Name: Daryan Chan

ID: 113973192

Section: NAA

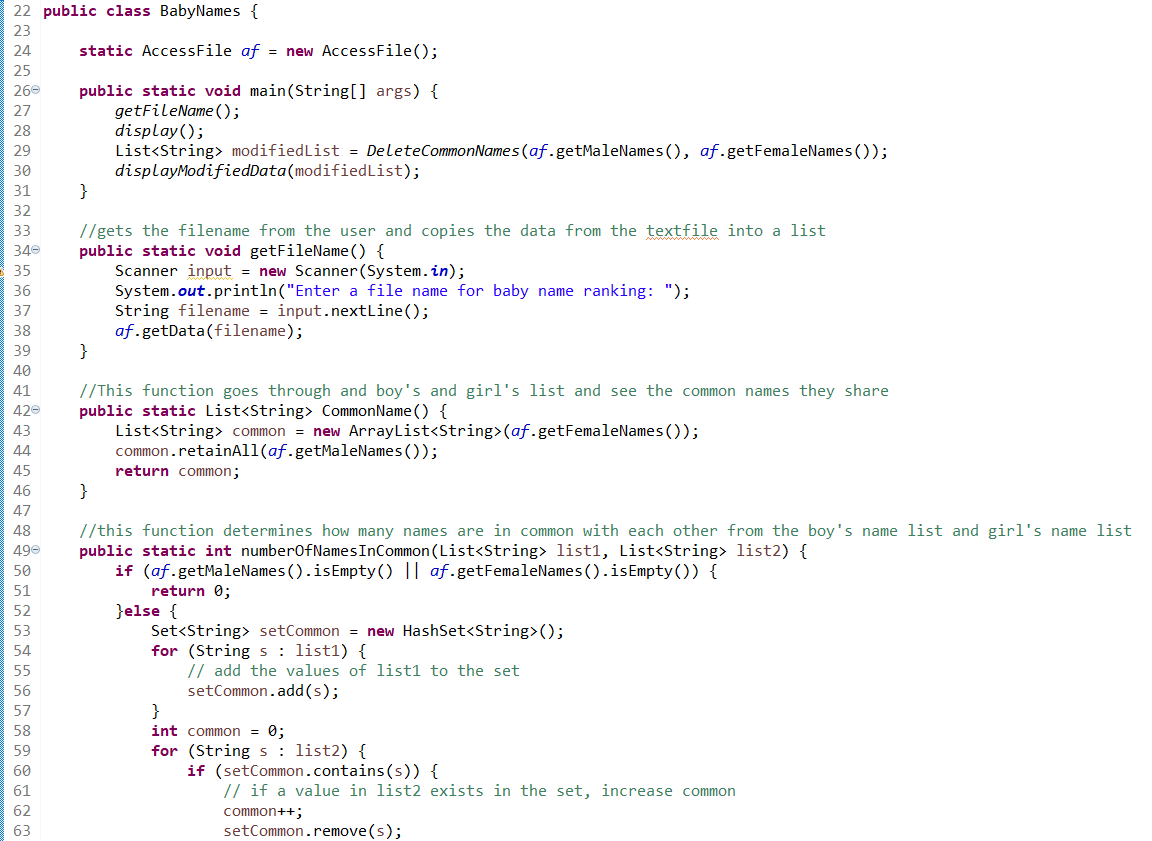


Figure BabyNames class

For this class, I created the main function that will call a method that will be used to call another function from another class after receiving the file name from the user. After that’s done, the display method will be called. This will be explained later. Afterwards, a List<String> modifiedList will receive the data from another function with 2 lists as its parameter. Finally, another display method will be called.



Figure BabyNames class continuation

For the display function, it will display multiple information and will utilize other functions to display the correct info. One of these is the numberOfNamesInCommon(). The purpose of this function is to compare the 2 lists in the parameter and see how many of the boy’s and girl’s name are unisex. If you look back at figure one, it determines if the lists are empty or not. If not it will create a new List and utilize the HashSet function to find all unique names. It basically adds all the names from the boy’s list and then going through the girl’s list, it will remove similar names. Afterwards, it will return the final list.

