

Homework 5 – Testing

Part 1

(Refer to Section 8 in the homework description)

Overall coverage of all code blocks after running EMMA: 47%

Screenshot of “EMMA Coverage Report”

EMMA Coverage Report (generated Wed Apr 18 15:43:17 MDT 2018)

[all classes]

OVERALL COVERAGE SUMMARY

name	class, %	method, %	block, %
all classes	98% (401/411)	44% (2046/4609)	47% (34553/73201)

OVERALL STATS SUMMARY

total packages: 8
total executable files: 278
total classes: 411
total methods: 4609

COVERAGE BREAKDOWN BY PACKAGE

name	class, %	method, %	block, %
net.fortuna.ical4j.transform	0% (0/2)	0% (0/4)	0% (0/74)
net.fortuna.ical4j.model.property	100% (127/127)	38% (824/2169)	33% (9945/30161)
net.fortuna.ical4j.model.parameter	94% (47/50)	34% (240/714)	35% (3442/9897)
net.fortuna.ical4j.model.component	100% (58/58)	38% (247/643)	46% (5428/11923)
net.fortuna.ical4j.util	100% (20/20)	82% (78/95)	72% (1351/1881)
net.fortuna.ical4j.filter	67% (4/6)	68% (13/19)	72% (218/301)
net.fortuna.ical4j.data	94% (17/18)	88% (89/101)	74% (2274/3086)
net.fortuna.ical4j.model	98% (128/130)	64% (555/864)	75% (11895/15878)

[all classes]
EMMA 2.1.5320 (stable) (C) Vladimir Roubtsov

(Refer to Section 9 in the homework description)

Number of product lines of code: $13720 + 2515 = 16235$

```
C:\Users\ahram\Desktop\CSWorkspace\CS471\ical4jFolder\ical4j>cloc src/main
1135 text files.
1135 unique files.
901 files ignored.

http://cloc.sourceforge.net v 1.64  T=1.58 s (188.6 files/s, 27093.1 lines/s)
-----
Language               files      blank      comment      code
-----
Java                    190        3620        18384        13720
Groovy                   97         370         3647         2515
XML                      2          14          52          130
HTML                     9          63         234          51
-----
SUM:                    298        4067        22317        16416
-----
```

Number of test lines of code: $6913 + 682 = 7595$

```
C:\Users\ahram\Desktop\CSWorkspace\CS471\ical4jFolder\ical4j>cloc src/test
163 text files.
163 unique files.
24 files ignored.

http://cloc.sourceforge.net v 1.64  T=0.37 s (437.5 files/s, 50890.9 lines/s)
-----
Language               files      blank      comment      code
-----
Java                    143        1984        8301         6913
Groovy                   17         187         543          682
-----
SUM:                    160        2171        8844         7595
-----
```

```
C:\Users\ahram\Desktop\CSWorkspace\CS471\ical4jFolder\ical4j>
```

Ratio of (Number of test lines of code) / (Number of product lines of code): 46.78 %

