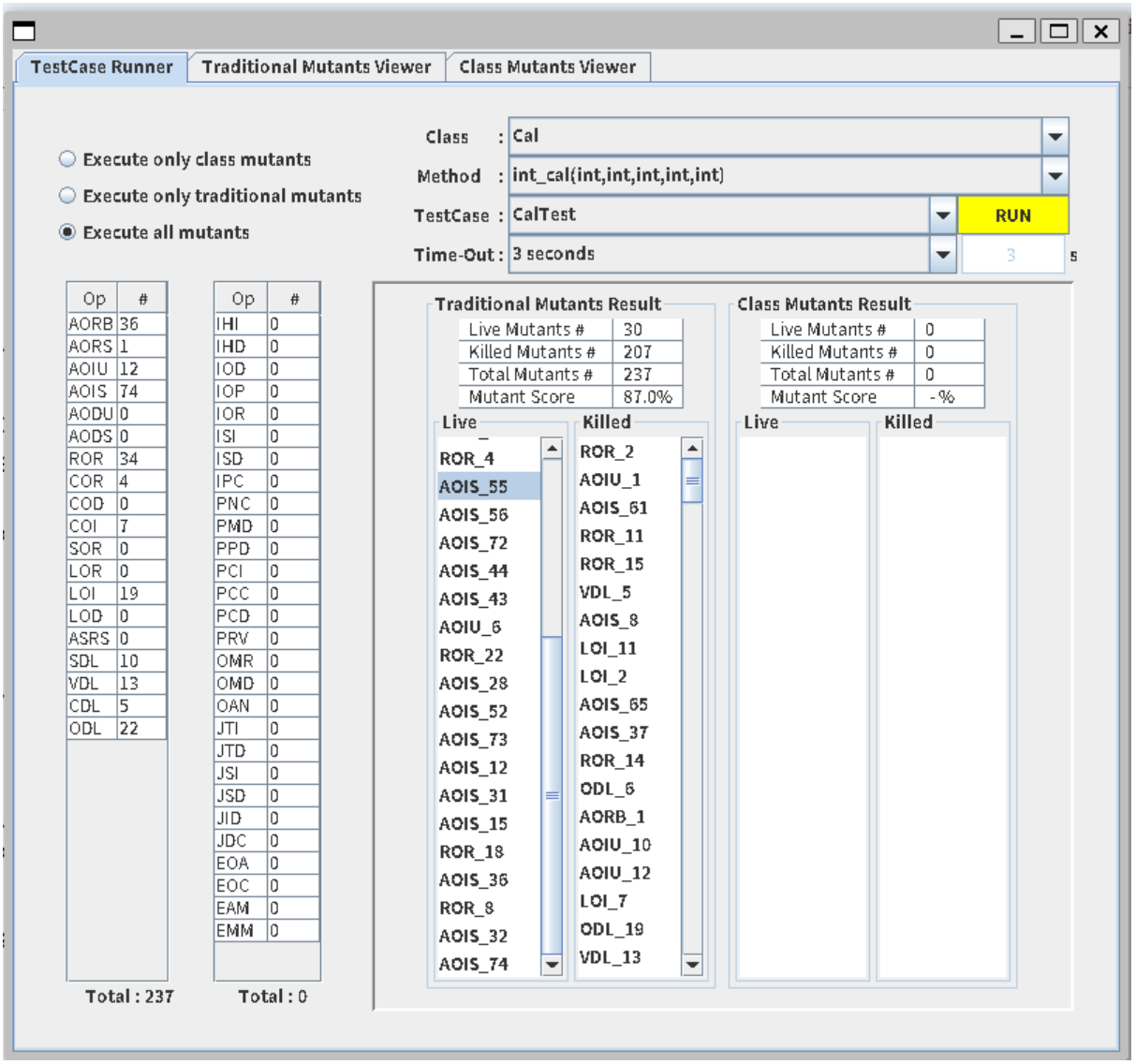
軟體測試 Hw3 Report　　311551169 劉宇承

Download and install the Java mutation tool: muJava. Use muJava to test cal().
Use all the operators. Design tests to kill all non-equivalent mutants. Note that a test case is a method call to cal().

(a) How many mutants are there?



237 mutants in total

(b) How many test cases do you need to kill the non-equivalent mutants?

I design **8** test cases to kill the non-equivalent mutants:

1. Assert.assertEquals(151, Cal.cal(5, 23, 10, 21, 2022));
2. Assert.assertEquals(121, Cal.cal(1, 1, 5, 1, 2020));
3. Assert.assertEquals(181, Cal.cal(1, 1, 7, 1, 2100));
4. Assert.assertEquals(182, Cal.cal(1, 1, 7, 1, 2000));
5. Assert.assertEquals(14, Cal.cal(1, 7, 1, 21, 2100));
6. Assert.assertEquals(59, Cal.cal(1, 1, 3, 1, 2022));
7. Assert.assertEquals(-6, Cal.cal(8, 7, 8, 1, 2022));
8. Assert.assertEquals(0, Cal.cal(7, 7, 7, 7, 2022));

(c) What mutation score were you able to achieve before analyzing for equivalent mutants?

I achieved **85%** mutant killed before I started to analysis the remaining mutant one by one
And achieved **87%** before submit my homework

(d) How many equivalent mutants are there?

In my test, I found **30** equivalent mutants. Since I didn't check all the 30 remaining mutants, I can't sure there don't exist non-equivalent mutants anymore.

Most of equivalent mutants are in following type:

1. make postfix add (a++) on variables that won't be used in the remaining program (e.g. AOIS_51)
2. change logic condition mark that won't influence the outcome (e.g. COR_4)
3. switch the sign of a variable that is used for modular 100 (e.g. AOIU_5)
4. cases that are already covered by program (e.g. ROR_4

Since these faults may produce an error state, they won't produce a failure outcome.

Github: