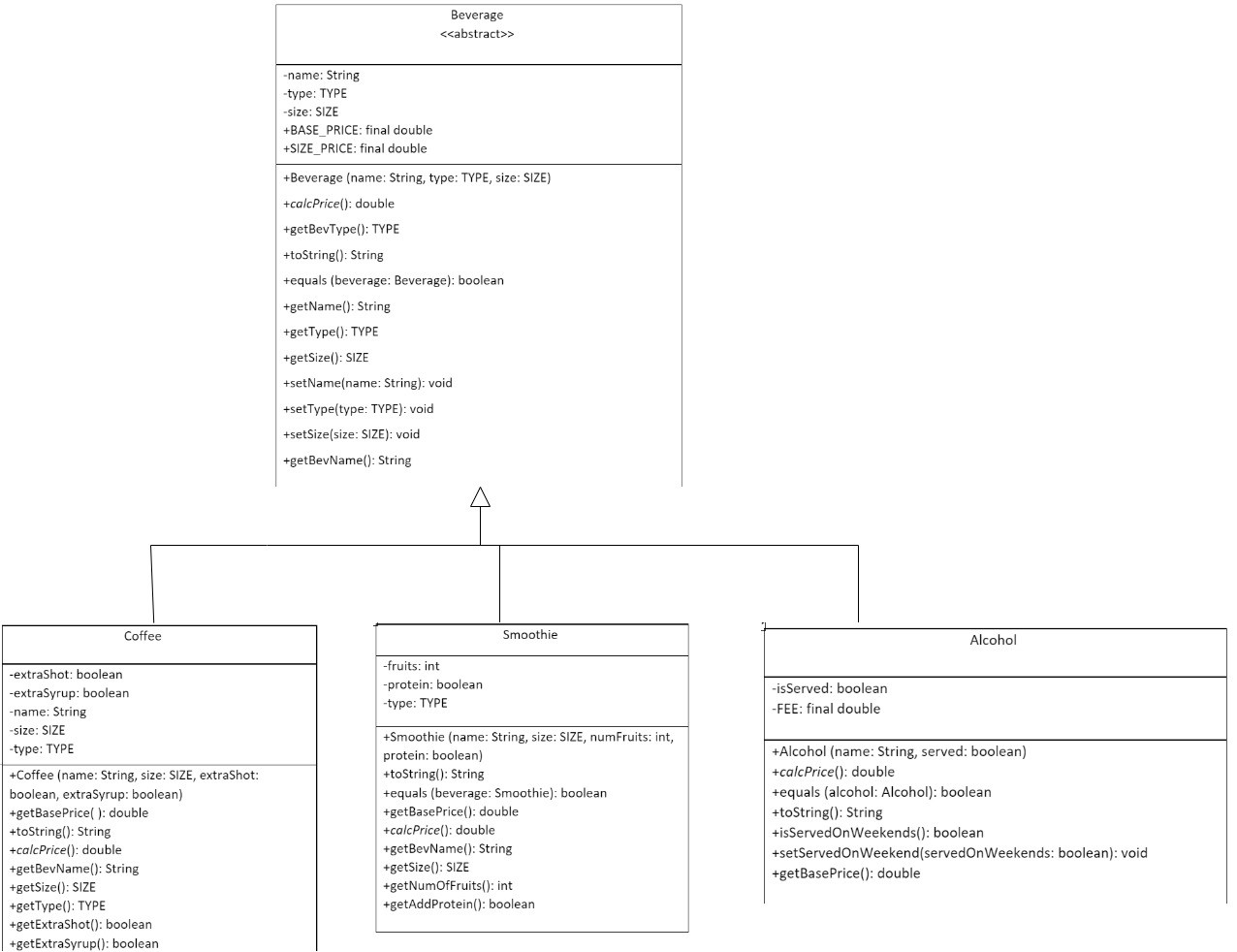
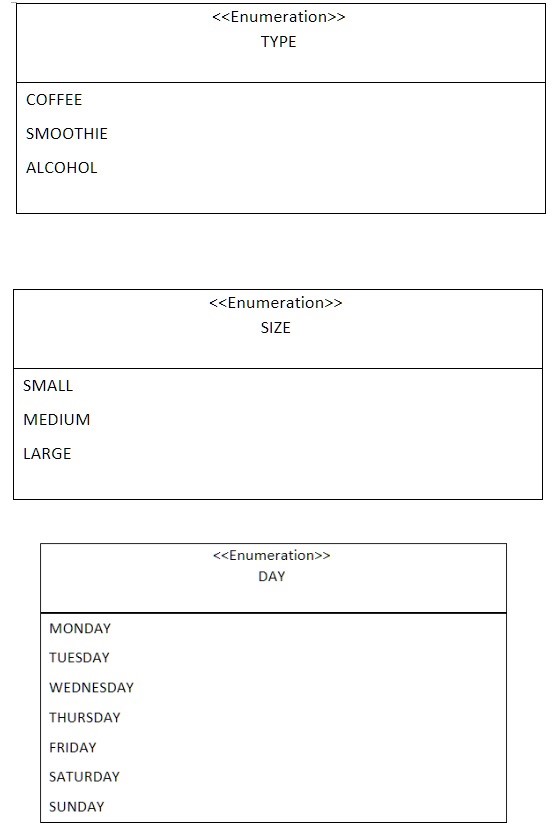
 Summary of Description: The BevShop offers 3 types of beverages: Coffee, Alcoholic and Smoothie. Beverages can be ordered in 3 different sizes: Small, medium, and large. All the beverage types have a base price. In addition, there are additional charges depending on the size and specific add-ons for each type of beverage. The BevShop has the following functionality: create and process orders of different types of beverages, provide information on all the orders, total amount on a specific order, monthly total number of orders, and monthly sale report.