The commonName() method is to get the actual common names. For this function, it will make a new List<String> common and copies the girl’s name. Then, it will only keep the girl’s name if the guy’s name are the same.

The getmaleNames() and getFemaleNames() are basically getter functions and will get back the list of boy’s and girl’s names.

The displayModifiedData receives a modifiedList, as mentioned from figure one. This will be in charge of displaying individual names and reversing the order of the lists.



Figure AccessFile class

For this class, I made 2 separate lists. One is a maleNames list and the other is for femaleNames. The getData() method basically creates 2 more lists and the purpose of those lists are to get rid of unnecessary information within the textfiles. Afterwards, it compares the data input to see which textfile is required. When a match is found, it opens the file and copies the data using a try and catch method. Once that is done, only the required data (the names) are copied to the maleNames and femaleNames lists. This will be done by calling the setter functions.



Figure AccessFile class continuation

The logic is explained above.



Figure AccessFile class continuation

The logic is explained above.



Figure AccessFile class continuation

The logic is explained above.

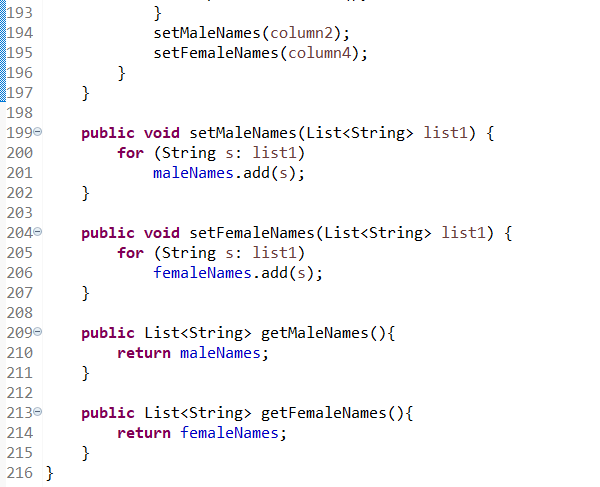


Figure AccessFile class continuation

The setter and getter methods basically set the data and retrieve the data.

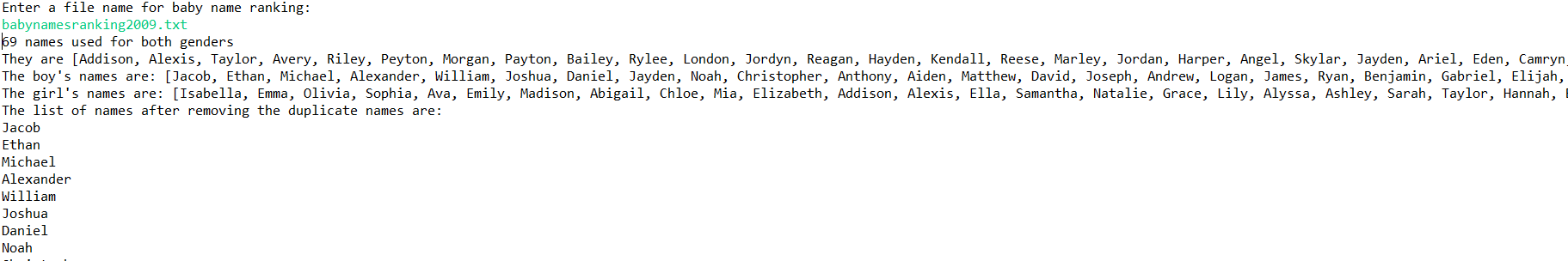
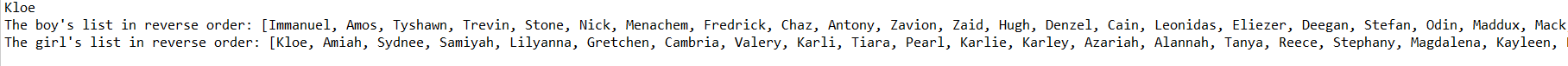


Figure Result

When the program runs, it will ask for the filename. When entered, it will get the number of times the names are the same. Then, it will display the boy’s and girl’s names that are similar, and then display all the boy’s and girl’s names. Afterwards, it will display the names individually. The list is too long so I won’t include all the screenshots.



