Gael Dupuy – 300063750 – Lab 8

What is a mutation being killed? :

A mutation being killed, means a mutation was done to the code, which failed the tests. This is a good sign it shouldn't pass them.

What is this report saying? :

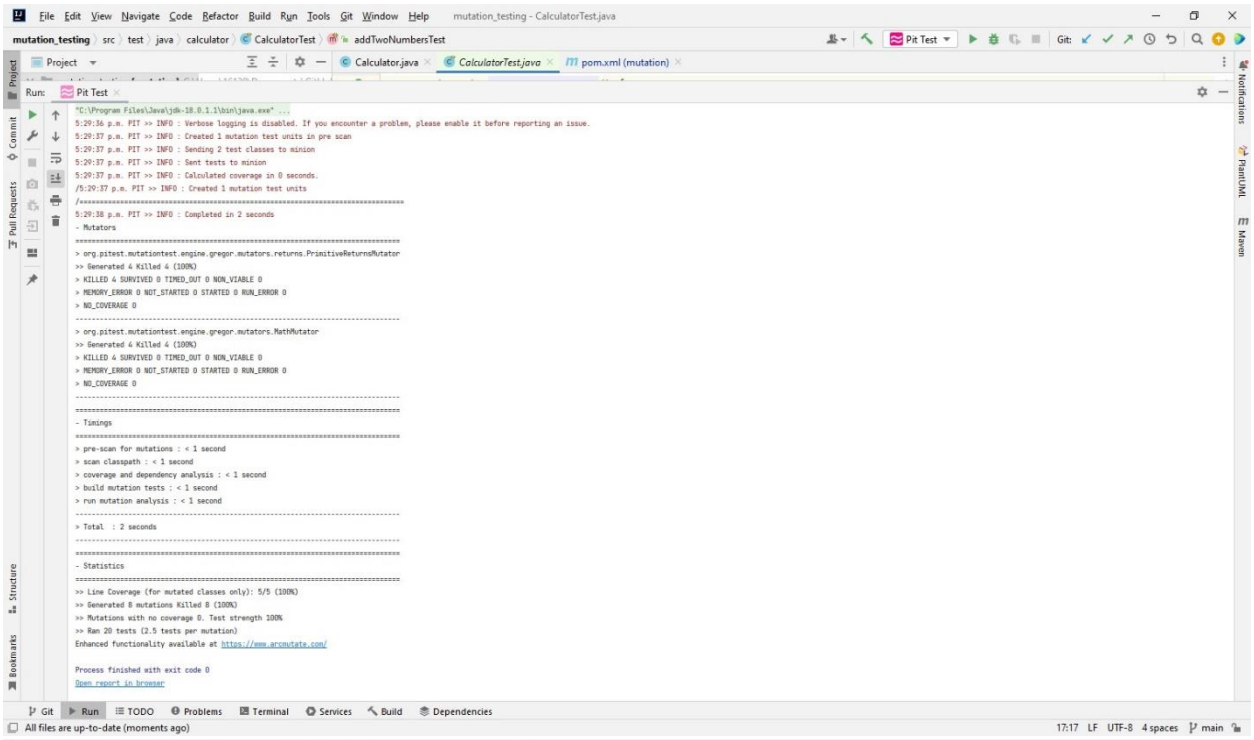
The top section "number of classes" shows the sum of the stats for all the classes tested. In this example, given only one class was tested, the number of classes and the Calculator.java sections look identical, but it wouldn't be the case if more than 1 class was tested.

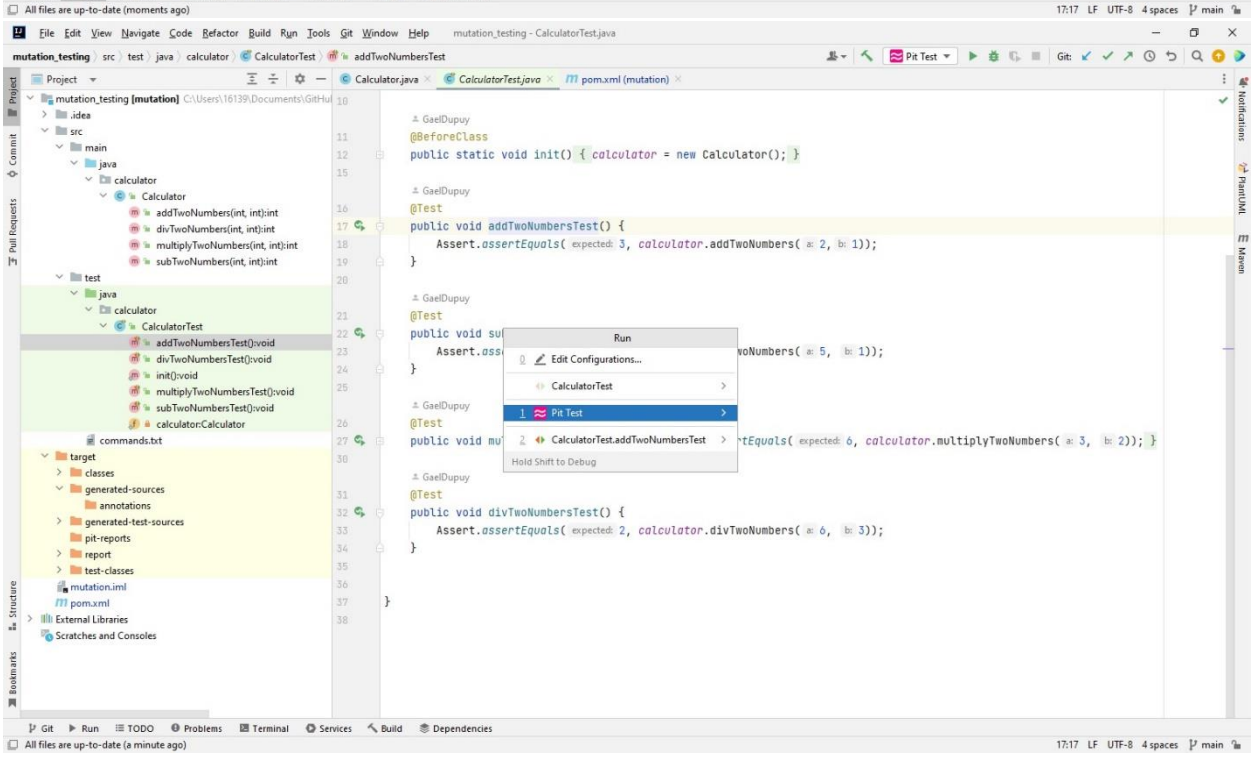
Line coverage indicates how many lines are covered. The result shows us that out of the 5 lines, 5 are covered.

