

SOEN 6011:SOFTWARE ENGINEERING PROCESSES

Deliverable #3

SHASHANK RAO

SID:40104247

<https://github.com/ShashankRao17/SOEN6011-Software-Engineering-Processes>

1 Problem-5

1.1 Source code review for Function F2 [tan(x)] for Team I

Objective

The objective of this code review is to ensure the supplied code for the function 'tan(x)' complies with the mentioned coding standards and guidelines given by the developer, to ensure better code quality if possible and to check for defects in the supplied code on the basis of correctness, efficiency, maintainability, usability and robustness.

Code Review Approach

1. To ensure transparency and fairness in the review process, an approach mentioned by George T. Doran in the November 1981 issue of *Management Review* called "There's a S.M.A.R.T. way to write management's goals and objectives" was used. This approach(also called the *S.M.A.R.T.* criteria) gives a set of objectives and goals to ensure the code review is carried out in an established manner. As per the paper, S.M.A.R.T. means the following(although over the years, the acronym has taken many different forms):
 - **Specific** - an area of improvement.
 - **Measurable** - quantify or suggest an indicator of progress.
 - **Assignable** - specify who will do it.
 - **Realistic** - state results can be realistically achieved, given the available resources.
 - **Time Related** - when can the results be achieved.

(It should be noted that not all of the S.M.A.R.T. techniques need to be quantified for a successful review.)

2. The coding review is also done using static code analysis plugin(CheckStyle) which gives a good sense of judgement on the various industry coding standards available and the ones used by the developer for his/her function and how they comply with them.

Coding Style Analysis

The coding style used by the developer is the Google Checks coding standard and has been verified using the CheckStyle plugin.

A mix of S.M.A.R.T. criteria, key quality attribute perspective and coding standards followed:

Criteria	Explanation
Specific	<p>The developer could have implemented a simple power function(Similar to '<i>Math.pow</i>') for enhanced readability of code instead of implementing it in the following mannner:</p> <pre>double value += (x * x) * ((T8));</pre>
Assignable(Maintainable)	In a scenario where another developer has to go through or contribute towards the development of this function, ease of maintenance due to low coupling is provided along with clear JavaDoc documentation.
Realistic(Correctness)	The functionality has been implemented using the Taylor series and reduction formula and hence the results are close to accurate and realistically achievable as there are only calculations and no approximations.
Time-Related(Efficient)	The system responds to users' input in quick time since most of the coefficients needed for Taylor series approximation are defined as static variables thus reducing the overall execution time and CPU utilization in the long run.
Usability & Robustness	The program interacts with the user using a Textual User Interface(TUI) with appropriate error handling mechanisms and messages in place to ensure robustness.
Coding Convention	The coding conventions decided on by the team have been followed in terms of naming conventions for variables, methods, etc. The only exception here the ordering of class structures since all method implementation has been done after the main function.

Conclusion

To conclude, other than the points on specific criteria and class stucture(coding convention), the overall program is well implemented to encompass all the key attributes.

2 Problem-7

2.1 Test review for Function F3 [sinh(x)] for Team I

Objective

The objective of this test review is to ensure that the supplied implementation for the function 'sinh(x)' complies with the mentioned requirements by the developer, check the correctness and precision of the given implementation and to check the quality of the unit test cases written.

Computing environment and testing approach

For the purpose of testing the implementation with all due fairness and transparency, the computing environment being used is the JUnit testing framework integrated into the Eclipse IDE. Additionally, testing is also carried out manually using the Textual User Interface(TUI) supplying manual inputs to the console window.

Test review analysis

- As per the given user requirement in section 0.2 of the function documentation, it states that " for large values of x the values of sinh(x) will be large in the decimal positions, hence for better readability the output values will be rounded off to 5 digits". But to maintain the correctness of the function and to ensure precise output to the user, the decimal point precision has been rendered as redundant in this case.
- To ensure robustness of the function, the developer has used appropriate error messages satisfying the system requirement given in the function documentation.
- In total there are 5 test cases written using the '*assertEquals*' assertion of JUnit. The test cases however do not test the independent functionalities such '*epowerX*' and '*sinhx*'. This could be an improvement wherein not only the functionality as a whole but the dependent functions/methods are also tested as standalone functionalities.

Conclusion

To conclude the test review, an area of improvement could be the addition of unit test cases for the dependant functionalities such as *epowerX* to test them as a standalone functionality.

References

- i. https://en.wikipedia.org/wiki/SMART_criteria
- ii. <https://nyu-cds.github.io/effective-code-reviews/02-best-practices/>
- iii. <http://thinkapps.com/blog/development/what-is-code-review/>
- iv. Doran, G. T. (1981). "There's a S.M.A.R.T. way to write management's goals and objectives". *Management Review*. 70 (11): 35–36.