Name: Daryan Chan

ID: 113973192

Section: NAA

Course: JAC444

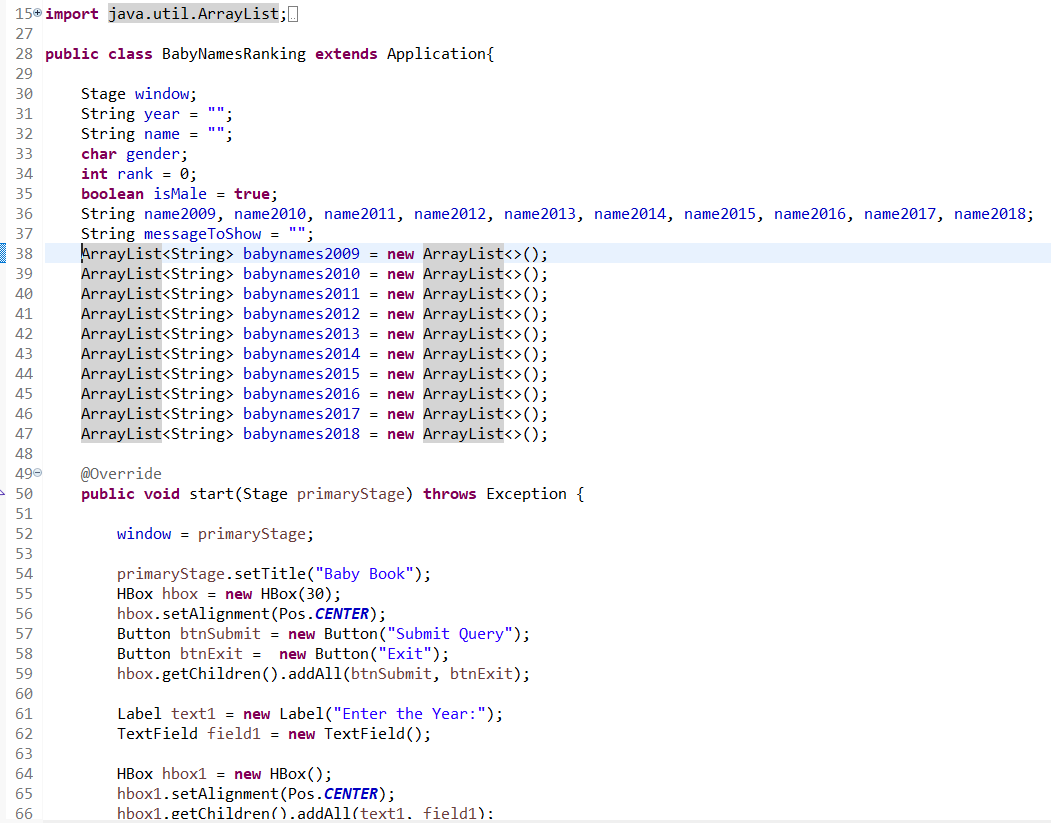


Figure BabyNakesRanking class

The first thing I did was to declare all the variables that I will use. Then I proceeded to create the GUI. I’ve used what I’ve learned from last workshop and applied it into this workshop. First I set the title as baby book. Then I created the buttons and arranged them in a HBox. Afterwards, I created a label that instructs the user what to enter into the TextField. Again, I arranged these nodes using the HBox.

