Programming Project 3

Jade Pearl

CMSC 215 6382 Intermediate Programming

Professor Romerl Elizes

**Approach**

As I moved through the project, I made sure to:

1. Reread chapters 14-16 to better understand the uses/functionalities of JavaFX and GUI. Watch the videos in the chapters as well as doing the interactive coding examples to get used to implementations in JavaFX.
2. Figured out how to use JavaFX in VSCode. Instead of using no build tools, I had to use Maven’s JavaFX build tools and use a package in the lines of my code to make it work.
3. I programmed the TripCost class first so that I knew how the calculations worked before I moved on to all the UI implementations in the Project3 class.
4. I first programmed the UI properties (i.e. alignment and position of fields) and them added them to the scene to see how the layout looked on the screen when I ran the program.
5. After completing the layout, I then had to implement the functionality of the button by establishing the handler and calling the calculation method totalCost, that I created in the TripCost class.
6. Troubleshooting and test cases for input/output accuracy

**UML Diagram**

Below is the UML Diagram I created for this project:

A screenshot of a computer

Description automatically generated

**Test Plans**

Below are each of the 5 test cases I carried out to make sure that the project worked properly:

**Test 1: No input at all**

**A screenshot of a computer

Description automatically generated**

Result: When I clicked the calculate button, nothing happened. No NaN or such result. It ensures that the user enters at least something because leaving it empty becomes garbage in the text fields.

**Test 2: Entering all zeros**

Input:

A screenshot of a computer

Description automatically generated

Result: The total cost text field has a $NaN result because in the calculation of the total cost, the total gas cost is calculated from distance/mileage times the price of gas per gallon. Since all zeros were entered, the calculator is picking up a division of zero which is not possible and throws errors.

**Test 3: Non-number input**

Input:

A screenshot of a computer

Description automatically generated

Result: No result. This shows that the user must input strings of characters that can be interpreted as an int or double (1.1 or 1, etc)

**Test 4: Sample input from project instructions**

Input:

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Result: The expected output of $755.42 was calculated, therefore I knew that the calculation method I wrote called totalCost in the TripCost class worked properly. I also tried using the drop-down menus/combo boxes and switched from miles and gallons to kilometers and liters. This shows that regardless if the user chooses to enter miles/gallons or kilometers/liters (or both), they will always get that same result and only the numbers input in the text fields matter.

**Test 5: Random negative number**

Input:

A screenshot of a computer

Description automatically generated

Result: The calculation output as $0.00 when I input a “-“ in front of the 1000, indicating a negative number. This was to ensure that the user only entered positive numbers or zeros. In the totalCost method, I made sure to only calculate everything if the numbers input were greater than or equal to zero. If not, I returned zero to the calling method.

**Lessons Learned**

Doing JavaFX and GUI for the first time was certainly hard and time consuming so I knew that I needed to start early. The lessons I learned while doing this project were:

1. How to properly implement JavaFX and GUI applications in VSCode
2. Working with different kinds of panes that will appear on the screen when running the program, GridPanes extensively in this case.
3. How to apply written input in text fields and then use those values further in the program
4. The use and importance of handlers when implementing functionality such as with buttons. Using handlers to call methods and have the clicking of the button perform the desired action.
5. Proper implementation of text fields and labels when creating a GUI/JavaFX application