Clinton Wadley

https://github.com/cwadley/CoffeeMakerQuest2

CS 1632 - DELIVERABLE 2:

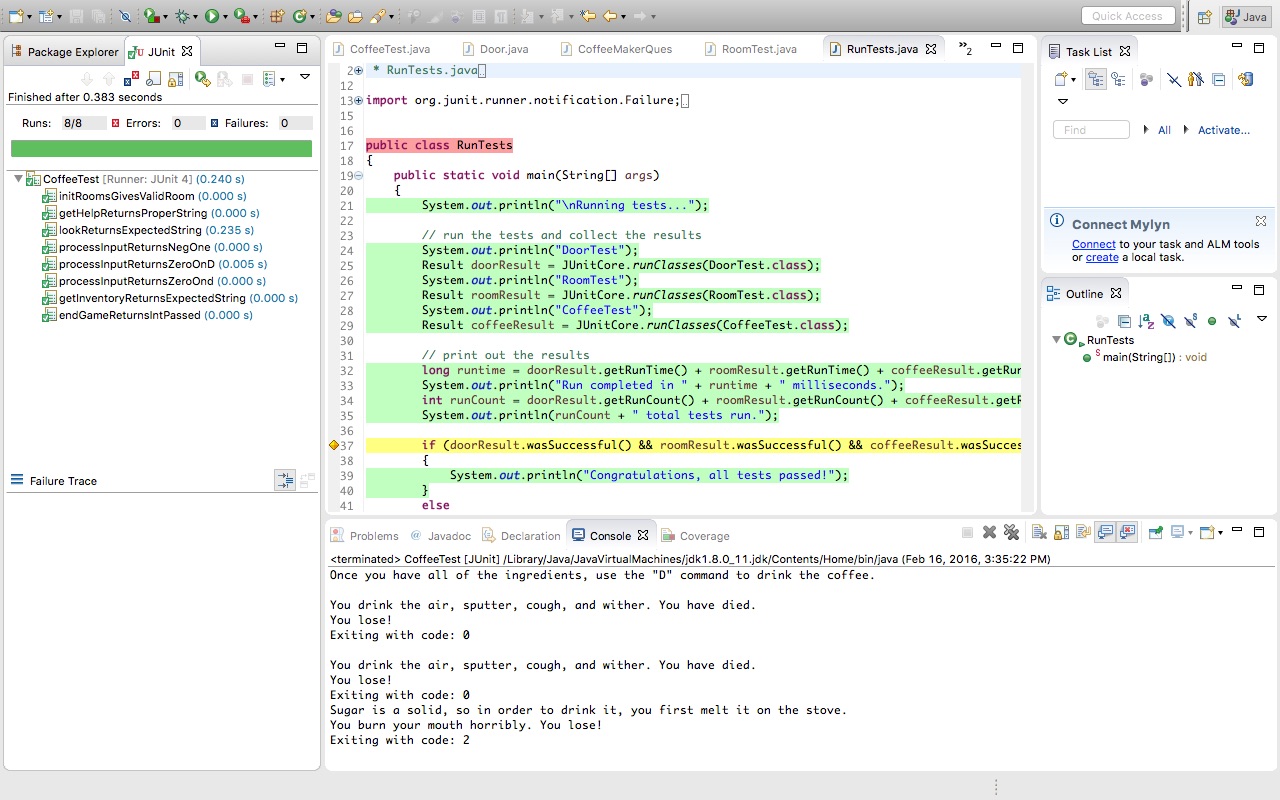
Unit Testing and Code Coverage

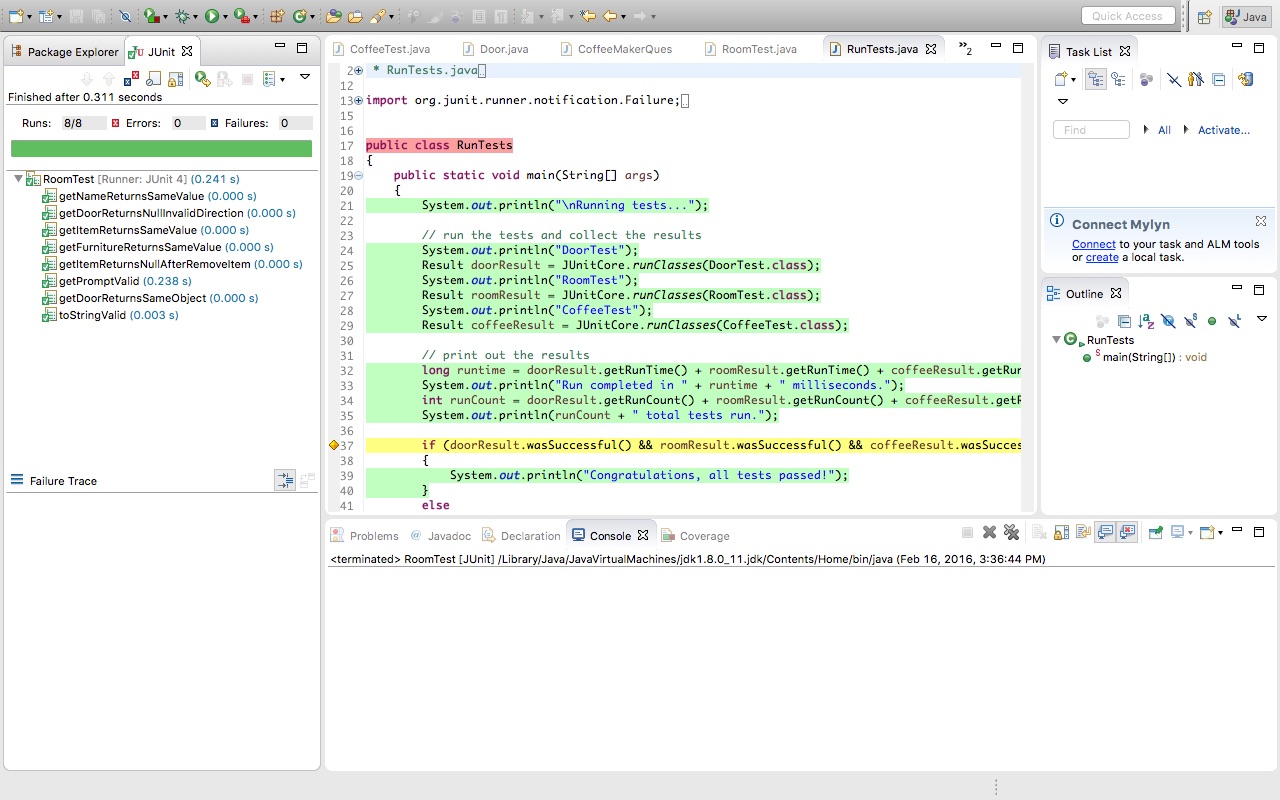
In this assignment, we were tasked with building a reproduction of the Coffee Maker Quest game in a test-driven way, and to write unit tests for all of the public methods. This proved to be quite difficult, as it required a shift in my usual development approach. Writing unit-testable code requires it to be broken up into many more methods than I am used to, and those