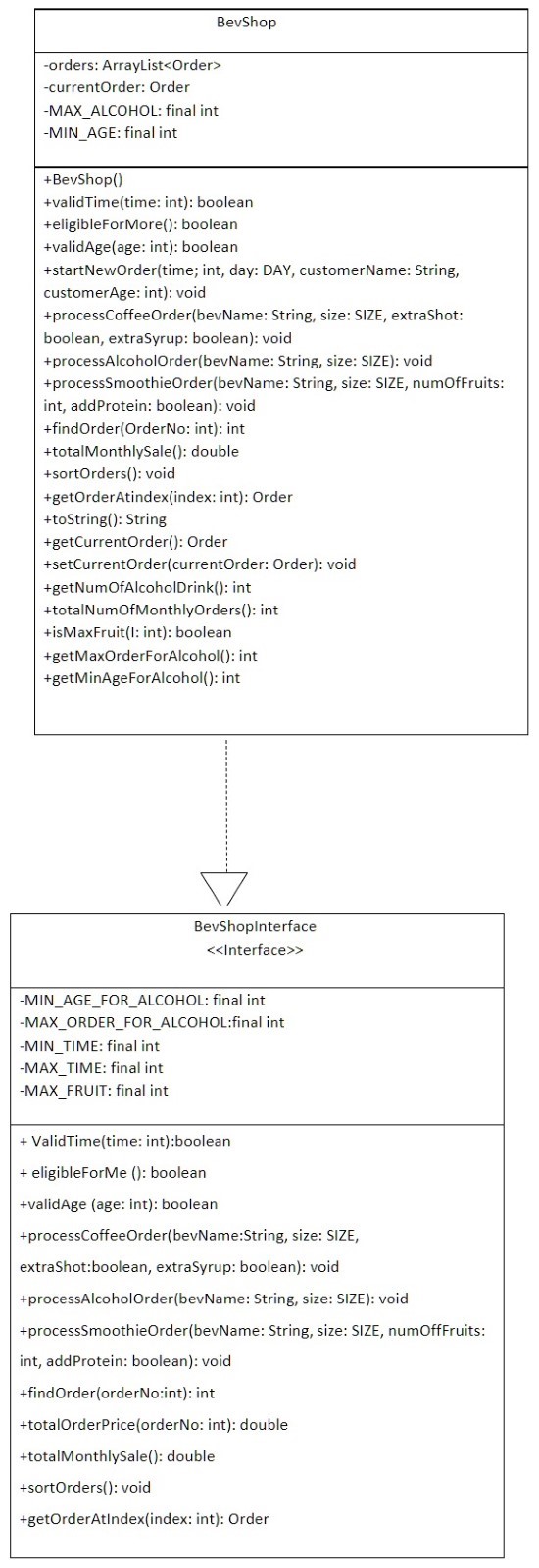