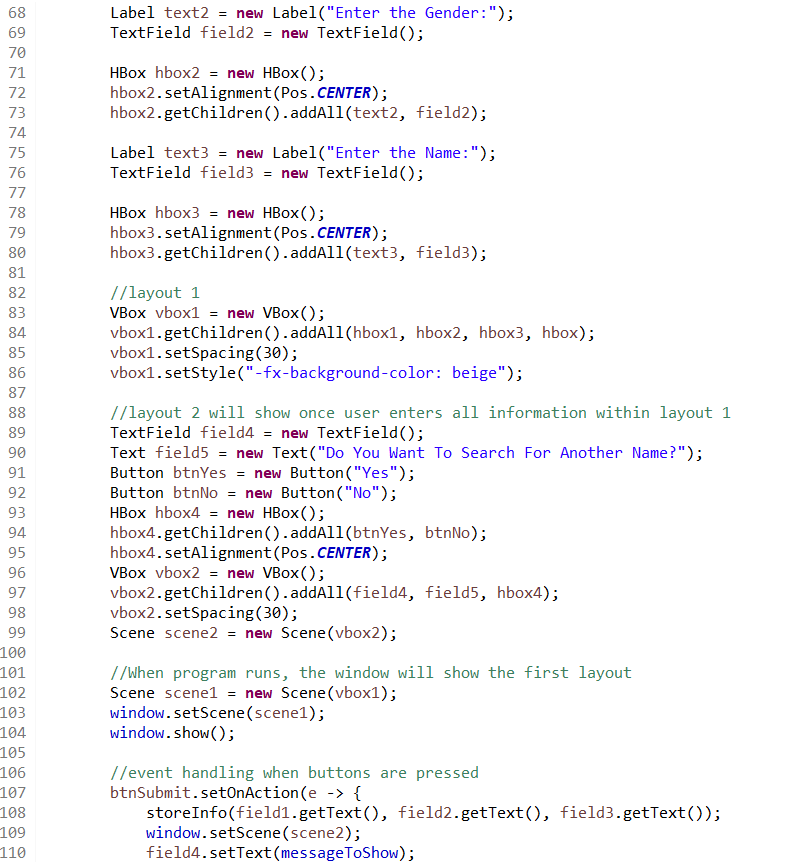


Figure BabyNamesRanking continuation

I proceeded to create other labels and TextFields to prompt the user what to enter. Then I placed the HBox I made into a VBox and designated it for the first layout. For the second layout, this will show a new Scene when the user entered the required info. The new layout consists of another TextField, which will show the result. I’ve also added two more buttons that will either search the ranking for another name or close the program.

Once those were done, I focused on the event handling. For the submit button, I made it so it will call the storeInfo function and pass along the data placed in the TextField by the user. Pressing this button also prompts for the new scene to activate and show the result to the aforementioned TextField.

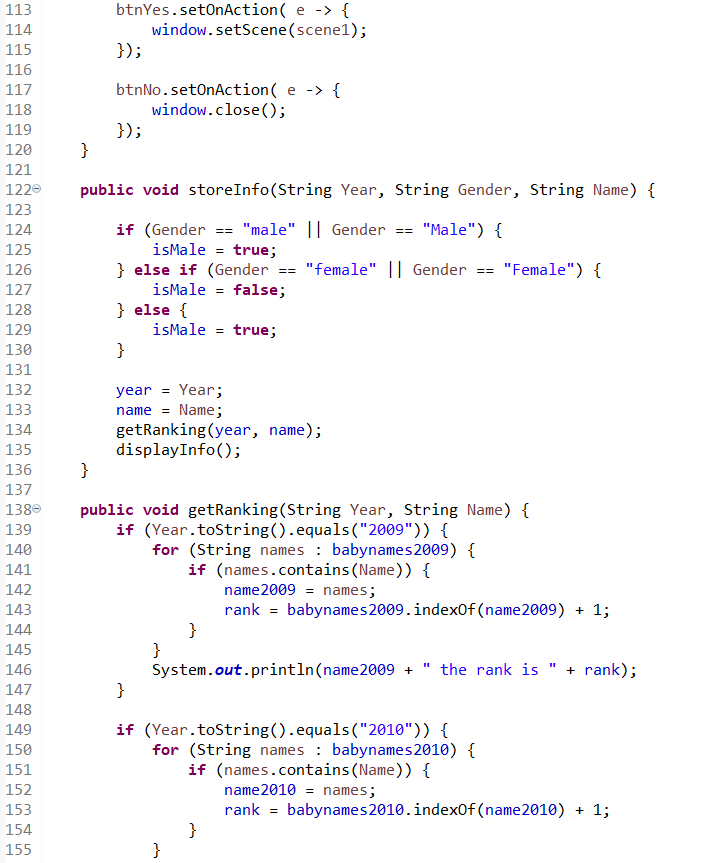


Figure BabyNamesRanking continuation

Afterwards, I have to set another event handling for the button in the second layout. When the user press the Yes button, it will go back to the first layout. If no is pressed, then the program ends.

The storeInfo method takes 3 parameters and basically saves those data. It determines if the input wants a male or female name, and then it calls another method and passes the year and name. In the getRanking method, it goes through many if conditions to see which year is exactly wanted. Once the year has been determined, it goes through a for loop and see if the name input by the user is contained within the array. If so, the name is copied to a name variable (ex: name2009) and also gets the index of said name. This is to get the ranking and I made sure to add one to it so the proper ranking is stored.



Figure BabyNamesRanking continuation

The method is explained above. One thing to note is that I added a System.out.println for each if condition for my personal use to see if the code is running properly.



