Ariel Baez

April 19, 2024

Project 2

During the creation of the software, I made sure to keep the requirements as the most important when writing the code. The contact class stated that the first and last name could not be null, so I made that the there where no more than 10 characters inputted. In the Contact test the JUnit ran a test to check and make sure the input was not longer than 10 characters. The task class made sure that the task and unique ID could not be more than 10 characters. If an ID was longer than 10 characters, the JUnit test would run an argument statement.

When I was doing each assignment my JUnit test was getting better each week. The task test had a very low coverage percentage, and the service test had a high coverage. When getting positive coverage, the JUnit was running correctly and checking the codes’ function. I used strings and arrays to make sure my code was covering all the bases. To keep track of all the contacts, I had made a list for the string the example is below:

Public Static List CONTACT = new ArrayList();

Constantly checking the code for errors and ways to make it better helped me perfect the code to its highest potential. Previous classes knowledge has helped with the creation of the code, but many things are hard to remember so going to look at videos and learning from them helps jog the old memory. I tried to make sure to keep everything organized and simplified throughout the codes that I had to create each time. Many times, doing trial and error and just seeing what this does helped me understand the process and testing frequently was the best idea. When I made the contact list, I made sure to label all the variables that were in a contact before comparing them to the requirements.

During all the project sections that I had to work with I believe testing was the most important to make sure that I was able to accomplish the requirements that were set. The biggest requirement for me was making sure that the names were not left blank and were less than 10 characters. I tried to make sure that all my names were alike to not cause any confusion later. The names in the JUnit were the same as the core classes. I made sure to go one step at I time to make sure that I kept everything organized and not causing any issues later making sure each requirement was covered before moving on to the next requirement. I used JUnit testing and white box testing to verify the program.

Static testing was not something I used during my design because I felt like finding the issues it was easier for me to run the code. I will say using the debug button as I am writing my code is very important to me because I can catch errors as I am writing. Static testing is used a lot especially when writing bigger coder to make sure that time constraints and budget are kept to a minimum.

Before starting these courses, I had no idea how much work went into building a program. I was not aware of how much software testing goes into the project. I thought when they tested the programs it gave you the errors at the end and told you how to fix them, I thought of it like spell check. A tester is very cautious with their design and is very detail oriented. As a tester myself throughout this course I believe that making the best program possibly takes time; at the end of the day, I believe we want to satisfy the client and give them a quality and functioning code.

The best way to limit someone’s thinking your code is biased is by making sure you are open to any changes offered and that you are keeping yourself humble. If we test and try to correct our own code to us it looks perfect but having outsiders’ view, they might find room for improvement or find errors we might have overlooked. Many times, we get a hot head and think nobody can program better than us and that in tail effects our code because we think we are untouchable.

Discipline is something as a software engineer that we need to have to deliver excellence in our code. We as engineers should follow an ethic of always being true to our design and never take any shortcuts to create a code and never skip a test that the code might need, to deliver the product on time.