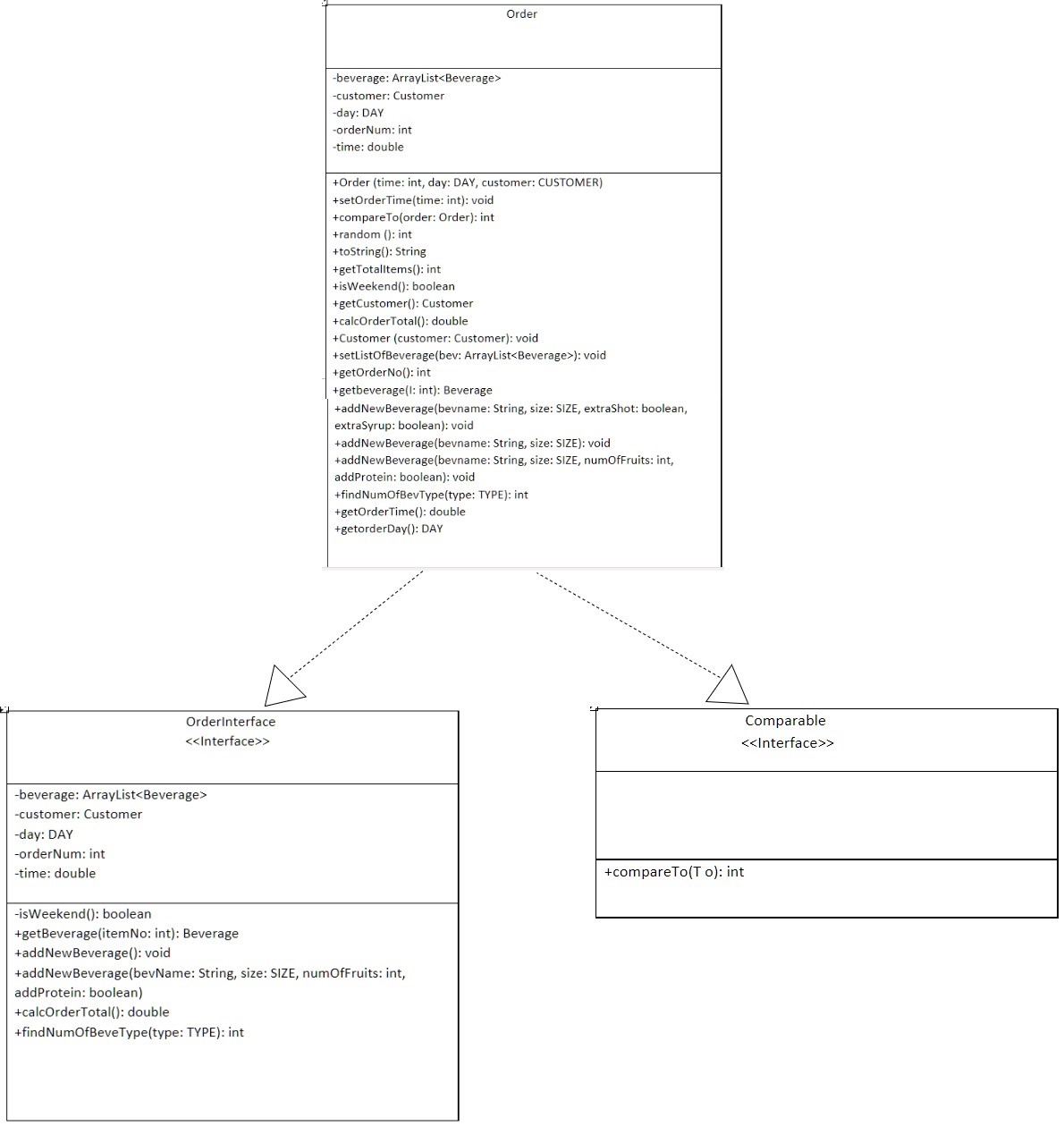
Due Date: 5/04/2022

