SOEN 6011 DELIVERABLE THREE

Qing Li 40082701

1 Source Code Review for Function 3

Review Approach:

The method that I used to do the code review is a combination of manual method and automatically method. The code review tool I used is Codacy.

For checking the programming style, functionality, comments quality and functional requirements, I did them manually by using the Eclipse.

For checking programming issue and complexity, I used the automatic code review tool-Codacy.

The set of standard practices or guidelines I used is PMD. PMD is a source code analyzer. It finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth. Additionally it includes CPD, the copy-paste-detector. CPD finds duplicated code in C/C++, C, Dart, Fortran, Go, Groovy, Java, JavaScript, JSP, Kotlin, Lua, Matlab, Objective-C, Perl, PHP, PLSQL, Python, Ruby, Salesforce.com Apex, Scala, Swift and Visualforce.

Overview:

Check Points	Satisfied?	Review
Following the Google Style?	Yes	The author follows the programming style we discussed together.
Function Well?	Yes	The author achieves the functionality and correctness of the Function 3. The author provides instructive error handling to user. For example, "Ohh, Please Enter y/n!" "The result is overflow."
Comments Quality?	Good	The author gives pellucid comments for the pivotal part of her code, which provides good readability to code reviewer.
Meeting Functional Requirements?	Yes	The source code meet all the functional requirements described in Problem 2.

Detail Review:
1.CalculatorModel.java

Size
-Lines of code:30
-Source lines of code:17
-Commented lines of code:7

Complexity
-Complexity:1

Issues
(1)"private double output;"
Since: PMD 5.0
Fields should be declared at the top of the class, before any method declarations, constructors, initializers or inner classes.

2.CalculatorController.java

Size
-Lines of code:425
-Source lines of code:282
-Commented lines of code:82

Complexity
-Complexity:22

Issues

(1)"public static double expm1(double x) {...}"
Avoid really long methods.
Since: PMD 0.6
When methods are excessively long this usually indicates that the method is doing more than its name/signature might suggest. They also become challenging for others to digest since excessive scrolling causes readers to lose focus. Try to reduce the method length by creating helper methods and removing any copy/pasted code.

(2) "public static double expm1(double x) {...x = hi - lo;...}"
Avoid reassigning parameters such as 'x'
Since: PMD 1.0
Reassigning values to incoming parameters is not recommended. Use temporary local variables instead.

(3)" if (x == 0 —— x == Double.NEGATIVE_INFINITY —— ! (x ¡ Double.POSITIVE_INFINITY) —— n == 0) "
Since: PMD 5.0
Use opposite operator instead of negating the whole expression with a logic complement operator.

3.CalculatorView.java

Size
-Lines of code:77
-Source lines of code:51
-Commented lines of code:19

Complexity
-Complexity:10

Issues
No issue, very great.

2 Test Case Review for Function 4

Review Approach:

The method that I used to do the code review is a combination of manual method and automatically method. The code review tool I used is Codacy.

For checking the programming style, test cases passing rate, comments quality and functional requirements, I did them manually by using the Eclipse and Junit Library.

For checking programming issue, I used the automatic code review tool-Codacy.

Computing Environment:
CPU: Intel CORE i7
Operating System: Windows 10
IDE: Eclipse
JAVA Library: Junit
Other tools: Checkstyle

Overview:

Check Points	Satisfied?	Review
Following the Google Style?	Yes	The author follows the programming style we discussed together.
All Test Cases Successfully Passed?	Yes	All test cases passed after running by Eclipse.
Comments Quality?	Good	The author gives pellucid comments for the pivotal part of her code, which provides good readability to code reviewer.
Meeting Functional Requirements?	Yes	The test cases are traceable to all the functional requirements described in Problem 2.

Detail Review:

1.LogTest.java

Size
-Lines of code:64
-Source lines of code:28
-Commented lines of code:18

Issues
No issue, very great.

3 Reference

1. https://github.com/pmd/pmd
2. https://www.codacy.com/