

Naba Alrubaye, Dillon Cote, Campbell Lindsay (Team Hangman)
10/20/2019 (Sunday)
Group Homework 7 – COMP730/830

The Homework 7 Report Questions

- **A link to your group's GitHub account which will have your group's source code**

https://github.com/drc1313/COMP730_HW7

- **Describe any problems your group had writing the unit tests**

We did have some problems when we were writing the unit tests, they are as follows:

- DAO layer was not ready for use when we were writing the unit testing
- Exceptions also were not ready to use when we were writing the unit testing

We got over these problems by mocking the DAO layer and Exceptions by using Mockito.

- **Did you change the classes that were tested? Why or why not?**

We did change some parts as mentioned bellow but thanks to Mockito for mocking the necessary methods or exceptions, so we don't need to modify or develop additional functionalities for unit testing. However, we changed some parts such as:

- I added new constructor inside the CustomerAccount
- Created NoAccountCreatedException class
- Created NoSuchCustomerAccountException class

- **Report on the percent of the code covered, as reported by Clover.**

We had multiple issues with Clover but covered the tests as follows:

- Testing the NoAccountCreatedException
- Testing the NoSuchCustomerAccountException
- Testing the createNewAccount
- Testing the updateCustomerName

The percentage is: a 100% for the CustomerAccount class.

- **Explain the difference between a Mocked object and a Spy'ed object.**

The attached reference is where we learned the difference as followed: “If you want to be safe and avoid calling external services and just want to test the logic inside of the unit, then use mock. But, if you want to call external service and perform calling of real dependencies, or simply say, you want to run the program as it is and just stub specific methods, then use spy.”

Reference: <https://javapointers.com/tutorial/difference-between-spy-and-mock-in-mockito/>

- **Do you think unit testing is important? Why or why not?**

Unit testing is a key part of any program for many reasons, 3 of which we will highlight. Unit testing is important because it helps segregate each part of the program and tests that those parts are working correctly. This helps by isolating the smallest piece of testable software from the rest of the code and determines how we would expect it to. Unit testing also helps by making the process more Agile in nature, creating an iterative development cycle. This is beneficial due to having to sometimes change old code and make edits. Lastly it assists by improving the overall quality of the program. It helps detect problems in the code before you write and finalize the code to predict issues and scenarios that can be sent to integration for mapping.

- **Name 3 key parts of the Observer pattern**

Those three key parts are – subject, observer list, and observers.

- **When would you use the Observer pattern? What problem does it solve?**

The observer pattern is used when a one to many relationships is required, so that when one subject changes, everything is updated to reflect the changes automatically called by their methods. This will solve having to manually change subsequent observers within the observer List.

- **Name 3 key parts of the Command pattern**

Those three key parts are – client, callback interface, and receiver.

- **When would you use the Command pattern? What problem does it solve?**

The command design patterns intent is to encapsulate into an object all the data required for performing a given action. This pattern allows us to decouple objects that produce commands, which include the method to call and the arguments to use. This solves object-oriented callbacks and encapsulates requests as objects.