 Integrity Pledge: I pledge that I have completed the programming assignment independently.

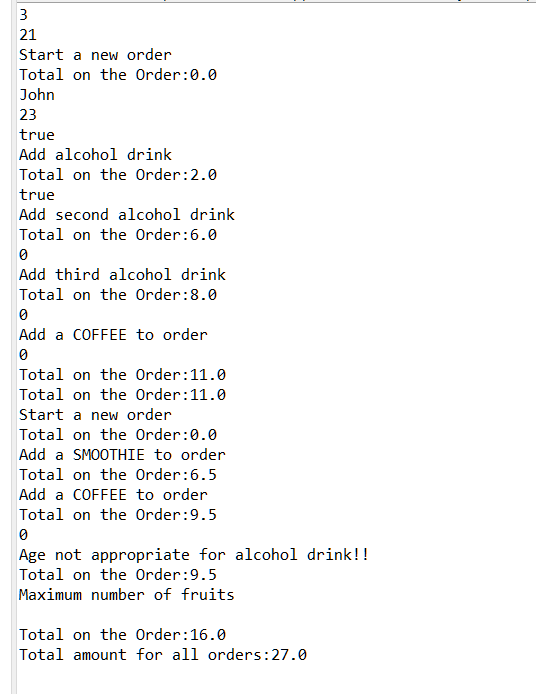
 I have not copied the code from a student or any source.

UML Diagrams:

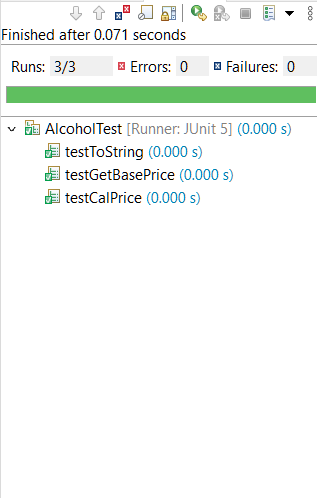
  


Screenshot of BevShopNoGUITest:

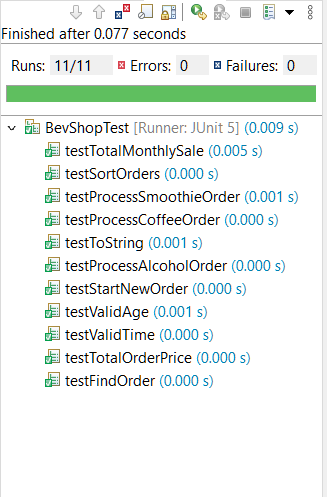


**JUNIT**:

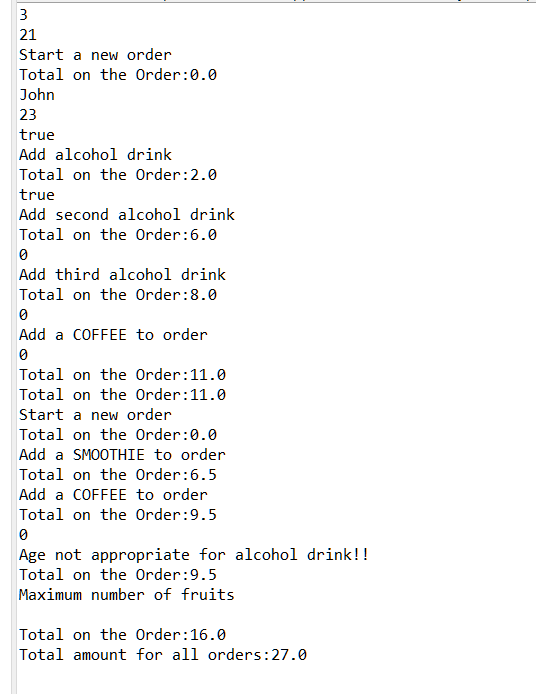
* + **Alcohol Test:**



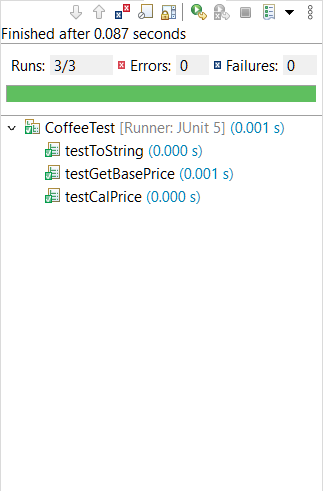
* + **BevShop Test:**



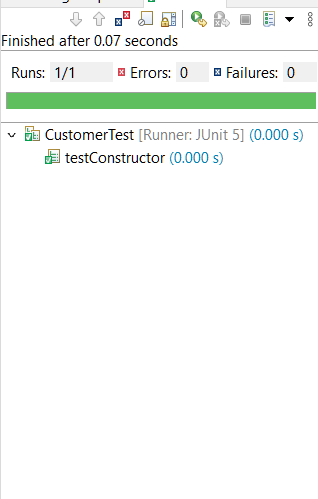
* + **BevShopNoGUITest:**



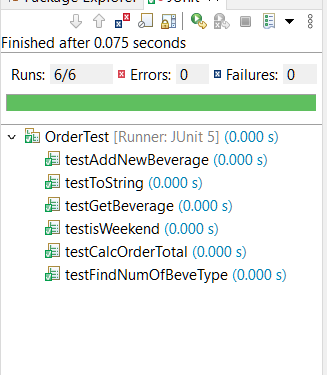
* + **Coffee Test:**



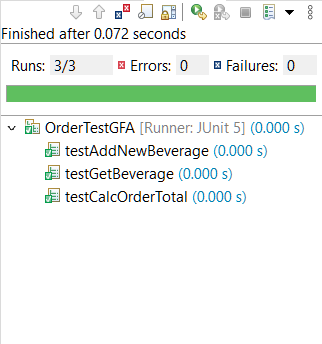
