Kunal Sharma
40053616

Deliverable 3

SOEN 6011
August 2, 2019

1 Problem-5

Code review for Function-6: ab^x
I have done the source code review for function-6 on the basis of some standard practices and guidelines which are as follows:

- **Naming Convention:** It is based on following conventions:
  - Camel case for functions and variable(starting from small letter)
  - Block level scope variable name should be in small letters
  - Class name should begin with a capital letter
  - Global constants(if any) should be in capital letters.
- **Readability:** It is checked based on following parameters:
  - Commenting and Documentation
  - Consistent indentation
  - Avoid obvious comments
  - Code grouping
  - Consistent Naming
  - Don't repeat yourself (DRY) principle
  - Limit line length
- **Commenting** It is done keeping following constraints in mind:
  - Much enough for a novice to understand what functions do.
  - Should not be over-commented
- **Architecture:** It is checked based on following constraints:
  - What architecture is chosen?
  - Is the chosen architecture followed?
- **Extendability:** It is checked based on following factors:
  - Is the code extendable?
  - How hard it would be to extend the functionality in future.
- **Code Quality:** It is checked keeping following factors in mind:
  - Choice of data structures to avoid time and space complexities.

Deliverable 3

Code Review: The source code for the function 6 was checked manually and using the tools like **CheckStyle and PMD** plugins on Eclipse IDE.

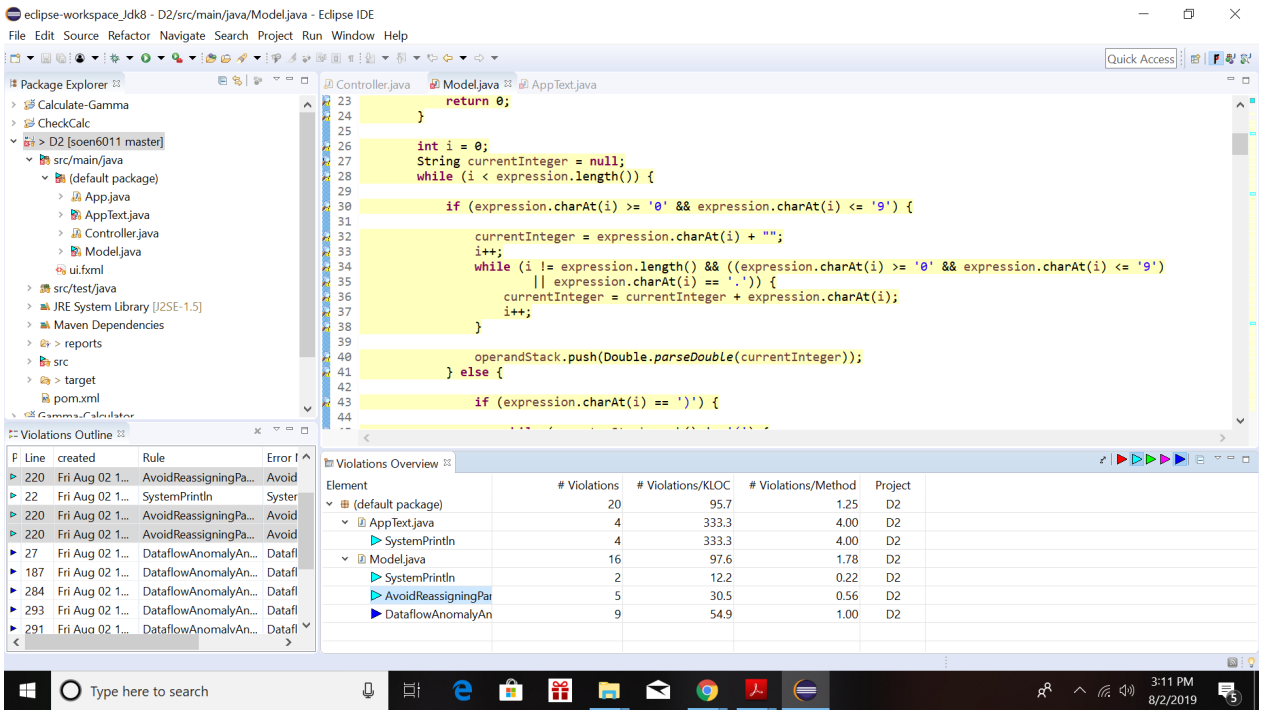

Figure 1: Code Review using PMD and Checkstyle

- **Naming Convention:** Developer has followed naming convention thoroughly. There could be an error of 5 percent but it is acceptable. Further, on checking with checkstyle and PMD it shows some errors for identation, but on checking manually, it shows that indentation is done properly to enhance readability.
- **Readability:** The developer has paid attention on readability of code. Code is perfectly commented at right places also developer has used Java Doc for providing good documentation. There is no commenting for obvious comments which is markable, though there could have been some commenting inside complex methods. Code is grouped and followed DRY principle. Also, developer has paid enough attention for limiting the length of lines.
- **Commenting:** Commenting is done in good amount. It was understandle to me. Also, commenting is not done for obvious places which is a very good approach to stop bloating of classes.
- **Architecture:** Chosen architecture is Model-View-Controller.