methods should, as much as possible, follow a give-then-return pattern. In other words, methods should by and large be focused on a single task, taking a parameter and returning a testable value.

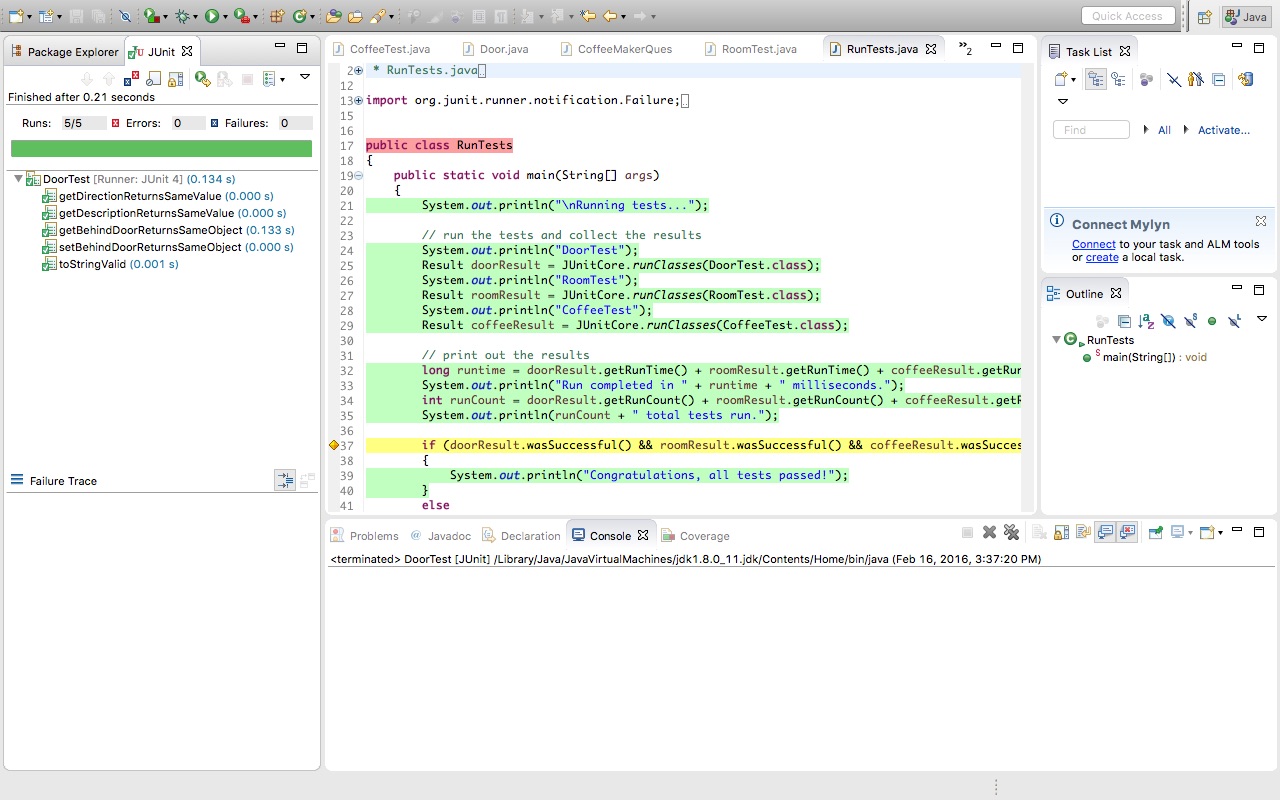
This shift in thinking was by far the most difficult thing for me in this project, and required me to rethink my approach several times throughout the project. I split many methods up into more, smaller methods. I also had many tests fail at first, both because of bugs in my original program files, but also bugs in my test code.

