

PROBLEM 4 and 6.

SOEN 6011

SOFTWARE ENGINEERING PROCESSES

Github address : <https://github.com/panjingya/SOEN6011.git>

Jingya Pan

40044079

Due Date: 7/26/2019

1 Debugger

Debugger support in eclipse is used to debug this gamma function calculator application. Followings are the advantages and disadvantages that i found during the usage and comparing with the guideline.

1.1 Advantages

Convenient operation: Eclipse support the debug perspective which can be run easily with 'Debug as' >>'Java application'. Before starting debugging, the pre-configurations are needed, which includes setting breakpoints. To define the breakpoints, double click the left margin of the line which need to stop. After some small convenient preparation steps, the debugging can start.

Intermediate status check: During the debugging, it allows to run the program while watching the source code and the variable. The execution of the program stops when encountering the breakpoints, offering the chance to check and modify the intermediate variables values clearly.

Support conditional breakpoints: When the program includes a 'for' loop which loops a large amount of time, for this case, doing the debug from very the very beginning is time-consuming. The conditional breakpoints can help in this case. After set the location of the breakpoints, it is able to change the properties(select 'conditional','suspend when true',and enter the condition), which allows the program directly stop at the breakpoints when it satisfies the condition.

Support drop frame operation: When debugging the program, it is likely to be unconsciously step over a method and not able to follow the change of a value, and need to do the debug from the start. The internal debugger in the eclipse support the drop frame operation, which helps to return back the previous step and call the method again to see the changes. However, limitation exists, explain in 1.2.

Support multi-threaded debugging: From the guideline, when the program stops by encountering breakpoints, the IDE support switching to another program. Which means the two program in this case will be non-influential to each other. This feature will also help the multi-threaded application, maybe in some cases, there is a need to investigate the different status of variables between multiple threads.

1.2 Disadvantages

Does not support actual 'roll back': In the advantages mentioned above, there is a feature called 'Drop frame', however, the operation itself is limited, is not ideal. It is able to step back and to call the function again, but the changed variable values are not able to be reverted. Therefore, calling the function again can see the change but not the actual value performed as the first time.

2 Tool-Checkstyle

The 'Checkstyle' plugin for eclipse is used to review the code specification and ensure that java code follows standard code styles. Followings are the advantages and disadvantages that i found during the usage and comparing with the guideline.

2.1 Advantages

Automatic checking: After installing the plugin in the eclipse, the tool can be used by right-clicking the project and select 'Checkstyle' >> 'Check code with checkstyle', then the warning represented by magnifying glass will show up in the code. In this way, it saves plenty of time to check the coding standard manually line by line, and we can easily modify the improper format by following the given instruction.

Specific checking: The Checkstyle tool can check the code according to the pre-set coding rules(needs to provide a sample configuration file), it can support almost any coding standard. therefore even the specific detail will be concerned. For example: the special valid variable name format, the maximum line of the method body, or even duplicate code checking, annotation format, identifier, etc.

Helps unification within a team: For a relatively complex project, it is likely to be performed by a team rather than individual. In this case, even a programming style documentation is provided by the team leader, it is still not guaranteed that all the members will follow the standard accurately and properly. Therefore, using the Checkstyle tool can help a lot to do the individual checking before merging all the works together. With the support of the tool, the whole program will be uniformed from the appearance.

2.2 Disadvantages

One change need re-compile: Checkstyle can give the warning and instructions only after compiling, which means each time even a minor change is performed, if you want to verify whether it is correct or not , a one more time compilation is needed. If the compilation is not performed, the warning will not be eliminated even the current format is correct, which is really not that intelligent.

Does not provide modifying function: From above we can see that the Checkstyle tool basically focused on reviewing code specification rather than enhancing the code quality and modifying code correspondingly like the tool PMD and Jalopy which provides the modifying function.

Itself have some own error: Sometimes the tool itself come out some errors which need to refresh, clean or build the Project again, or else try to clean all projects and restart Eclipse. Therefore, some extra effort and time is needed sometimes.

