Arooj Saeed

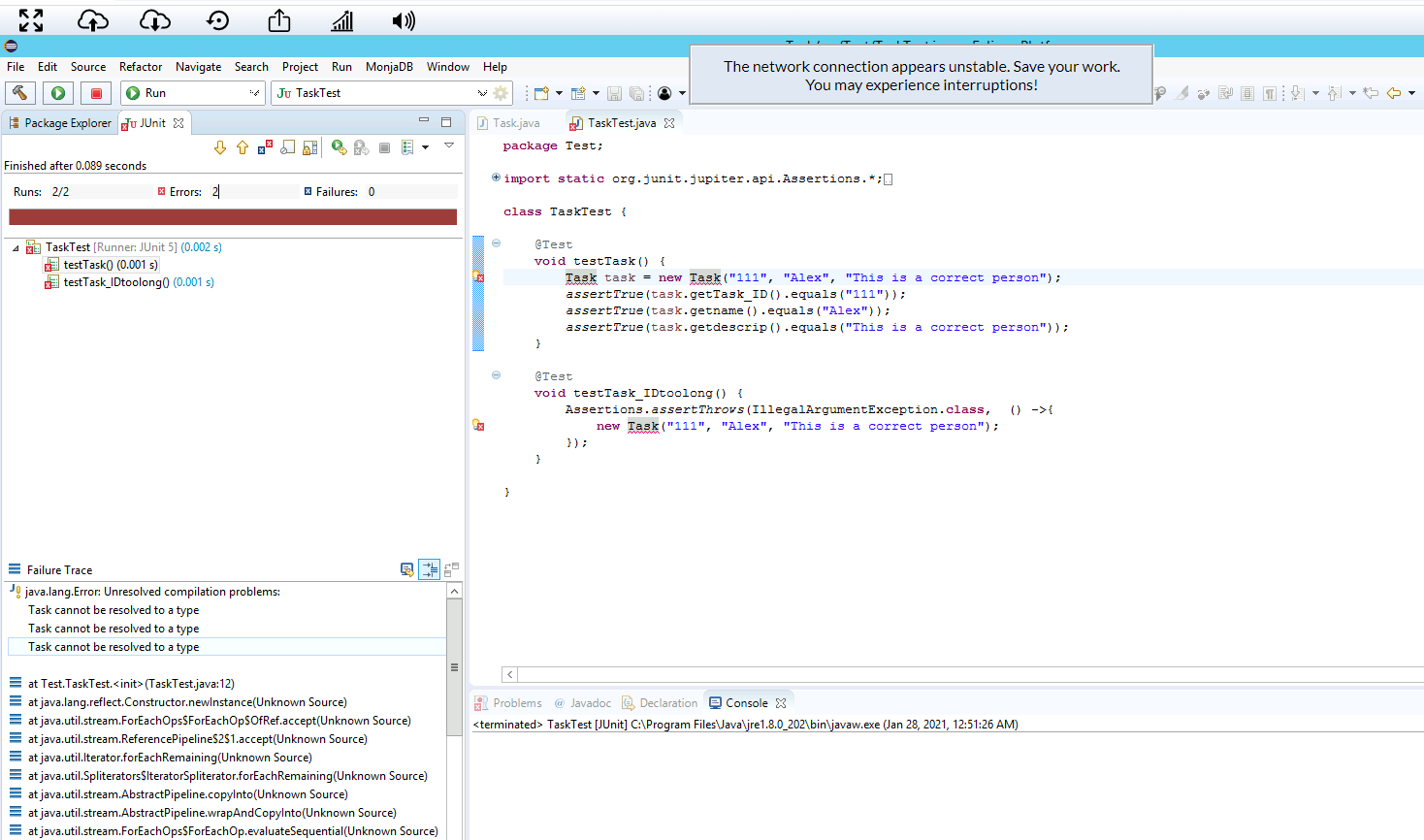
CS 320

Project two

February 19, 2021

This was my first-time experience for testing approach to the provided software requirements. As I am used to get the zip folder assigned to each assignment, this folder contains all the required files and I just have to add write own code in fix me section. Then I was good to go and after successful debug, I usually submit the assignment. When I see the requirements in module three, I was shocked that I have to write the whole new code by myself. Since I already have taken Java classes ahead, I just need to go through simple revision then I was able to write the assignment by myself. I feel proud of myself after writing the whole code. I tried to complete all the requirements that was provided in the module. For example, the first requirement was to add four files in the zip folder, I created all the files and after coding I zipped them and submitted them.

In module four, I was confident in myself but got stuck in JUnit testing. Since it does not debug properly as shown in the attached picture:



This was a whole new challenge, and I spent a whole week but still was unsuccessful.

In the contact service, I wasn’t sure how to use JUnit testing, so I didn’t try it. But after revising the module four resources, I had an idea on how to run the JUnit testing using the resource approach. I am confidence in my code and was 100% sure on my task.java and taskService.java coding files. But I am unable to do the testing on them since it does not compile. But I feel like it will go through the process after resolving the errors.

After writing the code and debugging it I did not find any error, which gave me a hope that my code is technically write. Plus, I mentioned all the requirements that was asked in the program. For example, defining the objects correctly,

protected String task\_ID;  
private String name;  
private String descrip;

In module three, I wrote the program using simple java tools with defining and implanting the strategies provided in module three resources. For example:

//Putting the requirements together  
//contact id requirements  
if(contact\_ID == null) {  
System.out.println("error ");  
}  
else if (contact\_ID.length() < 10) {  
System.out.println("error ");  
}

I feel like my code was efficient because it goes through a successful execution after debugging with zero errors. For illustration:

//phoneNum requirements

if(phoneNum == null) {

System.out.println("error ");

}

else if (phoneNum.length() < 10) {

System.out.println("error ");

}

//address requirements

if(address == null) {

System.out.println("error ");

}

else if (address.length() < 10) {

System.out.println("error ");

}

And in module four,

public Task(String task\_ID, String name, String descrip) {

if(task\_ID == null || task\_ID.length() > 10) {

throw new IllegalArgumentException("Error");

}

this.task\_ID = task\_ID;

this.name = name;

this.descrip = descrip;

}

We have worked with different testing techniques and strategies. From week 1, we started with introduction to these software tools and techniques. Later, we worked on why the role of testing is important. In week 2, we have learned about continuous integration which means sharing all the development tools and work on one platform. Moreover, we also have learned about the difference between dynamic and static testing. Static testing comes before dynamic testing. From week 3, we have been introduced to Junit testing, which is important step of testing. In Junit testing, different test cases were run to check the validity of java programming. From week 3 to 5, we have continuously developing java programs through milestones and then implementing Junit testing through various strategies. The program we created involve Contact Service, Task Service and Appointment Service. Junit testing is a framework of testing, we mainly learn how to run Junit4 testing.

In Junit 4 testing, we do the basic testing. We learned about Junit 5 testing too but did not

implement it yet. Junit 5 is the latest version of Junit family. It works with including dependencies to the system. Junit 5 consist of three main parts while Junit 4 have all in one place. Junit 4 requires java 5 while Junit 5 requires java 8 or above to run. In Junit 4 we use org.junit.assert while Junit 5 contains org.junit.jupiter.assertions.

Static and dynamic testing are performed to check the code from errors. But static testing is run before compiling the code while dynamic is performed after compilation. Both are perfect but dynamic testing is performed before static testing to check the requirements data with some sample data. Before running dynamic testing, we will not be sure about the defects. The dynamic testing while running help reduce the cost of execution and time. We will find the errors before executing the real product. The main difference is that static testing runs after dynamic and dynamic can only be generated when the code is ready to be executed. Both of the testing are really important in performing real world tasks.

Testing as a software developer is not an easy task. I have to focus on every detail of the project I was creating from writing a task to its test case, I have to see what is allowed and what is not. For this purpose, I simply uses the assignment rubric, so I can explain each and everything. It is very important to appreciate the complexity of the code as the whole software is standing on the code. As we have read in the discussion, failure to go through the steps of coding, can result in big incident and can cause loss to millions of dollars.

Yes, bias cannot be easy to deal with. I tried my best to solve my own problem but got stuck in testing, after constantly emailing to the professor and working with him on his suggestions, I still felt like I am stuck on Junit testing.

The best part of the software develop coding is to be discipline and follow the requirements. Developers are provided with specific time period to accomplish their task. Doing the task within the time frame is the key to success. I am planning to avoid this by setting goals to myself and completing my tasks within these goals.