**Test Run Screenshots**

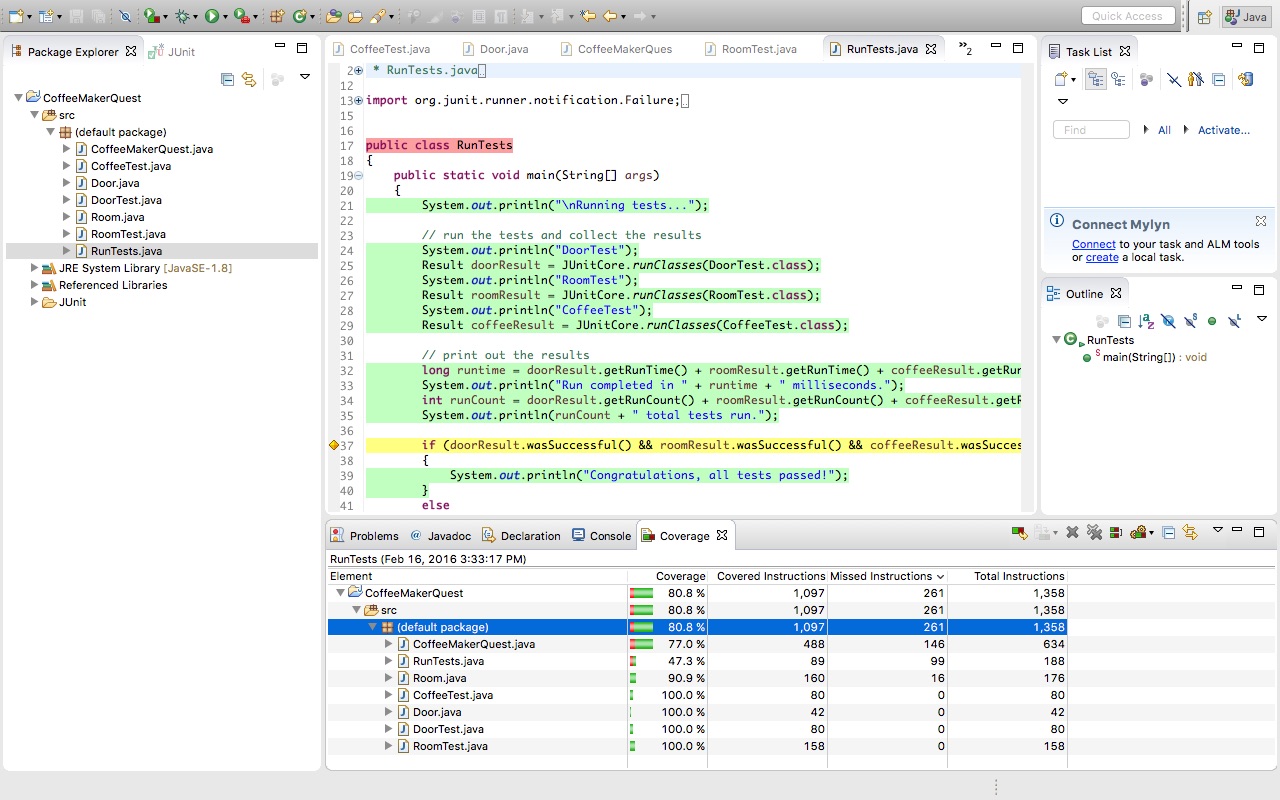
The program is broken into three classes, the CoffeeMakerQuest class, the Room class, and the Door class. The screenshots below show the test runs for these three classes.

****

****

****

**Code Coverage Screenshot**

****

I ended up with 80.8% code coverage. The amount of code coverage was decreased by string output code and certain branches of my program that proved difficult to test.