Software Engineering Process **Amit Sachdeva**
Topic: Deliverable 2 **40084627**
Prof. P. Kamthan Due Date: 29/July/2019
`https://github.com/amitsachdeva45/SEPPersonal/tree/master/Documentation/Deliverable2/Problem4`

Gamma Function

1 Changes from Delivarable 1

- **Changes in Constraints** :: D1: Constraint: It will through **Infinity** on value > 170.
  :: D2: Constraint: It will through **Infinity** on value > 109
- **New constraint for complex number D2** :: If user input $0 + i9$, it will return proper value as it is in format of $a + ib$, but when user input **i9** it will through wrong input error.

Problem 4: Quality Tools : CheckStyle

2 Check Style

2.1 Definition

Checkstyle is a code analysis tool used in software development to check if code is following proper rules and regulations set by java. Checkstyle is implemented in terms of modules of checks.

It includes many things like checking proper indentation, use of proper java doc, remove useless spaces and many more. It enhances the way of writing code. It can be installed in eclipse as a plugin.

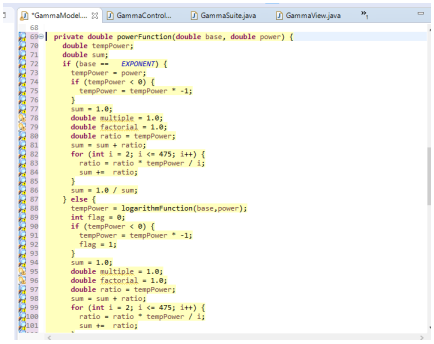

Figure 1: Checkstyle detecting code quality issue in my code

2.2 Advantages

- It checks proper documentation of public methods and variables (Java Doc).
- It removes the useless indentation.
- It makes code readable to another developer
- It removes the extra use of packages which decrease the load on a compiler.

2.3 Disadvantages

- Code quality not only related to indentation but it also includes refactoring which is not done in checkstyle
- It doesn't check the errors inside the code.
- It doesn't check the complexity of a program.
- It doesn't check the length of method which in coding world should not be greater than 25 lines.

Problem 4: Debugger

3 Eclipse

3.1 Description

Eclipse is java debugger which is used for mostly used for large projects in the industries. I have used eclipse debugger for my project. It helps me providing proper space for my project management with proper division of my design pattern implementation and test cases.

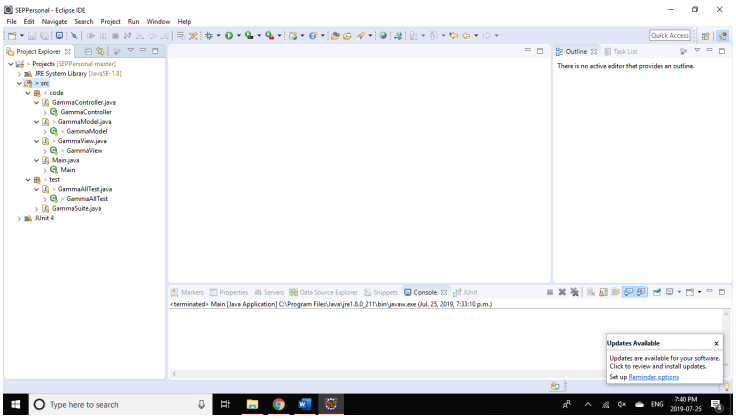

Figure 2: Proper division of my classes and test cases in eclipse debugger

3.2 Advantages

- It help in removing useless variables.
- It provides my inbuilt plugins like check style to improve code quality.
- It debugs the code and errors in my syntax
- It provide separate compilation of test cases and my project.
- It helps in generating Javadoc.
- It provides step wise execution of project by adding break points which help in tracking bugs inside the code.

3.3 Disadvantages

- Every plugin has so much dependencies which increase the size of project.
- It does not fix bad code, designs.
- Due to so much dependency during launching, it affect the speed of working.

Problem 4: Explicit Efforts

4 Correctness and efficient

Gamma Function is basically a factorial function of real values. I have used

$$\Gamma(x) = \int_0^\infty s^{x-1} e^{-s} ds \ \forall \ Re(x) > 0$$

I have implemented it using the core method of integration using graph addition by dividing my function's graph into small chunks and use formula of trapezium to add all chunks. It leads to good accuracy as tested and compared with proved values. Explicit efforts is made to judge the infinity value in reality, and maintain boundaries conditions which can even lead to null values. I have also kept in mind the small decimal values which can range from 0 to nearly $1E-2$ which is so small but are so important for small gamma value results. Apart from this, I have not used any inbuilt math function rather created my own by using **taylor series** which also plus point. So, every boundary condition and using core method of integration lead to high accuracy.

5 Maintainable

Every project need proper maintenance at regular interval of time. So, I have used a MVC design patterns which divide my code in different classes. So I have reduced the dependency of each other class, and any developer want to make changes in functionality of the function, then he/she should focus only on model class. Similarly, If anyone wants to change the view, then he/she should focus only view class. Along with that, proper coding standards is used which reduced the complexity of functions inside the class and proper division of work between the functions. No function is greater than 25 lines of code as defined standards. So, by implementation of design pattern, and proper use of coding standards, maintenance efforts have reduced.

6 Robust

In this project, proper handling of errors inside the models is done. For example if user enters negative value, it returns **negative input error** exception on UI but functionality of code does not hinder.

$\forall \quad value < 0 -> ERROR = "NEGATIVE\ INPUT\ ERROR"$

$\forall \quad value > 109 -> ERROR = "INFINITY"$

$\forall \quad value\ LIKE\ "any\ random\ string" -> ERROR = "WRONG\ INPUT\ ERROR"$

I have used proper history management where user can check what inputs he/she had entered earlier, and what are results which enhances user interaction. So proper checking of errors and using proper exception handling where required so that user interaction does not interrupt.

7 Usability

This project is used by scientist and researchers who indirectly or directly using this function. So, I have created proper **User Interface UI** for them as they don't know how to use command line. I have added functionality like history to check their input history. I have also kept proper error handling and informing user on UI.This lead to increase the usability of any user.

8 References

- https://www.quora.com/What-are-the-advantages-of-Eclipse-IDE
- https://checkstyle.org/eclipse-cs/
- https://www.eclipse.org/forums/index.php/t/206312/