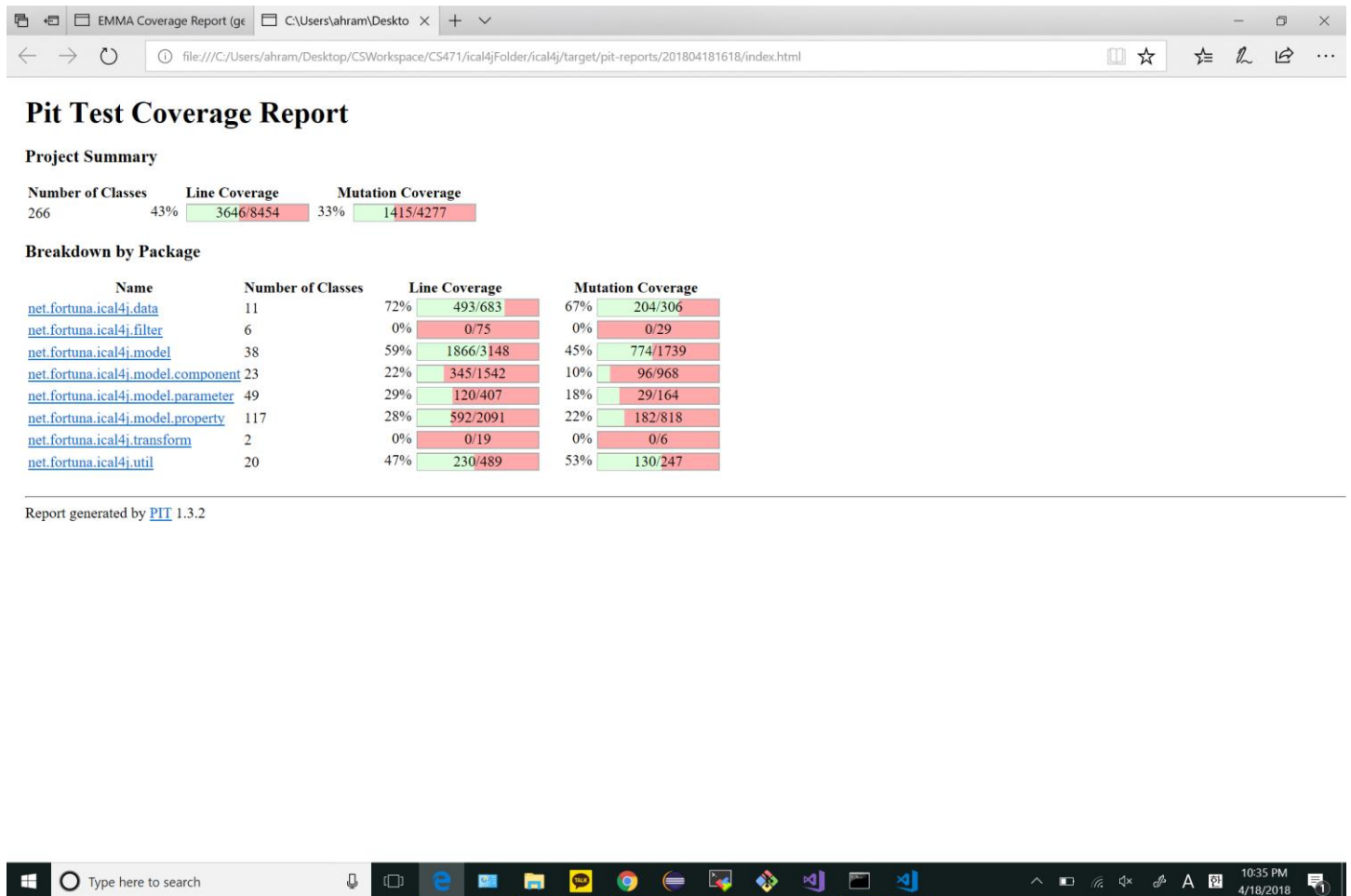
$(7595 / 16235) = 0.4678$ $0.4678 * 100 = 46.78 \%$

(Refer to Section 10 in the homework description)

Overall “Mutation Coverage”: 33%

```
-----
> org.pitest.mutationtest.engine.gregor.mutators.ReturnValsMutator
>> Generated 1236 Killed 406 (33%)
> KILLED 386 SURVIVED 138 TIMED_OUT 19 NON_VIABLE 1
> MEMORY_ERROR 0 NOT_STARTED 0 STARTED 0 RUN_ERROR 0
> NO_COVERAGE 692
-----
> org.pitest.mutationtest.engine.gregor.mutators.MathMutator
>> Generated 151 Killed 46 (30%)
> KILLED 43 SURVIVED 30 TIMED_OUT 3 NON_VIABLE 0
> MEMORY_ERROR 0 NOT_STARTED 0 STARTED 0 RUN_ERROR 0
> NO_COVERAGE 75
-----
> org.pitest.mutationtest.engine.gregor.mutators.NegateConditionalsMutator
>> Generated 1432 Killed 612 (43%)
> KILLED 548 SURVIVED 184 TIMED_OUT 63 NON_VIABLE 1
> MEMORY_ERROR 0 NOT_STARTED 0 STARTED 0 RUN_ERROR 0
> NO_COVERAGE 636
-----
[1;34mINFO[m] [1m-----[m
[1;34mINFO[m] [1;32mBUILD SUCCESS[m
[1;34mINFO[m] [1m-----[m
[1;34mINFO[m] Total time: 34:42 min
[1;34mINFO[m] Finished at: 2018-04-18T16:51:38-06:00
[1;34mINFO[m] [1m-----[m
ahram@DESKTOP-SIV5F8V MINGW64 ~/Desktop/CSWorkspace/CS471/ical4jFolder/ical4j (master)
$
```