* + **Customer Test:**



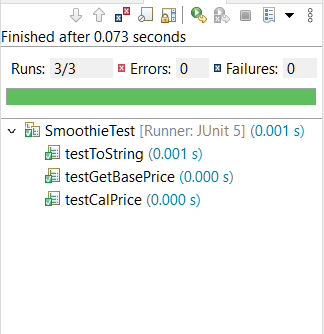
* + **Order Test:**



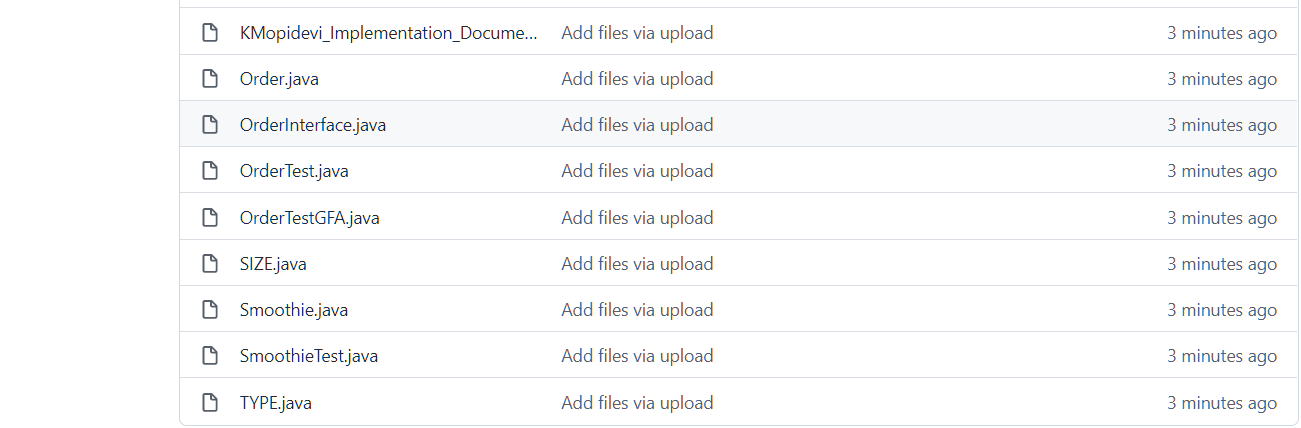
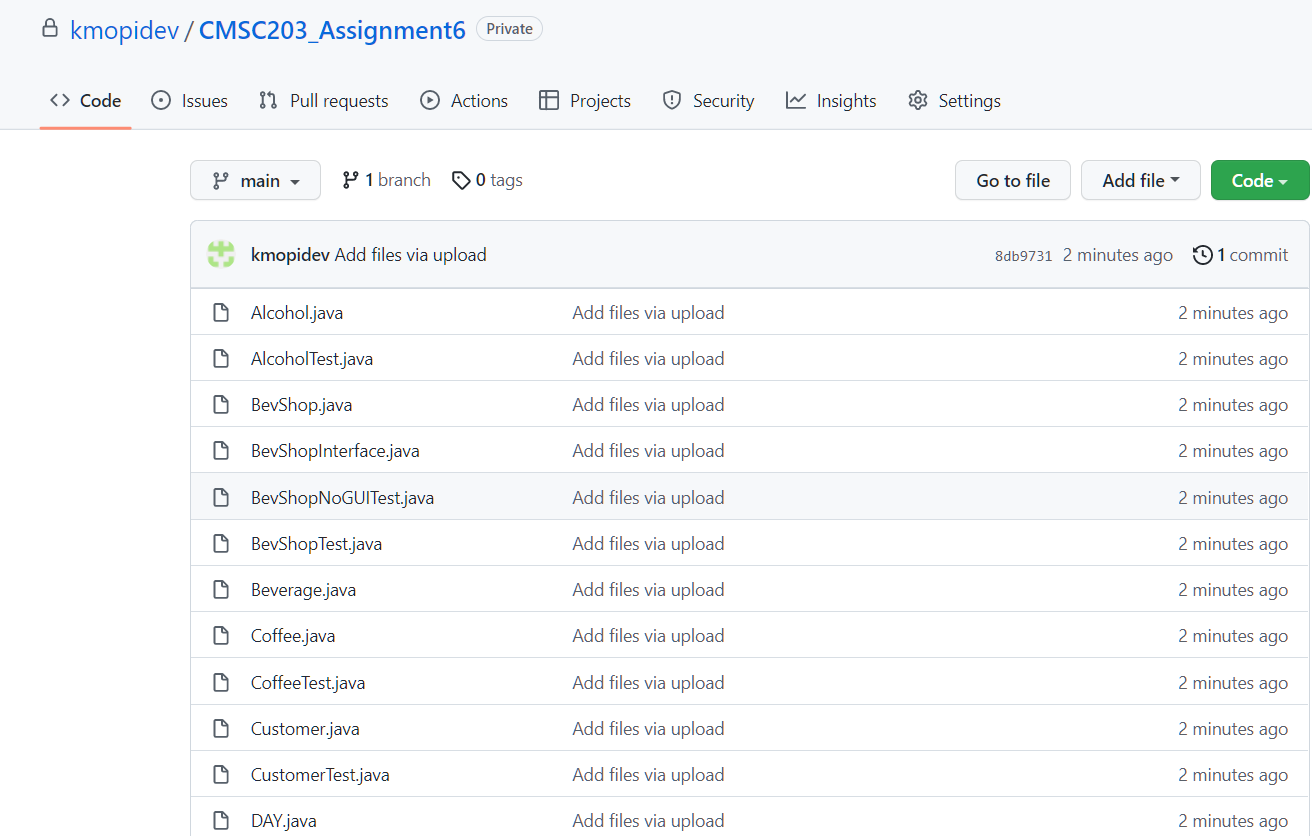
* + **OrderTestGFA**:



* + **Smoothie Test:**



GitHub:



**Lesson Learned:**

Write about your Learning Experience, highlighting your lessons learned and learning experience from working on this project.

