Robin Karlose
40089313

Problem 5

SOEN 6011
2nd August 2019

Manual Code Review of F9 - x^y

I decided to go ahead with a manual code review of my team mates function - x^y .

There are a couple of reasons why code review is important,some of which include

- Knowledge sharing across members of the team and with members of other teams
- Code legibility
- Consistency of code in large projects
- Correction of accidental errors in code

The approach I took in reviewing the code was I manually evaluated the code in basic details against certain guidelines which I mention in the following points

- **Purpose** - Whether the code achieve the purpose it was originally designed for and whether it meets all the requirements
- **Implementation** - How the code was actually implemented and transformed into something which generates output
- **Legibility and Style** - Whether it followed best practices and certain standard coding styles and whether the code was legible to other users/readers or not
- **Maintainability** - How maintainable the test code is especially with future functional changes/and or scaling

Discussion and actual code review

The following section contains the discussion and actual code review (with some basic details) against the four guidelines discussed in the above 4 points

PURPOSE

- The code achieves the authors purpose to some extent by calculating integer powers of numbers , but it cannot calculate x^y for real number values of y
- To be fair though , the limitation that it cannot calculate x^y for real number values of y is mentioned in the authors functional requirement documents

IMPLEMENTATION

- The code is neatly broken down into 3 classes and 1 test class - the 3 actual classes being APowerB.java , Calculator.java , ICalculator.java (which is actually an interface) and 1 testing class TestCalculator.java
- There is good use of abstraction especially with the class Calculator.java implementing the interface ICalculator.java
- Code does not make use of any libraries and is fairly dependency free
- If I were to re-implement it , I would go into much more logical detail in the functions and try and implement x^y where y is a real number using some approximation technique

LEGIBILITY AND STYLE

- Code is implemented in a simple but extremely lucid manner
- High code readability with presence of comments everywhere
- Code has presence of JavaDoc

MAINTAINABILITY

- Presence of a test class called TestCalculator.java
- The function can be used in a scalable way in a calculator
- Proper Integration tests need to be carried out if the function is used in a calculator along with other calculators
- Change Requests (CRs) should not break this existing code
- More detailed comments would help when integrating with other functions of the calculator
- Room for external documentation like ReadMe , User Manual etc

References

The following article from www.medium.com greatly helped me and inspired me in my code review. The link is below:-

https://medium.com/palantir/code-review-best-practices-19e02780015f

Testing and Test Review of F10 - σ

I decided to test the code with the following situations

- My own test cases , based on my knowledge of the Standard Deviation Function
- Existing Test cases in the calculatorTest.java file
- Replicated all test cases for both categories of Standard Deviation - Sample and Population

The testing environment was my preferred IDE for Java - IntelliJ.

I executed the program and was presented with a GUI where I manually entered my own test cases.

Also I ran the calculatorTest.java file to see if all test cases presented by the author passed.

Testing Observations

The following were my observations during the testing process:-

- The test cases were comprehensive, extremely well written and covered all possible scenarios
- The GUI provided an appropriate error message when I tried to enter anything other than numbers - like characters , strings , special characters etc
- For cases without commas, it was considered one whole number and the error message was different - "Please enter numbers separated by commas"
- Negative numbers were also adequately dealt with
- I did not even notice a small difference in SD when I calculated the SD with the positive equivalents of the same numbers . For instance I got the exact same result when I calculated the SD for "-200,-4.5,-172.89" and the SD for "200,4.5,172.89"
- The calculation for SD was extremely accurate,correct to many decimal places
- Equal support and attention given to both categories of SD - sample and population
- Overall - good robust testing with testing principles and all variety of test cases kept in mind
- Curious to see how this function would perform in integration testing along with other functions of the calculator