Screenshot of “Pit Test Coverage Report”



(Refer to Section 11 in the homework description)

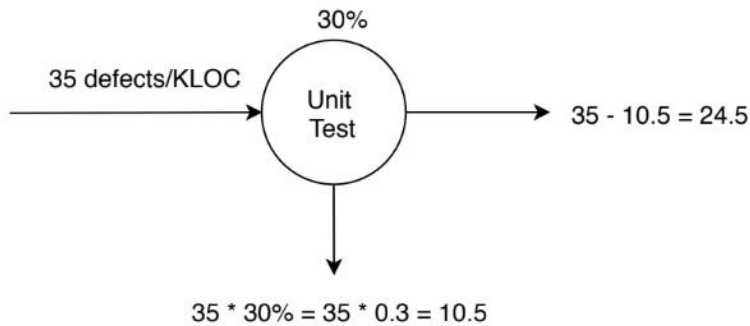
Comparison of three instruments of the effectiveness of the test suite

EMMA is quite fast to get the runtime, and has no external library dependencies. EMMA and PIT create a HTML report with the results for every class. EMMA is only for Java, and PIT is not the only mutation testing framework for Java, but it is the most popular and the one most maintained. PIT took lots of runtime. Ratio of test is easy to install.

Part2

(Refer to Section 12 in the homework description)

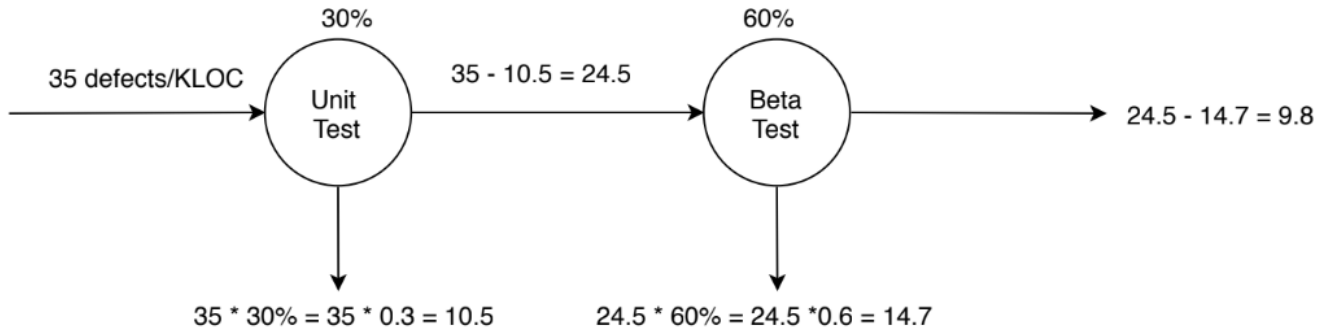
Discuss the Defect Removal Model DRM1:



Estimation of the delivered defect density using DRM1:

24.5

Discuss the Defect Removal Model DRM2:



Estimation of the delivered defect density using DRM2:

9.8

How does DRM1 compare with DRM2?

DRM2 has lower delivered defect density than DRM1. Unit tests easily measured during automated testing. Beta testing validates the product meets our customers' needs.

Part3

(Refer to Section 13 in the homework description)

Discuss your plan for improving a given Defect Removal Model

Higher code coverage is higher test effectiveness. To maximize number of automated tests makes to improve code health, allow safe refactoring, and allow adding feature faster. No single defect removal activity is 100% effective. Improve overall software quality by choosing complimentary defect removal activities.