9 Result continuation

Lastly, it will display the boy’s and girl’s name in reverse.



Figure Map class

For this task, I’ve taken the liberty and personalizing this assignment. I made some variables, (ex m\_name, timestamp, etc). Afterwards, I created a constructor that will be used to initialize my variables.

The setLog will basically copy the name to m\_name and set the time. Afterwards, it will call the start method again.

The getName() is a getter method and will get the m\_name.

The getCity is another getter method and will get the city name. it will receive a country and call another method, getCountryIndexPosition, to receive the index number of that city. It will store that number and use that index to retrieve the city from the city list. If the index is -1, it means that the country isn’t recorded.

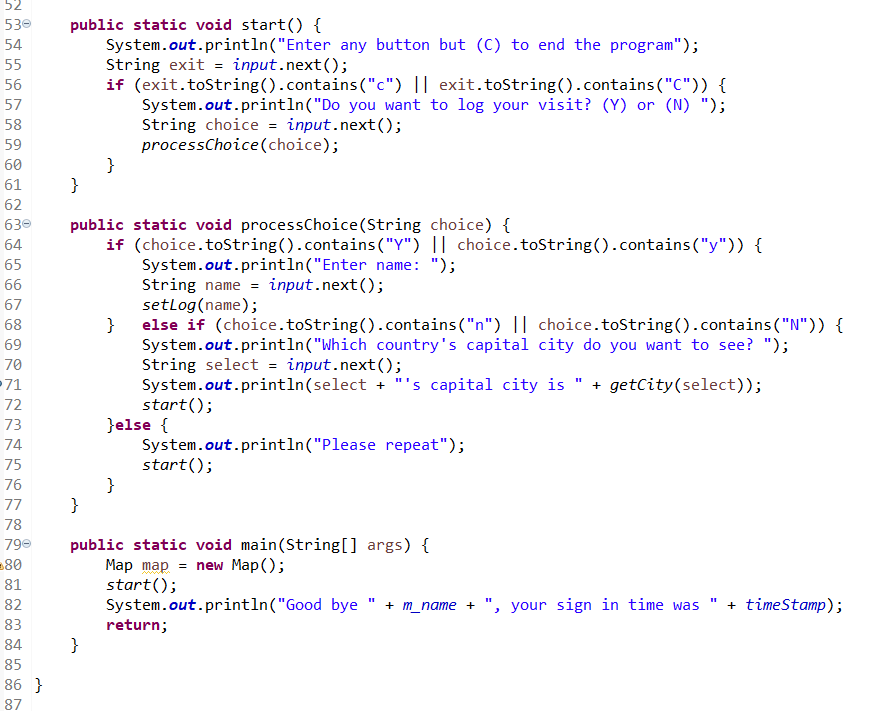


Figure Maps class continuation

The start() method will prompt the user to either being or exit the program. It will also ask the user if they want to log their visit. After they entered their choice, it will call another function, processChoice() and will include one parameter. The next method determines what the user enters and acts accordingly. If the user wants to log in their visit, it will set it, if not it will ask the user which city they want to see based on the country they want. Then, it will call the functions as explained from the previous figure. When the user enters they are done, it will display a good bye message and will include the name of the user as well as the time they signed in.

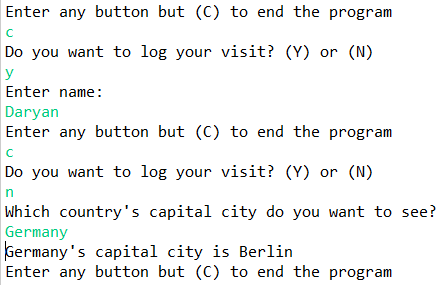


Figure Result

Result works as expected.