Lab #6

Oriana del Mar Quevedo Araujo – 216685869

Lassonde School of Engineering, York University

EECS 3311: Software Design

Section B

Alvine Boaye Belle, Ph.D.

Introduction

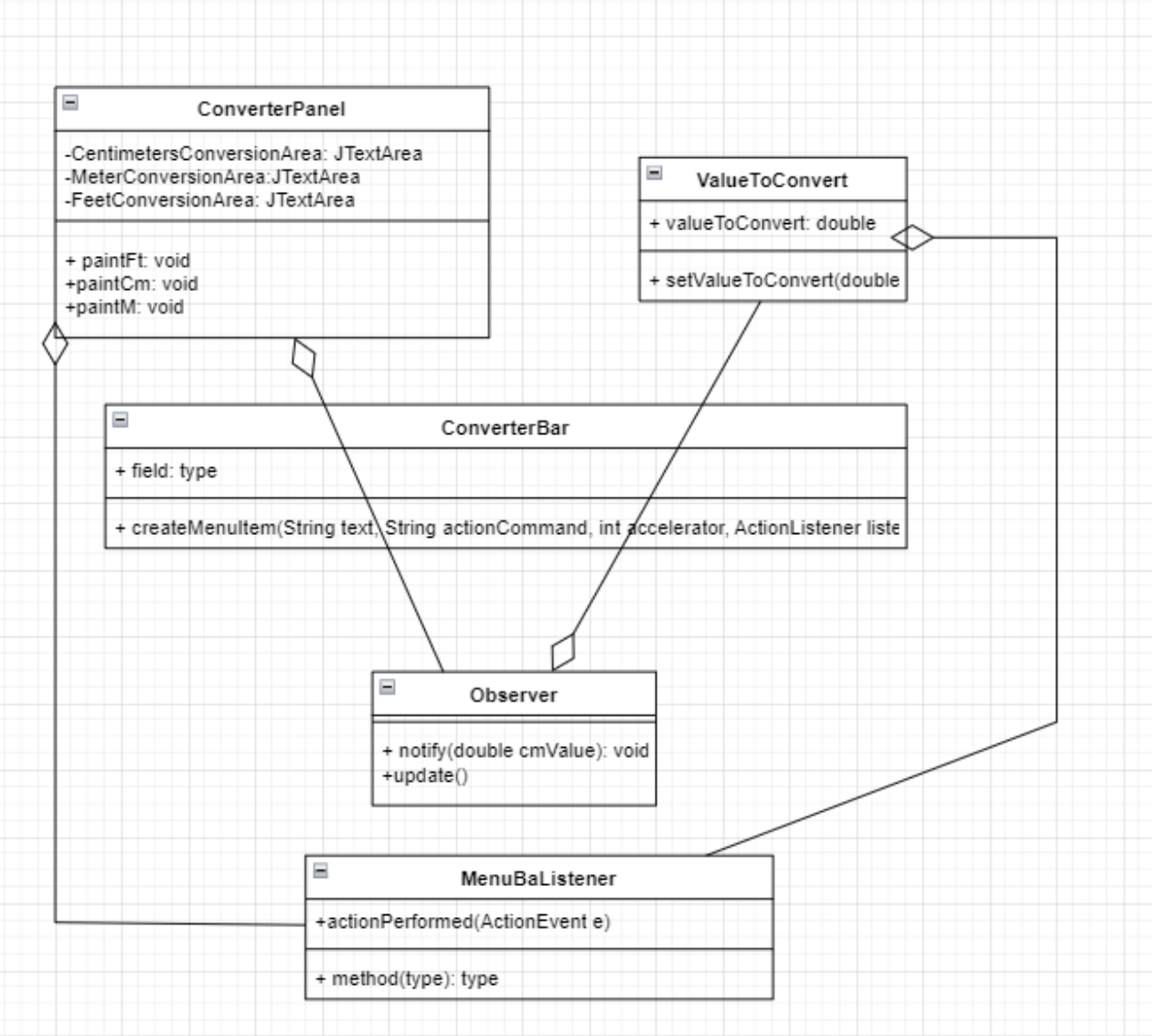
The software project is about a converter that can convert cm to ft and vice-versa. The goal is to learn how to implement MVC design pattern to construct this software in a proper manner so that it can be extensible. Also, it will allow us to learn about the observer pattern to understand how to notify other controllers that a field has been updated.

The challenges associated with this project are the implementation of such design patterns and understanding how to implement the observer and communicate between different classes.

To carry out this project we must use design patterns to implement our code and make it clean. Also, we must use composition to compose a class of an observer for example. MVC is mainly going to be used to implement the behavior of the software.

For the rest of this report, I will be presenting a design of the software through UML diagrams and provide a description about each class. Furthermore, I'll explain how each design pattern was used and finally give a conclusion on what was done and the challenges that we faced.

Design of the Solution



The software uses MVC design pattern to split the functionalities and responsibilities of classes. Also, this pattern makes it easier to decide where to implement each component of the software.

Implementation

For the model we have ValueToConvert which holds a variable to save the Centimeters input from the client. For the view we have 2 classes ConverterBar which is the toolbar where the user clicks to update the model, and which is aggregated with a listener.

The second view is ConverterPanel which extends JPanel that draws the text areas where the user inputs a value and checks the converter value.

For the Controller we have MenuBarListener which is aggregated with the ConverterBar. When the user clicks on the button the controller will save the value of the user in the model. When the value is saved in the model the observer pattern which is implemented in the observer class will observe that and notify the meter and feet conversion text areas and update the fields.

Many OO designs have been used here such as aggregation and design patterns that help us structure properly our software.

The MVC pattern helped us divide our software and Observer pattern makes it easier to update.

Conclusion

What went well in our software is the implantation of MVC which made it easier for later work when assigning tasks. But it was hard to understand and implement the Command pattern. For the Observer it was quiet straightforward to understand how to notify and update fields.

Due to personal circumstances, the lab had to be completed by myself. Thus, I cannot comment on the advantages and drawbacks of completing it in group. I was unable to work with my team members and they worked on their own code themselves.