**JAC444 Workshop 8 – Walkthrough**

**Name: Kian Dadkhah Shokrollahi SID: 119369205**

**Task 1**

Program starts in the main of RandomNumList. A new object of type RandomNumList is initialized, which stores a list of random numbers inside the listOfRandom member. The list of random numbers is printed to the console window.

The program then displays the sum of all random numbers displayed by adding them together inside a loop (line 34).

Then the sum is divided by the number of random numbers (25) in order to display the average.

Text

Description automatically generated

**Task 2**

Program starts in the main of PrimeFactorsApp, which sets the stage and creates a JavaFX GridPane window. This window indicates to the user to enter a number for the program to check whether it is a prime number. It includes a label prompt, a text box, and a “Calculate” button.

Graphical user interface, application

Description automatically generated

Once a number is entered and the “Calculate” button is clicked, its Action Listener will create a new object (primeObj) of type CheckPrimeFactors and pass in the Integer-casted value of the text box as its constructor argument. This object will take this incoming number and decide whether it is prime (via checkIfPrime(int)). Then, the Action Listener calls on the setText property function of the factorsLabel Label object, and pass in the value returned from calling on the display() function of the primeObj object. This returned value will differ depending on the outcome of checking if the entered number is prime.

If the number is prime, a text will appear under the “Number:” label of the Prime Calculator window stating that the number is prime.

Ex. Let’s enter 7, a prime number.

Graphical user interface, application

Description automatically generated

If the number is not prime, the display() function will return the prime factors of the number entered, which is displayed similarly as a result.

Ex. Let’s enter 36, not a prime number. The prime factors of 36 are 2 x 2 x 3 x 3

Graphical user interface, application

Description automatically generated

**Task 3**

This program takes a static 2 dimensional array of Strings consisting of 1, country names, and 2, country capitals, and uses the Map Interface to create key value pairs. This allows a user to query for a given country (key) and see its corresponding capital city (value). The program starts in the main of the CountryCapital class. First it creates a new object of type CountryCapital. CountryCapital is a class which consists of a Map collection of two Strings called countriesCapital. It also has a private method called populateData() which inserts the static 2 dimensional array into the member Map object.

The main method uses a Map Entry iterator to loop through the countriesCapitals object and display each country available for the user to query. Then, a prompt is given for the user to enter a country name.

Text

Description automatically generated

Once a country name is entered, the uppercased value of the input is used to check if a value for the given key exists. If it exists, the value will be displayed as the capital city of the country entered, and the program will exit.

Ex. Let’s enter Austria as the country name.

Text

Description automatically generated