CS-320-T4208 Software Test Automation & QA

Module Four Journal – Unit Testing Approach and Writing JUnit Tests

Computer Science Department

Southern New Hampshire University

Eric Farkas

3/26/2023

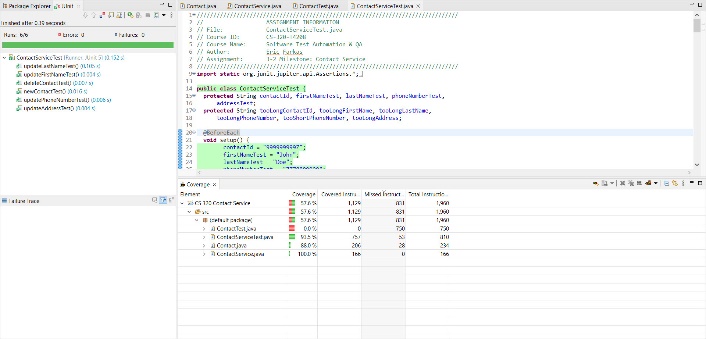
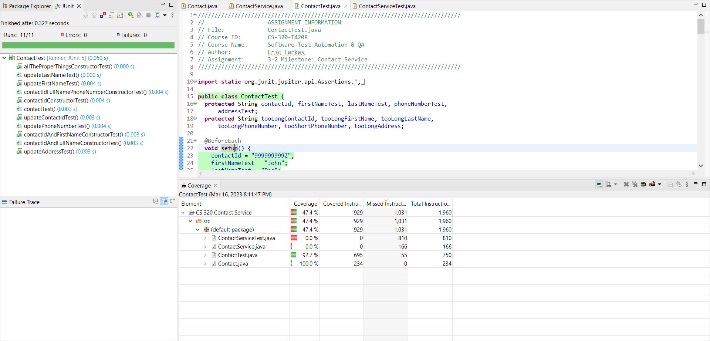
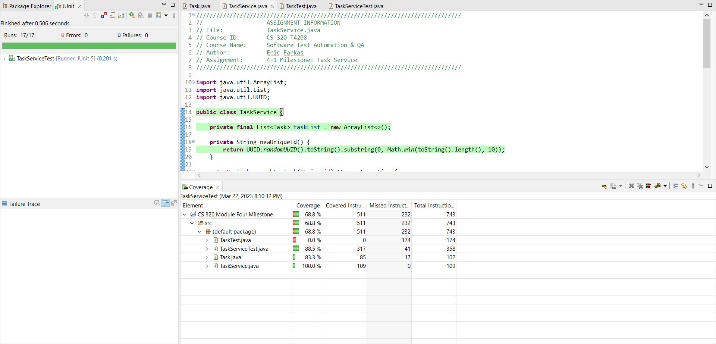
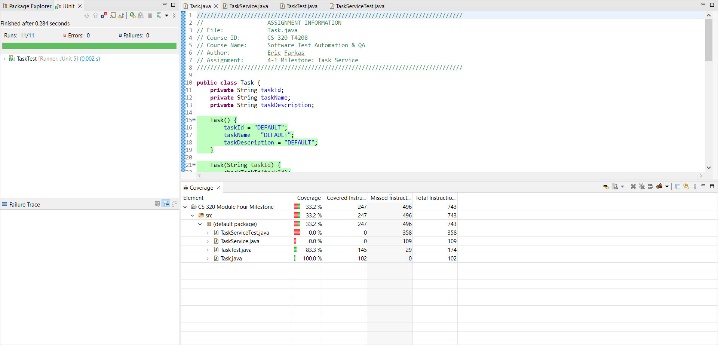
The past two modules have focused on unit testing using JUnit. I have limited experience in this area but JUnit appears to be an adequate and easy-to-use framework for unit testing. The framework helps a tester ensure they have reached the simplest testable part of the code and is easily customized to match the code specifications. It also allows default values to be defined for easy visibility between known good and random inputs. The specific questions related to this assignment are addressed individually below.

**To what extent was your testing approach aligned to the software requirements? Support your claims with specific evidence.**

The testing approach was focused on the requirements. This is not to say that best practices and general concerns did not influence the final methodology but the main focus was the requirements. For module three, the contact information had maximum length limitations as a primary requirement and a secondary requirement to ensure fields were populated with data or not null. This was easily accomplished by assigning an object with a maximum length to each input string. This made testing easy because I could establish default values within the length criteria and then add values that were too large and too small. There were also requirements to guide users if improper input was provided. For module four, the task requirements were similar from a maximum length perspective but the fields are longer and the data is less predictable than for the contact objects. The same approach was used to assign objects a maximum length for each input string and ensure they are not null. The same test method worked here by providing default data plus data that was too short and too long. End user guidance was also implemented in the module four code.

**Defend the overall quality of your JUnit tests for the contact service and task service. In other words, how do you know that your JUnit tests were effective on the basis of coverage percentage?**

By breaking down the requirements into small and easy-to-test units, I am confident the testing is effective. This is supported by my coverage values being returned as 100% for all classes in modules three and four. It is important to avoid reliance on the JUnit tests as a sole source of quality information as the tests are only as good as the method established to validate against the requirements. It is reasonable to do this in this assignment as the requirements are simple and without ambiguity but it is not reasonable to believe this will always be the situation.

**How did you ensure that your code was technically sound? Cite specific lines of code from your tests to illustrate.**

To me, technically sound means that it functions without errors, meets the requirements, is clear and concise and follows best practices. I first established a class and method list using clear and easy-to-understand names. I knew default values for the input strings would need to be declared. I listed several variations of sample data that met the requirements and then selected data that would be easy for others to understand and count characters. There were several code iterations to simplify it and to make the code efficient and easy to understand. More comments could have been added for clarification, but the code’s nature is simple and commenting was outside the assignment’s scope. I reviewed the code visually to ensure no typographical or syntax errors while checking that the methods, objects and strings were consistent. After compiling, the code worked as expected, and the automated testing resulted in 100% coverage. I added some user feedback as the code felt incomplete with no feedback. Examples of this are on the last page of this document.

**How did you ensure that your code was efficient? Cite specific lines of code from your tests to illustrate.**

The efforts put into planning, code development, review and testing contribute to efficient code. No extra effort was put into this code to increase efficiency but I did use methods to assign data specifically to memory in a way that the strings and objects are efficiently managed. This should prevent memory leaks which would cause inefficiency. I also reviewed the logic to ensure no loops would repeat unnecessarily. Examples of this are on the last page of this document.