Figure BabyNamesRanking Continuation

Code has the same logic explained above.

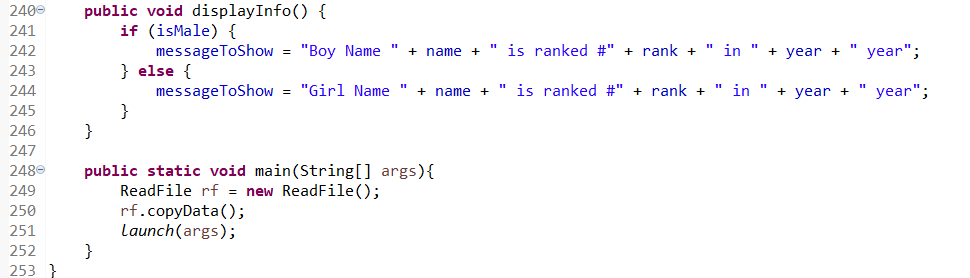


Figure BabyNamesRanking continuation

For the displayInfo method, it makes checks which gender is required. When that has been determined, then it copy the string with the appropriate information.

The main method basically calls the copyData method from another class to copy the data from the baby ranking name files. This will be explained in the next figure. The main method will also make sure the GUI is created.



Figure ReadFile class

This class has a copyData method. When called, it will go through many try and catch statements. This basically uses the scanner class and copies the info, line by line, from the textfile into arraylists.



Figure ReadFile class continuation

The rest of the code is explained above. One thing to note is the System.out.println. I added this to make sure all the data within the textfiles are being copied and so I can personally see the names when I later enter the name I want into my program.

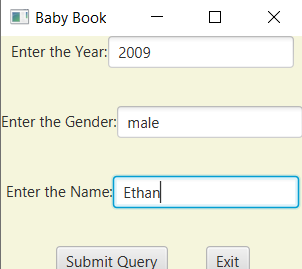


Figure Result

An example of the data being requested.

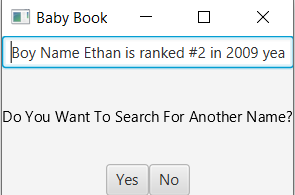


Figure Result

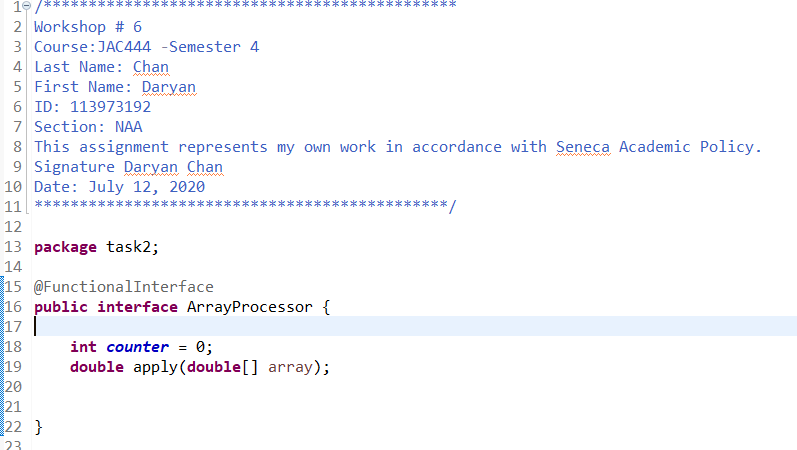


Figure ArrayProcessor interface

The interface here has the un-implemented method of apply, which will be used in the class that implements this interface.

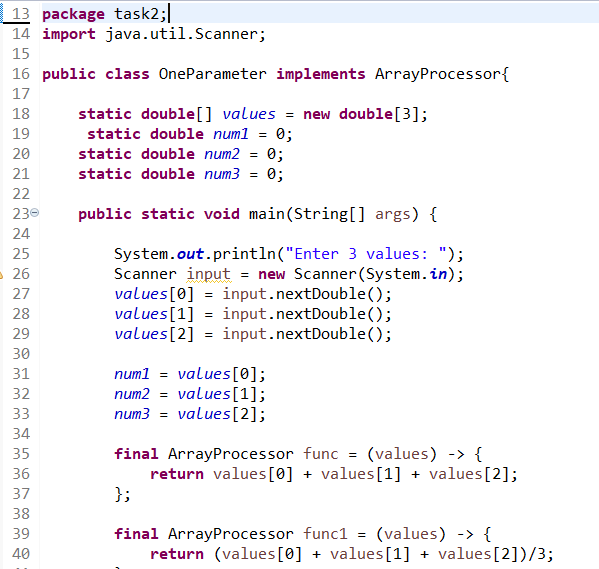


Figure OneParameter class

This shows the variables that will store input entered by the user. The main method will prompt the user to enter 3 values, which will then store the values in the variables. Then, lambda expresioned will be used to do various things, such as to add all the values, find the average, etc.