  - **Model**: It has two classes separating the necessary logic. All methods require for the implementation are in these classes.
  - **View:** Developer has used JavaFx for User interface. It works just for necessary UI logic and communicates with controller for update.
  - **Controller:** This class takes input from user interface and send it to model and updates the result provided by model. It does it actual function.

  The architecture is followed correctly keeping in mind the functionality of this architecture.

- **Extendability:** The architecture followed is MVC which supports extendability. So, in future in developer wants to extend the functions of calculator, he/she can easily do that by making more classes in desired places.

- **Code Quality:** Developer has Hashset, stack, single loops instead of nested loops, it increases the run time and reduces the time complexitites. Also, developer has paid attention to use less number of variables, and used only at places where required to lesses space complexity.

Conclusion: The code overall is good and written keeping in my the standards of programming. Also, developer paid very keen attention to small details for making it readable, understandable, robust and maintainable. Though, at some places as mentioned above could be slight changes, but overall code is good and shows programmer's deep understanding of programming concepts.

Repository for code review: The code is take from this repository:
https://github.com/rahls7/soen6011. It contains D2 folder where the project is located.

Problem 7

For test case review I cloned the repository into my workspace and performed evaluation of test cases. I had to add Junit5 library in the project to run test cases. The developer for F7 has written all the text cases for the requirements. Test cases also test all the functions implemented. Though, test methods are two but it has 5 test cases.

Report

Below is the screenshot for the test cases review conducted.

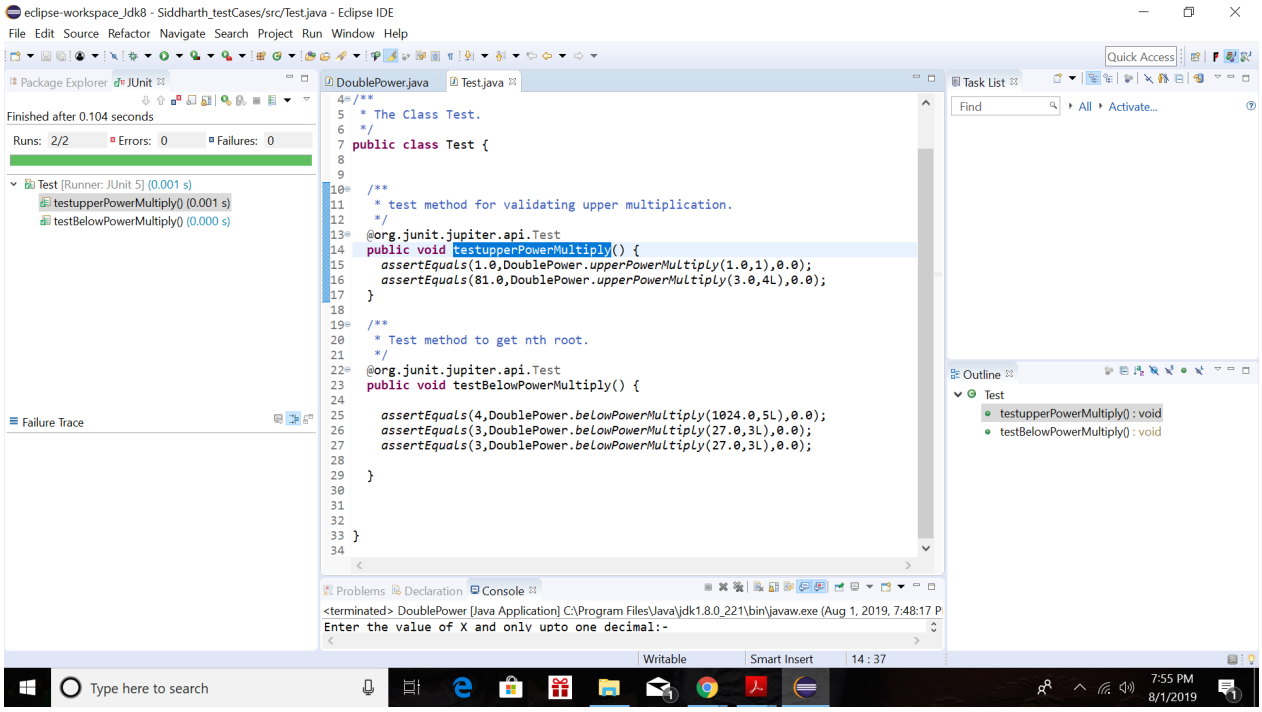

Figure 2: Requirement Testing Report

Summary and Recommendations:

The test cases are written for all requirements but developer could have written some more test cases for error handling, showing robustness of code. Apart from that test cases runs properly.

Repository

The test cases review is done for function 7 which is taken from this repository:
https://github.com/Siddy234567/SOEN6011Sharing