=>What have you learned?

I learned how to calculate the price of all the beverages including the weekend extra price for alcohol, coffee, and smoothie using the super keyword to get the information from the beverage class. Created enumerated classes in the project TYPE, SIZE, and DAY. I learned how the interfaces work where it is only an abstract class used to group related methods with empty bodies and to access the interface methods by another class, I had to use implements keyword instead of extends. I learned how to call the toString method from one class to another using the super keyword super.toString(). I had to refer back to chapter 10 Inheritance to review the concepts of how super and subclasses work, inheritance, and polymorphism which helped me a lot in completing the assignment successfully. I also had a problem with the JUnit, but I added the library in the modulePath not the classpath, so I had to fix that.

=>What did you struggle with?

This assignment was a bit very challenging because it had many classes, few were extending the superclass, and few were implementing interfaces. I got very confused in mainly concepts of polymorphism, searching an arrayList, and selection sort. This assignment also had too much calculation calling super classes and other classes. I struggled with calculations. I made a few logical errors. So, I went to the MC tutoring center for help where they explained my mistakes to me and guided me to fix the code and make it run. This helped me to fix the logical errors in a few calculations.

=>What would you do differently on your next project?

I would start the project much earlier than the due date and complete the implementation document also. Though I wrote the code earlier it is taking me more time to complete the implementation document where I am feeling more stressed. Hence, I will try to do the above steps for the next project, so I do not feel anxious and have more time to check back to see if I completed everything on the checklist and have time to make any changes if needed. Also, I will have the time to fix the code if there are any errors before I submit the assignment.

=>What parts of this assignment were you successful with, and what parts (if any) were you not successful with?

I was successful in importing Assignment 6 java files to the eclipse project named Assignment6. I was also successful in adding the JUnit library to my Assignmet6 project. Then I wrote the code correctly with the help of the suggestions from the tutoring center. I struggled in calculations like sortOrder and other calculations. I wrote some code with the help of the textbook but there were few errors, and it was not running well, so with the guidance of the MC tutors now it is running well.

=>Provide any additional resources/links/videos you used to while working on this assignment/project.

I wrote the code myself by referring back to chapter 7 for ArrayList and selection sort algorithm. And referred to chapter 10 to review the concepts of inheritance, polymorphism, abstract classes and methods, and interfaces. After reviewing those concepts, I started to work on writing the code, but there were few logical errors in the calculations because they did not pass the JUnit test. So, I went to the MC tutoring center to help me understand why there were logical errors. I was making mistake in calculation using selection sort and the calculation using arrayList. Hence, they helped me understand the concept better and I was able to fix the problem in the code.

Assignment 6 Check List (include Yes/No or N/A for each item)

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N or N/A** | **Comments** |
|  | **Assignment files:** |  |  |
|  | * FirstInitialLastName\_ Assignment6\_Moss.zip | **YES** |  |
|  | * FirstInitialLastName\_Assignment6\_Complete.zip | **YES** |  |
|  | **Program compiles** | **YES** |  |
|  | **Program runs with desired outputs related to a Test Plan** | **YES** |  |
|  | **Documentation file:** |  |  |
|  | * Comprehensive Test Plan | **YES** |  |
|  | * Screenshots for each Junit Test | **YES** |  |
|  | * Screenshots for each Test case listed in the Test Plan | **N/A** |  |
|  | * Screenshots of your java file BevShopNoGUITest run | **YES** |  |
|  | * Screenshots of your GitHub account with submitted Assignment# (if required) | **YES** |  |
|  | * UML Diagram | **YES** |  |
|  | * Lessons Learned | **YES** |  |
|  | * Checklist is completed and included in the Documentation | **YES** |  |