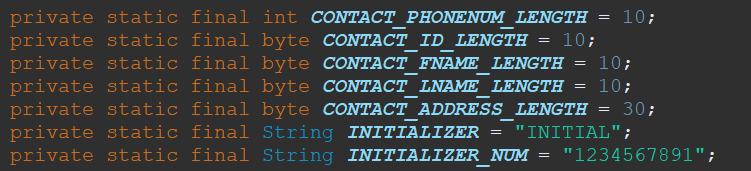
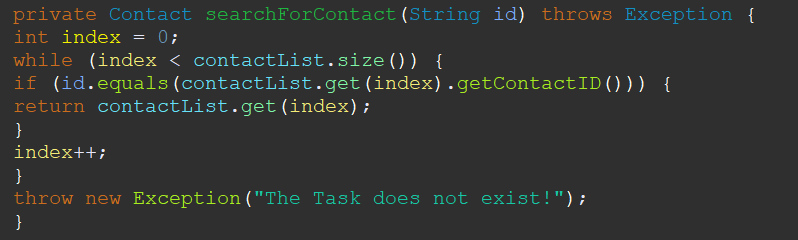
Zachary Nicholas

October 13, 2023

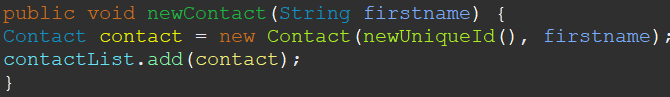
CS-320 Software Test Automation

Project Two

The way I had approached creating all features for this software’s requirements was to make sure I looked at exactly what they had wanted the software to do such as the hard limits they had set involving specifically the Contact service and how they wanted the ID, first name and, last name shouldn’t be longer than 10 characters for which I set up a test case that would throw an invalid argument exception if one of these fields is either left blank or larger than 10 characters. Another place that I had analyzed the requirements was for the Appointment service and how the date needed to not be before the current date which my test cases accounted for with the date java utility. From what I can tell my test cases had achieved everything they needed to do as well as making sure that my code had in built ways to restrict the amount of freedom that the contact service had by limiting the number of characters each of them could have

When it came to my experience in coding the Junit tests I found them to be mostly enjoyable as they were mostly straight forward after doing the weekly readings for them. With the Junit tests I believe it accounted for all of the requirements that was asked of me and continued to be very useful in rooting out areas in which I could’ve forgotten especially if the invalid argument exception was thrown. I believe the code that I made was very modular and uses the appropriate data structures which I believe is best shown in my contact service in its search for contact and delete function

With this method I used proper syntax for a while loop as well as throwing an exception if the contact, we were looking for wasn’t found in our array list. With another one of my contact services of adding a new contact only took about 3 lines which I believe shows its conciseness



During this project, I had used White-box testing which is an automatic testing. What white-box testing does is let a system take some input and evaluates how the system processes that input to provide the output we suspect. Often white-box test is performed by the same developers that had originally wrote the code, this is done purposefully as they have full knowledge of the code’s structures and inner workings. As the sole developer for this project, I had the knowledge of what every line and the input and outputs were supposed to be, The test cases that I made also made sure that there weren’t anything that slipped past in order to keep with the requirements I was given.

In this case there was no need to use black-box testing, as often the testers are independent of the development team and often don’t have any knowledge of the inner workings of the code that the developers would. In this case I was not required to have a dedicated or outside tester validate my code.

Black-box testing is often used in development of larger consumer products, some of those being the development of operating systems of a phone or computer. Often the consumers of these products are the end users, which are often not given access to the inner workings of that product and only see the product as its final form. Often the testing that is preformed is the white-box testing variant as people generally don’t have a dedicated tester for all of their work.

During this project I often hadn’t needed to make anything this in-depth or requiring the number of classes and different objects and variables inside it, so the plan I went with was to meticulously go through each of the different requirements in order to make sure I had all of the things required by my client in this project. This was easier to do after creating the first one as most of the requirements were the same or at least similar enough that they could be reworked into what I need for that specific case.

During this project, I was often confused on what exactly I needed to do but found it easier to do after looking into the background of the assertions rules, I also found it easier to understand as the weekly readings continued. The sheer gravity of the project I made over these weeks is really crazy to me as I went into this not knowing anything about assertion or about Junit testing.

I feel as though it is very disciplined in any profession especially one like software development or testing as often these products will make their way to an end user which could affect them in negative ways if we didn’t make sure to fully develop it correctly or missed things that could pose a security threat in the future. I think one of the main things that often gets to people in a development role is ego which can often make people rash and cut corners because “they have done something lie this before” or “its not their first time working in this” which can prevent someone from doing their work diligently and can often cause complication for the software if they have a cocky air around them. Often in these scenarios it’s good to approach them in a way that first takes into account all of the requirements of the client and make sure to have a checklist that ensure their completion.