##### Ryan Bermel

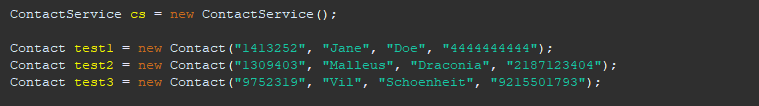
##### CS-320

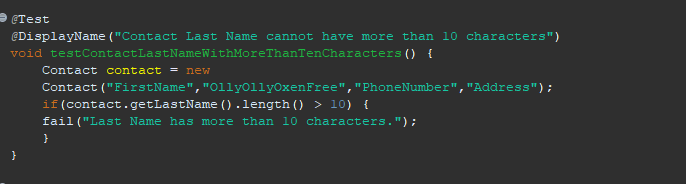
##### Project Two

##### 4/11/2023

Unit testing is an important step when it comes to building a software solution. By doing unit testing, one can reduce the risk of errors emerging and improve the quality of the code. The approach I used aligned with the system requirements for the three features created. For example, in ContactService there was a requirement for the contact to have a unique user ID that could not be longer than 10 characters. It also could not be null or updatable. A first and last name that could not be longer than 10 characters or null, as well as a phone number that had to be exactly 10 characters was also required. Lastly, an address that couldn’t be longer than 30 characters was included.

The overall quality of the unit testing was decent. I tried to create code that would fulfill the requirements that were needed. I didn’t change much in the code this time around as it seemed to be working fine. I did, however, struggle a bit trying to piece everything together to ensure that it ran properly. After taking some time and consulting forums to try and come up with a viable solution I was able to get it in working order. An example of this is shown below:



As I am not the best at coding, I tried to incorporate multiple techniques to ensure that my code was up to snuff. One such technique was I added a test case to check to make sure that the information being entered met with the requirements that were in place. 

This code also checks the length of the information to ensure that it wasn’t longer than 10 characters to keep it efficient.

**Reflection**

The Dynamic testing method was one of the testing techniques I used in this project which tests the dynamic behaviors that occur in the software's code. This technique tests the variables that are considered not constant and finds the weaknesses in certain areas during the software's runtime. A technique that I did not use during this project would be static testing, static testing being a way to test the code without running it. The biggest difference between these two testing techniques would be that the identification of most defects cannot be so easily found with the use of dynamic testing. This includes the development standard breaches as well as the detection of dependencies and the inconsistencies in the software models.

The mindset that I tried to adopt while working on this project was one that was analytical and calm. I used caution throughout this project to watch for errors and to use trusted tutorials to ensure success. It is important to appreciate the complexity and interrelationships of the tested code as each class needed to meet the given requirements and work correctly.

When it comes to reviewing your own code, I think that it is important to try and leave out one's own biasness. Despite how confident one can be in their written code; the code needs to run and work properly without major bugs or errors. If one is too confident in their code to assume that the code works without running the proper tests, it could lead to major issues in the future.

Self-discipline is an important aspect in one's work. Being disciplined in the workplace can ensure that everything is working as intended and won’t lead to any major errors down the line. It is important to avoid cutting corners to avoid compromising the overall quality of the code and the performance of the final product.