

CONCORDIA UNIVERSITY



SOEN-6011 Software Engineering Processes

SCIENTIFIC CALCULATOR

Function 9: Power Function

DELIVERABLE 2 (D2)

Problem 4 & 6

GitHub : <https://github.com/san089/SOEN-6011>

Author: Sanchit Kumar(40081187)

July 29, 2019

1 DEBUGGER

Options Considered for Debugger

1. IntelliJ IDEA 2019.2 Built-in Debugger.
2. Eclipse 4.6 (Eclipse Neon) Debugger.

IntelliJ IDEA 2019.2 Built-in Debugger is selected for use.

1.1 Description

IntelliJ IDEA 2019.2 has a build-in debugger which provides a debugger UI with views for frames, variables and watches. It also provides the user with the ability to run multiple debugging sessions simultaneously. Also, the user can make use of customizable breakpoint properties such as conditions, pass count, and so on.

1.2 Advantages

Below are some advantages:

- Ability to run multiple debugging sessions simultaneously.
- Suspend code execution when a field is accessed for reading or writing.
- Does not crash unlike eclipse debugger and its easier to maintain debug profiles.
- Efficient as compared to other debuggers available.
- The ability of remote debugs in which we can debug an application running on a standalone server.

1.3 Disadvantages

- Debugging UI is less user-friendly as compared to other debuggers.
- No support for overhead statistics for code inspection.
- Faced issues when debugger fails to starts.

2 Source Code Quality Checker

the tool used: Checkstyle Version - 8.23

2.1 Description

Checkstyle is a static code analysis tool for checking Java source code for adherence to a Code Standard or set of validation rules. It provides features such as finding class design pattern and modern design problems. The latest version of Checkstyle is 8.23, with new added rules and checks such as InvalidJavaDoc check.

2.2 Advantages

- Ability to set up custom rules using configuration files.
- Wide range of rules provided by default configuration files.
- Portability - you can use checkstyle for various IDE's such as Eclipse, IntelliJ.
- Easier to integrate with external tools since it was designed as a standalone framework.
- Provides real-time and on-demand scanning options.

2.3 Disadvantages

- Only able to analyze java code that is written with ASCII characters only, no support for UTF-8.
- Code should be compilable to get valid violations.
- Cannot determine the full inheritance hierarchy type.
- Cannot determine the type of expression.
- Cannot implement some of the code inspection features such as Exception class inherited from java.lang.Exception class.

3 Achieving Code Quality

3.1 Correctness

- Mapping the requirements to the modules and test cases to make sure requirements are covered.
- Written requirement specific test cases to make sure requirements are covered and tested.

3.2 Efficiency

- The significant amount of calculation time in my program is for the calculation of log. To make the calculation faster I represent the number x as $x \times 10^y$. By using such notations for the number, the calculation can be made efficient as the part 10^y can always be calculated using a faster algorithm.
- Another step taken to improve efficiency is by overriding the number of iteration run by the program to converger Taylor series values. User has the flexibility to explicitly provide a number of iterations to run for the series.

3.3 Maintainability

- To make program maintainable the first thing is to improve the code readability. I used consistent indentation, proper comments, code grouping and consistent naming scheme to make the program more readable and easy for the maintainer to understand.
- I have written test cases that are automated and modular. Automated unit test cases help the maintainer to easily the program module after changes are made.
- Used a good software design which provides the ease to add new capabilities.

3.4 Usability

- I have used proper terminologies and provided appropriate error messages and sometimes the detailed explanation of the occurrence of the error to make the program more usable.
- I have removed unnecessary information from Text-based Interface that may hinder the important information from the user.
- Tested the code extensively to make sure the user does not run into unforeseen errors.