3 Quality Attributes

3.1 Usable

The calculator application provides a relative easy and clear front look, which will less likely to make user get confused. In addition, the calculator support the on/off button, which are very familiar elements to users because of the analogy learning in reality.

3.2 Maintainable

The calculator application is maintainable which is achieved by separating the function modules. For example, the verification of input will be necessary for all types of calculation, no matter for gamma, addition, subtraction, multiplication or division. Therefore instead of doing the verification in each type of calculation, I made the verification method separate from the calculation methods. When the verification function is needed, call the method directly. In this way, the program will be easy to maintain and modify for the future changes.

3.3 Correct

The correctness and accuracy is the primary request for the calculator application, since it calculate to 15 decimal place. Therefore, a relatively accurate result will be given through the application process.

3.4 Efficient

The calculator application is efficient which is achieved by simple and clear process, therefore, the user will get a quick respond after entering the input to the application. When the user click a button or perform a specific action, the corresponding method will be triggered to execute the program and return an accurate result and show it in front for presentation.

3.5 Robust

The calculator application is robust which is achieved by concerning the special(extreme) cases and dealing with them correspondingly. For example, for the gamma function itself, the negative integer is not a valid input. Therefore, when the special case occurs, the application will give the instructions instead of break down and show the unreadable error message.

4 Unit test cases

4.1 Test Case 1

Description: Check the user input is a real number or not.

Requirement: According to the Requirement R1.

Test Priority: High

Test Designed by: Jingya Pan

Date for Test Designed: 7.25.2019

Test Executed by: Jingya Pan

Date for Test Execution : 7.26.2019

Name of the Test Title: isNumeric

Pre-condition: Run the application, and enter an input in the field.

Test steps: Get the user input as a string, and change it in the big decimal, if the exception is caught by the block, which means the user input is not a real number. Maybe some special characteristics are included.

Test Data: "123", "120.9", "sss", " ", "!", "—", "@ ".

Expected Results: If it is a real number, then the input will be checked with other criteria; Otherwise, the error message will show up says "*please input a real number except negative integer".

Status: Pass

4.2 Test Case 2

Description: Check the user input in special case. For example, some special format double has the same value of the integer.

Requirement: According to the Requirement R1.

Test Priority: High

Test Designed by: Jingya Pan

Date for Test Designed: 7.25.2019

Test Executed by: Jingya Pan

Date for Test Execution : 7.26.2019

Name of the Test Title: isNegativeIntegerForDouble

Pre-condition: Run the application, and enter an input in the field.

Test steps: Get the user input as a double, comparing the input value with the floor of the input, if it is smaller than $1e - 10$, then the user input is actual an integer. According to the requirement 1, the gamma function not accept the negative integer, therefore there is also an additional condition to check if the value is a negative number.

Test Data: -4.0, 4.0, -4.2.

Expected Results: If it is not a negative integer, then the input will be processed calculating; Otherwise, the error message will show up says "Negative integer is not acceptable, try other numbers".

Status: Pass

4.3 Test Case 3

Description: Perform the gamma function for user input.

Requirement: According to the Requirement R2, R3.

Test Priority: High

Test Designed by: Jingya Pan

Date for Test Designed: 7.25.2019

Test Executed by: Jingya Pan

Date for Test Execution : 7.26.2019

Name of the Test Title: lanczosGamma

Pre-condition: Run the application, and enter an input in the field, and click the button named "Gamma".

Test steps: Get the user input as a double, process the gamma function algorithm, and the result will show up for presentation.

Test Data: 1, 2.0, 6, 10, -2.2, -10.2.

Expected Results: If the user input is 0, then 'infinity' will be shown on the page, which is restricted in the front page; When the input is sufficient large, eg:143, the result will show infinity; Otherwise, the gamma function result will be presented for the given input.

Status: Pass

4.4 Conclusion

The calculator application is tested by using the Junit tesing with the Junit 5 standard. Based on the series of 'assertion' method, the running result and the expected result is compared to check if all the test cases are pass or not.