

Dongmin Shin

#### -Unit Test

I covered **99** percent of instruction and **93** percent of branches by using JUnit test. I used JUnit from classwork and directory. Since the directory perfectly worked, I used the directory I used for a class assignment and made it as GitHub repository. After I tested my test I used JACOCO to see what branches and lines I am missing. I could not use the mutation coverage method because due to the technical issue. I was not able to solve the problem. I did not use Automated testing because SUT was written in a complicated method. It was more efficient to use just regular JUnit testing for time-efficient. There are 9 Test classes for each class. Some of the display or printing functions, I just called a function in order to not miss the branch.

#### -SUT

Lots of code was written in a similar way. For example, calling a constructor will automatically call other functions. The function. The SUT was working fine, and there was no critical bug, but I had to fix some minor errors such as file path and function that does not do anything or has nothing in it.

#### -Mvn pit test

It is not working. I assume that there is classpath error. Since this SUT does a bunch of different stuff such as reading text files and other weird functions, there might be a function or library that pit tests can not read.

-mvn org.pitest:pitest-maven:mutationCoverage

Was causing

-[ERROR] Failed to execute goal org.pitest:pitest-maven:1.4.2:mutationCoverage

(default-cli) on project arithmetic: Execution default-cli of goal

org.pitest:pitest-maven:1.4.2:mutationCoverage failed: Unsupported class file major version 57 ->

#### -----Additional Things-----

-I attached 2 JPG Jacoco result images names jacoco\_result and jacoco\_result\_details

-All implemented test files in [JUnitTest/src/test/java/arithmetic/](#)