Mutation coverage shows how many mutations were killed out of the ones created. In this case, they were all eliminated which is ideal (you want this to be 100%).

Test Strength: Shows how many of the mutants for which there was test coverage for were killed. In this case its 100%.

Pictures of the test being run and of the result/web report:


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
mutation_testing - CalculatorTest.java
mutation_testing › src › test › java › calculator › CalculatorTest › addTwoNumbersTest
Pit Test
Git:
Project
Calculator.java
CalculatorTest.java
pom.xml (mutation)
Run: Pit Test
"C:\Program Files\Java\jdk-18.0.1.1\bin\java.exe" ...
5:29:36 p.m. PIT >> INFO : Verbose logging is disabled. If you encounter a problem, please enable it before reporting an issue.
5:29:37 p.m. PIT >> INFO : Created 1 mutation test units in pre scan
5:29:37 p.m. PIT >> INFO : Sending 2 test classes to minion
5:29:37 p.m. PIT >> INFO : Sent tests to minion
5:29:37 p.m. PIT >> INFO : Calculated coverage in 0 seconds.
/5:29:37 p.m. PIT >> INFO : Created 1 mutation test units
/5:29:38 p.m. PIT >> INFO : Completed in 2 seconds
- Mutators
> org.pitest.mutationtest.engine.gregor.mutators.returns.PrimitiveReturnsMutator
>> Generated 4 Killed 4 (100%)
> KILLED 4 SURVIVED 0 TIMED_OUT 0 NON_VIABLE 0
> MEMORY_ERROR 0 NOT_STARTED 0 STARTED 0 RUN_ERROR 0
> NO_COVERAGE 0
> org.pitest.mutationtest.engine.gregor.mutators.MathMutator
>> Generated 4 Killed 4 (100%)
> KILLED 4 SURVIVED 0 TIMED_OUT 0 NON_VIABLE 0
> MEMORY_ERROR 0 NOT_STARTED 0 STARTED 0 RUN_ERROR 0
> NO_COVERAGE 0
- Timings
> pre-scan for mutations : < 1 second
> scan classpath : < 1 second
> coverage and dependency analysis : < 1 second
> build mutation tests : < 1 second
> run mutation analysis : < 1 second
> Total : 2 seconds
- Statistics
>> Line Coverage (for mutated classes only): 5/5 (100%)
>> Generated 8 mutations Killed 8 (100%)
>> Mutations with no coverage 0. Test strength 100%
>> Ran 20 tests (2.5 tests per mutation)
Enhanced functionality available at https://www.arcmutate.com/
Process finished with exit code 0
Open report in browser
Git Run TODO Problems Terminal Services Build Dependencies
All files are up-to-date (moments ago)
17:17 LF UTF-8 4 spaces main


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
mutation_testing - CalculatorTest.java
mutation_testing › src › test › java › calculator › CalculatorTest › addTwoNumbersTest
Pit Test
Git:
Project
mutation_testing [mutation] C:\Users\16139\Documents\GitHub
.idea
src
main
java
calculator
Calculator
addTwoNumbers(int, int):int
divTwoNumbers(int, int):int
multiplyTwoNumbers(int, int):int
subTwoNumbers(int, int):int
test
java
calculator
CalculatorTest
addTwoNumbersTest():void
divTwoNumbersTest():void
init():void
multiplyTwoNumbersTest():void
subTwoNumbersTest():void
calculator:Calculator
commands.txt
target
classes
generated-sources
annotations
generated-test-sources
pit-reports
report
test-classes
mutation.iml
pom.xml
External Libraries
Scratches and Consoles
Calculator.java
CalculatorTest.java
pom.xml (mutation)
GaelDupuy
@BeforeClass
public static void init() { calculator = new Calculator(); }
GaelDupuy
@Test
public void addTwoNumbersTest() {
Assert.assertEquals(expected: 3, calculator.addTwoNumbers(a: 2, b: 1));
}
GaelDupuy
@Test
Run
Edit Configurations...
CalculatorTest
1 Pit Test
2 CalculatorTest.addTwoNumbersTest
Hold Shift to Debug
GaelDupuy
@Test
GaelDupuy
@Test
public void divTwoNumbersTest() {
Assert.assertEquals(expected: 2, calculator.divTwoNumbers(a: 6, b: 3));
}
}
Git Run TODO Problems Terminal Services Build Dependencies
All files are up-to-date (a minute ago)
17:17 LF UTF-8 4 spaces main

Pit Test Coverage Report

Package Summary

calculator

Number of Classes	Line Coverage		Mutation Coverage		Test Strength	
1	100%	5/5	100%	8/8	100%	8/8

Breakdown by Class

Name	Line Coverage		Mutation Coverage		Test Strength	
Calculator.java	100%	5/5	100%	8/8	100%	8/8