VIN: FINAL PROJECT MEMO CS445

Topics seen in class that are covered in this project:

- OOD Methodology

- UML Basics - Interfaces

- Polymorphism and Inheritance

- Designing With Objects

- OO Principles - Testing

As it is possible to observe, most of the issues the professor has explained in class are posed in the Wine Club. This is a good method to assure students have attended the classes either on campus or online.

As another good point of it, is makes the students to learn even if they do not want, because if not, the probably fail this course missing the submission of this project.

Project status

The submission includes all the required deliverables, named as follows:

- cs445 (code)

- Final memo.pdf

- VIN use cases.pdf

- UML use case diagram.png

- UML class diagram.png

- UML Search Sequence Diagram.png

Some use cases are not implemented, such us “modify shipment”.

In the "VIN use cases.pdf" are explained all the use cases implemented on this project, the ones that are left are not coded.

Number of hours

Number of hours you needed to get the code working:

- from 160 to 200 hours.

The number of hours you spent preparing your submission:

- from 10 to 12 hours (UML diagrams, memo and use cases explanation)

Challenges

1. Since the very first moment, the big deal has been how to build this project. How to start coding, there was no javadoc to follow the steps so figure out how the structure had to look like has been the main challenge for it.

2. Jackson libraries. It is the first time I have been working with them and I have spent a lot of hours trying to have the code working. Also I think that the information about Jackson libraries is a mess, there are too many different libraries and at the end I downloaded a .jar file including everything that I needed.

3. How to create a script or a makefile to make this project run in the command line. It is not the first time I have run code from the command line but it is the first project that includes libraries that need to be included during the compilation time.

4. Deal with a project with such dimensions has been hard. I did not just need to code and forget another class that I had just coded before, I had to take into account everything, I could not miss anything in order to make it works.

5. I had some troubles trying to save and restore the state of the Club. It did not work with the code that was provided by the professor so I had to change everything on there. Finally I have used Object Mappers to carry it out.

6. The most important challenge, the time. In the beginning I did not thought it could steal me that amount of time. It seemed to be easier.

Solutions

Using Maven I could solve almost all of the above challenges. With time and patience it is easy to learn new technologies.

To create the project I used the following commands (once I downloaded Maven):

```
$ mvn archetype:generate
-DgroupId=com.companyname.bank
-DartifactId=consumerBanking
-DarchetypeArtifactId=maven-archetype-quickstart
-DinteractiveMode=false
```

This instruction auto-generated all the structure of the project so I just had to copy and paste the classes that I already had implemented in the corresponding folders.

Unit Testing

The code has been tested using JUnit. Maven automatically runs the test files when you compile/install the program.

The whole program has been tested without leaving any important aspect. Main classes has been verified that work correctly, however, I have not tested classes that just includes a constructor and getters/setters.

Coverage

Once again, I have used Maven to check the coverage of the application. Including a plugin called “cobertura”, it generates a very detailed report regarding coverage of the code.

If you want to check how accurate is the coverage you can spend some minutes watching which classes are the ones that decrease the total amount of lines covered by the test.

Recommendations

I would have appreciated a more clear description of the project. Not just what the VIN club is, a more detailed explanation about the command line interface would have eased the work a lot. Some times I did not know what was needed as a return of an execution and I had to figure it out by myself.

Also in the description of the project there are some functionalities that are not required then to implement the CLI. I am not saying it is a bad thing, just that I started coding following the description and after that I had to change some stuff due to the modifications.