Alfredo Rodriguez
Juan Gutierrez
November 8, 2020
Instructor: Daniel Mejia
Programming Assignment 4

I confirm that the work of this assignment is completely our own. By turning in this assignment, I declare that I did not receive unauthorized assistance. Moreover, all deliverables including, but not limited to the source code, lab report and output files were written and produced by my partner and I, alone.

1. Program Explanation

For this assignment, we had to merge both our PA3 programs into one. We used half of Juan's program and half of Alfredo's program. Once we were done merging both programs, we refactored it to add new functionalities to it like asking the user to input a password and creating an interface class.

The toughest part of this assignment was merging both programs. We decided to go over each of our programs and make a list on which implementation would work best from the two. Once we had our list, we started adding them into a new project that we added to Github for version control. While we were adding our methods or classes, we were changing either parameters or object creations in order for them to work correctly since we both had different styles of coding. Once we figured that out, it was pretty straightforward to continue with the new functionalities asked by the instructor.

We broke down the problem into smaller problems and we asked ourselves who will be best to tackle those problems. We made another list and divided up the functionalities between the two and made comments about each other's implementations before pushing our changes to Github. Overall, it was a challenging assignment at first, but we got to get a groove on how to work together and everything ended up being successful.

2. What did I learn?

For this assignment, we learned how to work in teams to create a functioning program. Both of us had never used version control so we learned a lot from this. We believe our solution works very well with the problem and it is hard to improve on it but if we had to say one thing, we believe we can improve in the bank statement class. Having a lot of print statements to print out a bank statement might not be the most useful approach.

Merging both our programs took us about three or four days to get it working properly. After that, it took us only a few hours to implement the new functionalities.

3. Solution Design

For the overall functionality of the program, we created something that looks like a bank website where we have the user interact with the computer asking if they want to deposit money, withdraw money, transfer money and to select the accounts that they would like to do all that. We created multiple classes that some interact with each other and also created a lot of methods in order for the program to work correctly. We created new classes and method to have the bank manager print out a bank statement. We also added a method to have a file as input to do all the things a user can do.

Our approach was to split up the functionalities and comment each other's work for what could work best. This approach was really helpful because we needed to be in constant contact because the implementations had to work with both our styles.

We went ahead and used an Array List because that is the data structure that we are more used to when working with Java. We did not make any assumptions. The steps in the assignment were straight forward and we implemented whatever was asked.

4. Testing

To test our program we focused on procedurally testing individual parts of our code with input that could be given that would be useless to the program and break it, we utilized white-box testing since we both had access to the source code and understood the underlying logic to our program.

We did not test our solutions enough, since we found out from demoing with another team that they were able to break the program in certain spots. To improve our testing practices in the future we should create a list of what we plan to test and inputs we should put in to test the program. Test cases we used for our program included, attempting to use negative numbers where they should not be allowed, attempting to give characters and strings as input where an integer or double was expected, and trying to find individuals that did not exist in the bank.

We broke our program many times and used those breaks to fix and create a more robust program to handle unexpected input from users.

5. Test results

The results of our testing were many exceptions thrown by the system as well as many unintended actions by our program that we were able to fix without too much issue. Screenshots include in attached file as well as document outputs

6. Code Review

Person One (Alfredo Rodriguez)

My first impression of my partner's code was surprised because his main method was really short compared to mine. I spent a lot of time on mine trying to make it with fewer lines and couldn't come up with something. When I saw his main method, I was surprised how clean it looked.

One thing that I really liked about his program was the use of switch statements. I used a lot of if else statements on mine and had to do the try catch to prevent an invalid input. When I saw his switch statement, it was a better implementation because of the default case without using try catch. Another thing that I liked was how clean the program looked.

There was nothing really that I didn't liked about his program but if I had to say one thing it would be that he included all of his methods in the RunBank class. Looking at his code did not really change my understanding of the Bank system. We had a lot of similarities which was really helpful because we did not disagree with anything any of us did or wanted to do.

Person Two (Juan Gutierrez)

My partners code seemed solid when I went through it. The most notable thing I saw when going through it was their implementation of create account made much more sense as well as the implementation of their transaction log had taken up less lines of code than mine. There was not anything that I really disliked in my partner's code.

Looking at my partners code did not change my understanding of the Bank System but it did provide me with a different point of view of how they approached certain problems in creating a bank system

7. Reflection

The process we went about combining code began with us reading through and getting an understanding of each other's PA3 code and creating a plan of how to tackle the assignment before we started. After looking through each other's code we began coding the simpler portions of the program that were most similar in our programs, those being: Person, Account, Checking, Savings, and Credit. Once these classes were combined and working properly we began combining the more complex portions of code taking the best portions of each other's code while having each other easily contactable in case we needed elaboration on certain functions and how they worked until we had the required functionality for PA3 ripe for additions needed for PA4.

The process of understanding each other's code mostly consisted of us individually going through each other's code and asking questions about functionality as we came across blocks of code we may not have initially understood completely while looking at it.

The main problem we faced was figuring out how to approach combining our code and what we should combine where. This issue was quickly remedied once we took time in a voice call to hash out a game plan, which covered which code we planned to implement and where it would be most useful.

8. Demo of another team

We demo'd with Minh Dang and Ervey Montes. We understood their process to perform tasks perfectly. Everything was self-explanatory and we didn't struggle in knowing what to do. We did break their code. They did not handle some exceptions when the user would input letters instead of numbers. Besides that, we could not break the code further. From what we saw, they did include all the functionality requirements. We were really impressed by how neat their program looked.

9. Demo for another team

We demo'd with Minh Dang and Ervey Montes. They understood the prompts of our user interface and did not struggle in knowing what to do. They did break our code. When they used the pay someone functionality, when they selected to search for the other user by name, our program threw out an exception because we were doing the equivalence with a string and integer. They also tried to send more money than what the current user had in their account and it did go through which resulted in negative amounts, which was an oversight on our part. They also brought to our attention that a csv file with new balances and accounts was not created when a user created an account but only after they performed an operation in the bank in addition to that new csv file not being able to overwrite old version of the file. The rest of what the other team broke and noticed were minor issues, one being one of the created files being of the wrong format and the account numbers for a new user not iterating correctly. but we fixed the pay someone method as well as the new bank csv file creation ability to overwrite and to be created after creating a new user and the minor bugs with formatting and iteration of account numbers. We do include all the functionality requirements.

Person One (Alfredo Rodriguez)

I contributed in this project by adding half of my PA3 program and with new functionalities. To help solve the problem, I created a list to break down parts in which I could work on and he can work on. I sent him multiple messages to understand parts of his code or to tell him what I plan to do and when I pushed some changes so he can get his IDE updated. We both contributed a lot for this assignment so I will say that I did 50% of the assignment.

What I learned from working with a teammate is that not everyone will code the same way I do and there will be a lot of different ways to solve a problem.

Person Two (Juan Gutierrez)

I contributed about half of my PA3 to this assignment with new functionality we added for PA4. To help solve the problem I pushed ideas of ways to solve specific issues, as well as setup the Git repository for the assignment. I contributed about 50% to the assignment.

Working with a teammate taught me how valuable getting a second opinion is, since two people can look at the same problem but have two different ideas of how to approach it. Having a teammate also proves valuable in cases where a single person may not see a viable solution but someone else does and they are able to make up for each other's shortcomings as individuals.