Jacob Slaubaugh

Java 2

Bob Trapp

Version 3 Project Reflection

Java 2 has taught me a lot this semester. At the end of Java 1 I was still a little confused on some of the things that were taught to us like, what a method and attribute was and how it works. But Java 2 has really cleared up to me on how we can use stuff like that and why it’s important. Object Oriented Design includes encapsulation, abstraction, polymorphism and inheritance. After this semester it makes more sense to me because of us having to practice it.

Our first version of the project wasn’t the best because we didn’t communicate very good as a group. We also didn’t have a DAO to use for CRUD functionality in CSV files. A DAO is something that I was uncertain of until we started working as a group to create the project. I learned that it shouldn’t be used for user interactions with your program. It should implement an interface that can talk to your database or other types of persistence mechanisms. An interface was another reference type that was unclear to me until we started creating the CSV part of the project. We set up our project so that our DAO classes could inherit the abstract methods of the interface. So that whenever we needed to add a new way to create, read, update and delete a different file type the new class could implement the interface.

After fixing our DAO and interface classes in version 1, our version 2 went a little faster. In version 2 we made a better user interface by limiting what the user can enter. For example, we changed the date so that the program would use the computers local date, instead of allowing the user to enter something wrong. Another thing we did was change the menu to provide what accounts the user could choose from. If they didn’t type in an account that was listed, it would keep asking them to enter the correct account.

The last version of our final project was done individually. Since we had everything coded for CRUD in an interface all I had to do was create the class for the database DAO and implement the interface and then add a couple new methods to match my use cases. My program would call the stored procedure with the parameters it got from the user, and then store the results into an arraylist that could be printed in the main menu. Something that I didn’t have time to mess around with was the main menu. It should’ve been set up so that it was only for database transactions instead of having CSV and XML included.

Even though working in groups was a little painful in the beginning, it was something that helped me get a better understanding with what we were taught in Java 2. It provided me with good examples of when we could use an interface. It also gave me an idea of what it would be like working for a company that organizes projects in groups. I learned how to code CRUD functionality for CSV, XML and database files. I would have liked to spend a little more time on the database part of Java. I wasn’t very sure how to use triggers, so I made everything into store procedures. Making a stored procedure transaction was another part of SQL I would’ve liked to talk about because it was something I added to my delete stored procedure towards the end of my project. Overall Java 2 was very useful for when it comes time to go get a job.