Nicholas Skinner

5/15/2017

CS362

\*\*Might not compile on flip due to mvn being out of date on the flip server\*\*

**MUTATION RATE:**

**Beginning rate:**

The beginning rate for my tests seemed a little low, with the line coverage stating that I had 60% coverage, and the mutation coverage was ~47%, individually it came down to 70% of appt was covered, 65% of calday was covered, and 40% of timetable had been covered, I had accidentally included calendarmain within my initial run, so my percentage would have been slightly higher

**Final Rate**

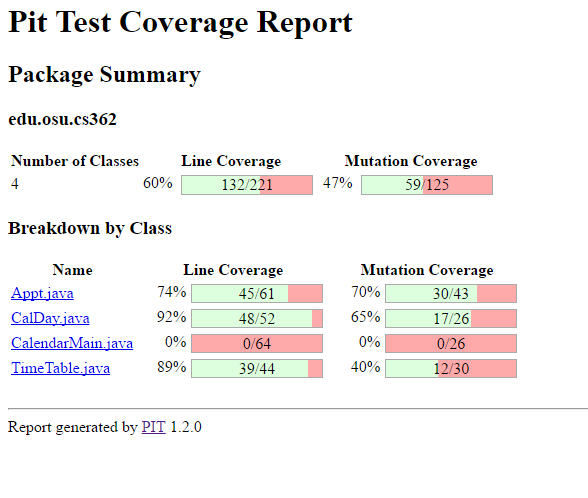
My final rate for coverage felt a little pitiful, but I was unable to understand how to make a dent in the remaining mutations. Thanks to evosuite, I had ended up with 68% of mutations being covered, and 94% of my lines had been covered. This came with 77% of the appt file mutation coverage being covered, 62% of the calday file being covered, and 60% of timetable being covered.

I believe that these scores are very low, and I would not pass a product that had only 70% test mutation coverage, however, there is significant increased coverage, in some cases having about 50% more coverage than it had before. Evosuite, and whiteboarding out the exact methods and interactions were able to contribute to these greatly increased percentages.

I believe I understand why I was getting the coverage that I was anticipating, I needed to reach these methods and try to understand why the thread would die within its limits. I did not understand how methods like getApptOccurences or getApptRange were workinghowever , and was unable to create effective test cases for them. Because of the confusion from working with a new tool, and misunderstanding of the flow of the program. I’m not certain why they were able to survive, and if I was able to understand, and if the code was all reachable, I would be able to get decent coverage

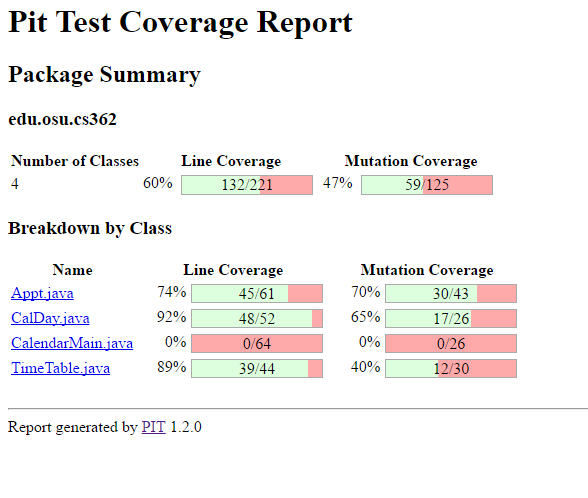
**Screenshot progress:**

**Beginning**

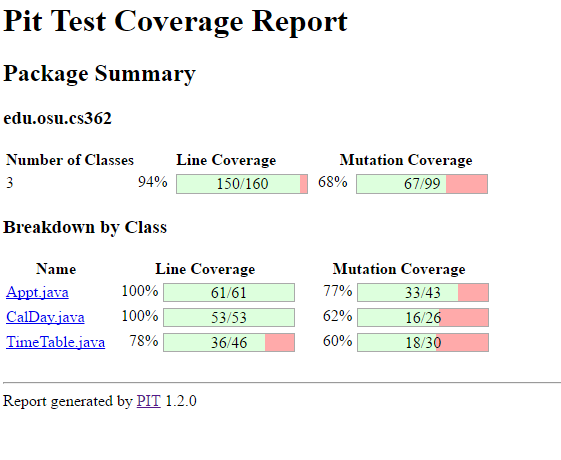


**Personal Progress:**

After a number of attempts to make additional test cases that approached some of these mutations, I had been unsuccessful to increase the percentage, or the line coverage due to not understanding code. This is basically the same image as the previous, but here it is with my added code:



**Final**



**OVERALL EXPERIENCE:**

**Problems faced:**

One large problem I feel with running these testing assistance programs is the initial setup and run, there often isn’t enough documentation for specific errors a user may be receiving, and troubleshooting materials for more obscure errors are almost non-existent making it more time efficient to outright delete a method that had been made rather than troubleshoot it.

**Overall opinion:**

I liked the graphic display of where exactly an error is occurring, and the displays of where exactly a mutation is reaching problems, however, I wish that these mutations were a bit more descriptive with the problem that some of them occur, and I couldn’t understand the problem they were having with most of the mutations