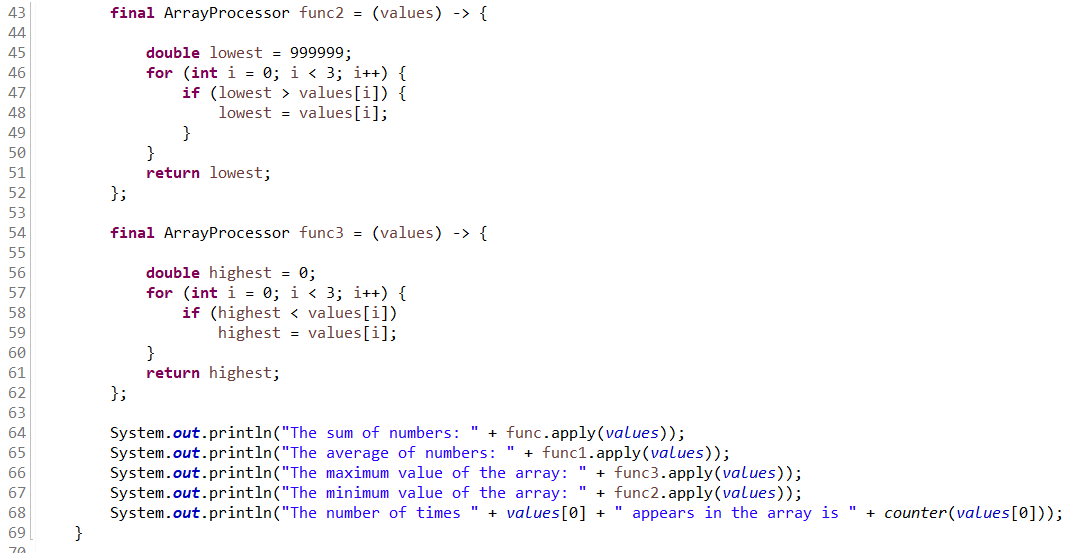


Figure OneParameter class continuation

The last two lambda expressions will find the max and min values of the array.

The System.outprintln will display the results for each lambda expressions.

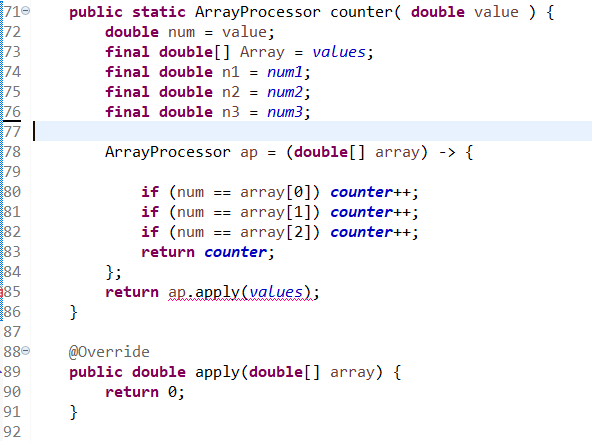


Figure OneParameter class continuation

Sadly, I couldn’t figure out these last parts. I tried to code how many times a value will appear in the array and I had trouble coming up with how to use the abstract method. Hopefully I won’t lose too many points for this part but I’ve watched many youtube tutorials and I’ve yet to figure out how to implement the desired result that the instruction wanted.

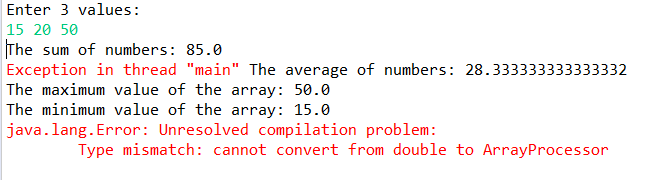


Figure Results

As you can see, it still compiled and the sum, average, max, and min were all calculated just fine. The only hiccup is that I couldn’t figure out how to see how many times a value reoccurs using the function shown in the instruction.