SE 450: OO Software Development

Assignment 4: Design Patterns (2)

Problems

Instructor: Vahid Alizadeh

Email: v.alizadeh@depaul.edu

Quarter: Fall 2022

DEPAUL UNIVERSITY
COLLEGE OF COMPUTING AND DIGITAL MEDIA

Last update: October 30, 2022

Assignment 4: Design Patterns (2)

Submission

- **Due date:** Please check the slides and announcements
- **How to submit:**
  - Prepare your report in a PDF file for all exercises which includes your answers, explanations, any diagrams, or screenshots showing different executions of your code.
  - Please provide your name and exercise info as a comment in the header of each source code.
  - Place source codes inside sub-folders related to each question (ex. Q1, Q2,...).
  - Compress the PDF document and all sub-folders into a single file.
  - No screenshot, hand-written, hand-drawn, or diagram created in a photo editor are accepted and result in losing the credit. You must create your diagrams with a UML design tool.
  - Submit on D2L (Submissions - `Assignment 4`).

Goals

- Problem Solving and Architecture Design using Software Design Patterns.
- Using Chain of responsibility and Decorator design patterns to solve a real-world problem.

Deliverable for EACH of the following 2 exercises:

- **The UML class diagrams in your report document.**
- **Screenshots showing different execution scenarios in your report document.**
- **The implementation codes.**
  - You must use the sample GUI code in the course GitHub repository and complete it. The GUI code is based on these libraries : `javax.swing` and `java.awt`
  - The code must have graphical user interface to test the functionality.
  - There are many hints commented in the code.
  - Your code must be executed without any error, problem, or warning.
  - What execution scenarios should I take screenshot?
    - The results of input value=1 for each of the three possible units (FOR EACH EXERCISE).
    - The results of invalid input (string, negative, and null) (ONE SET IS ENOUGH FOR BOTH EXERCISES).

Assignments

Exercise 1. (50 pts) Chain of Responsibility Design Pattern

This problem combines the decorator pattern with the chain of responsibility pattern(CoR) on a Length converter program (LCP) with a GUI . The LCP performs conversion from Kilometer to one of the following three units: Mile, Yard, and Foot.

The input string specifies the amount to be converted and dropdown menu indicates which unit it will convert to. The CoR pattern will be applied to the processing of the input string to generate a number representing the converted amount. The LCP user interface is seen as a client making a request to convert the input to a given unit. Three handlers are available, one for each unit Mile, Yard, and Foot.

The final result should be similar to Figure 1:

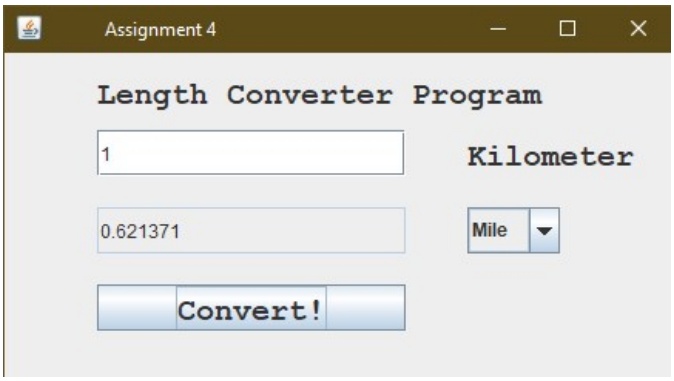

Figure 1: Demo of exercise 1

The resulting processing flow for LCP is shown schematically in Figure 2:

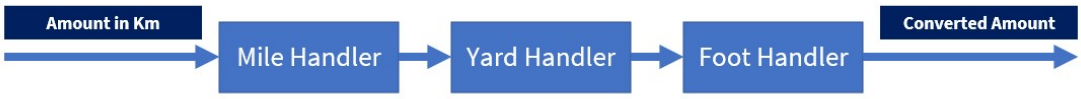

Figure 2: Exercise 1 CoR Schematic

Exercise 1. (50 pts) Decorator Design Pattern

Taking into account the previous exercise, we want to format The text appearing in the output field of the LCP GUI. This text is a string that has to undergo three decorations:

- Round output to 2nd decimal (e.g., 218.723 to 218.72)
- Write output in exp. notation (e.g., 218.72 to 2.1872e2)
- Add the unit name to the converted amount (e.g., 2.1872e2 to 2.1872e2 Yards)

The final result should be similar to Figure 3:

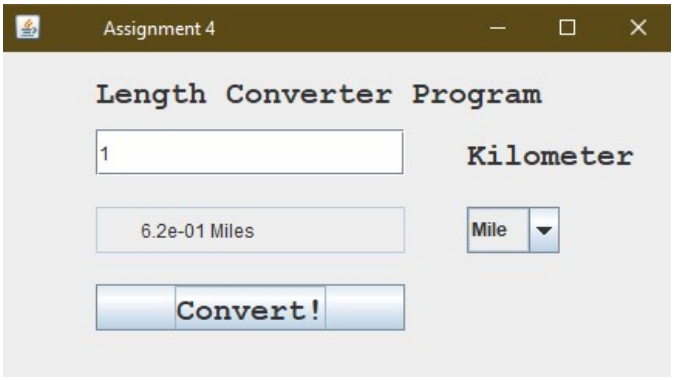

Figure 3: Demo of exercise 2

The resulting processing flow (including the chain of responsibility) for LCP is shown schematically in Figure 4:

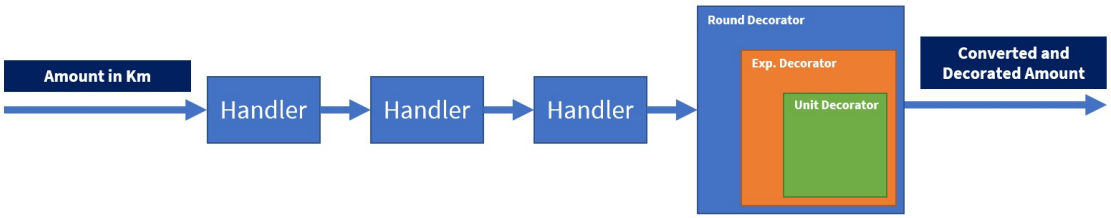

Figure 4: Exercise 2 CoR + Decorator Schematic

- If you have any questions, review the lecture OR ask for help on the course forum.
- Enjoy OO programming!