SOEN 6011 - Software Engineering Processes

Problem 5

Nigel Rodrigues

Github link - `https://github.com/nigelrodrigues07/SOEN_6011_Nigel`

Github link for Problem 5: `https://github.com/amitsachdeva45/SEPPersonal`

August 4, 2019

0.1 PROBLEM 5 : CODE REVIEW FOR FUNCTION F5(Gamma)

Peer code reviews are all about collaboration and not competition [1]. Following are my observations for reviewing the code created by my team member for the Gamma function:

0.1.1 CODING STANDARDS

The check for coding standards was performed manually. The coding standards that were decided in the previous deliverables were followed in the implementation. The use of camel case, making essential methods public or private, proper use of comments, proper functionality to avoid God classes and understandable variable names were implemented correctly.

0.1.2 CODING STYLE

This check was done automatically through the use of Checkstyle. There were no violations and the code structure was clean and readable.

0.1.3 QUALITY ATTRIBUTES

By manually going through the code, the quality attributes of readability and maintainability were observed. The Checkstyle implementation and the use of design pattern enabled this. The run-time of the overall code to process the solution was close to three seconds.

0.1.4 FUNCTIONALITY

This was performed by manually checking the functional aspects of the code. The logical flow of the code was understandable. Various cases were tested out and checked for consistency via the GUI. The tests displayed the correct solution which were verified by a gamma calculator.

0.1.5 USE OF DESIGN PATTERN

This was manually checked. The design pattern used was of the type Model-View-Controller. This design pattern fit well with the GUI used to display the function and make the user experience simple and effective as shown in the figure below. The GUI also kept a track of the user's previous operations making it convenient to be readily available if it had to be used in the future.

There was a message to tell the user the exact input structure i.e. $a + ib$ required to calculate the gamma function. This gives me an understanding that a lot of thought was given to usability and effective functionality.

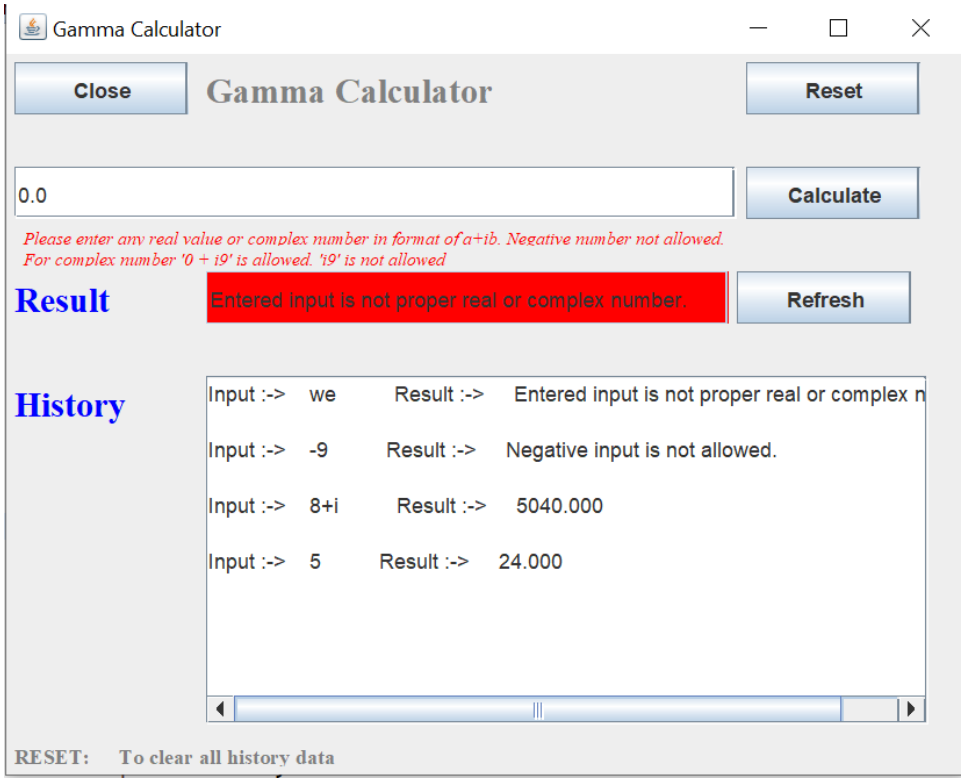

Figure 1: GUI for gamma function

0.1.6 SUGGESTED IMPROVEMENTS

- MORE ARITHMETIC OPERATIONS: The functionality that a calculator provides i.e. to add, subtract, multiply or divide could be incorporated in the current implementation to cover all aspects of the operation calculations.
- DECOMPOSITION OF METHODS: For better understanding, the number of lines in a method could be reduced to 20. Major functions could be further broken down.

Bibliography

[1] 9 Code Review Best Practices + How to Do Code Reviews — Perforce. [online] Available at: https://www.perforce.com/blog/qac/9-best-practices-code-reviews