Objectives:

* Arrays

**There are 6 challenge exercises, each worth 16.6%**

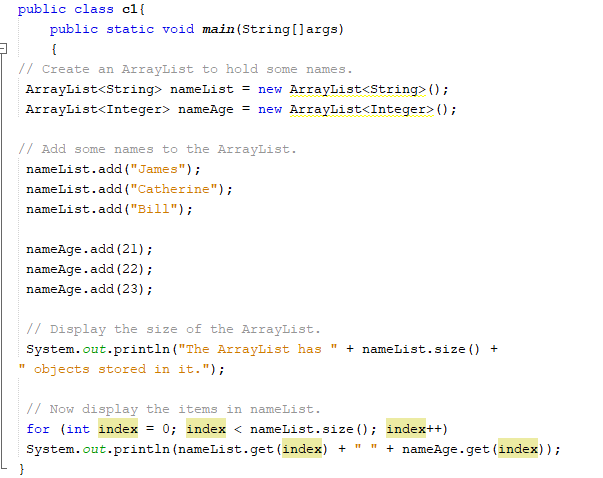
**Project #1 (Passing Arrays as Arguments)**

Graphical user interface, text, application

Description automatically generatedText

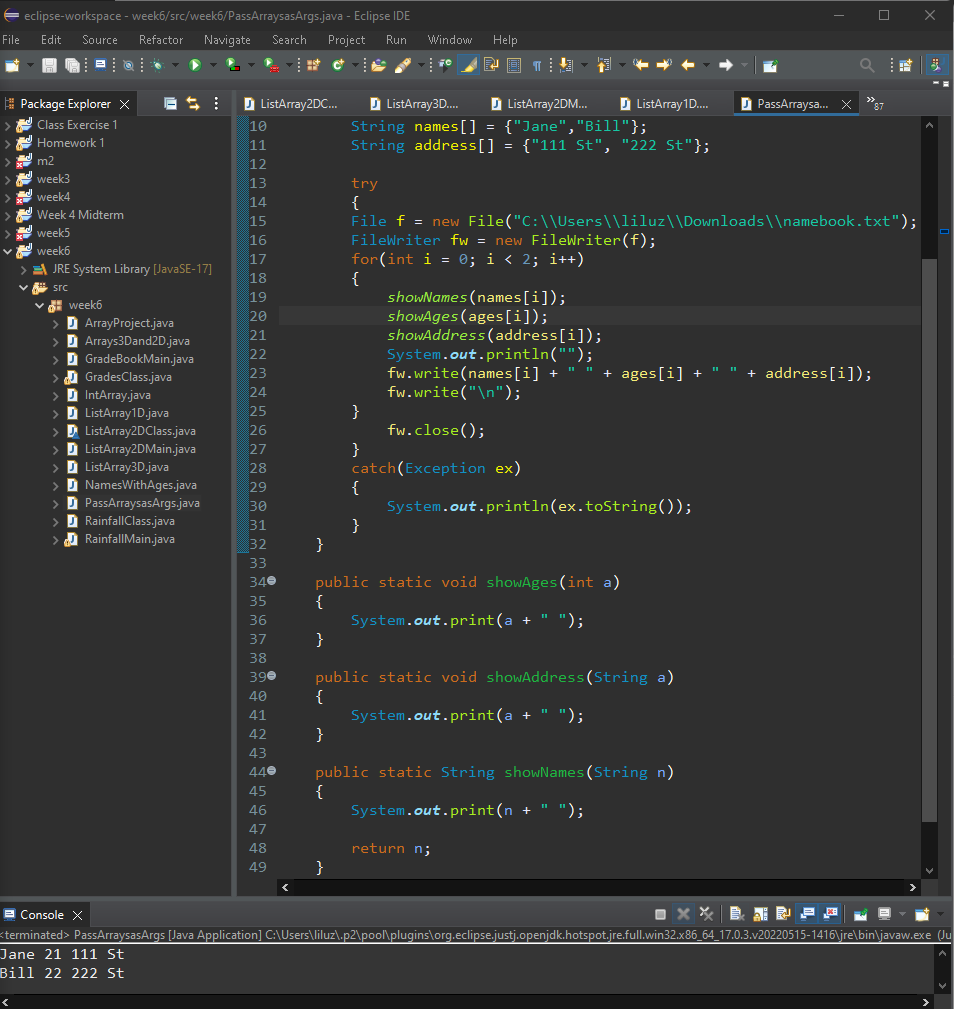
Description automatically generated

**Project #1 (A), Using an array list to add data to the Array list**

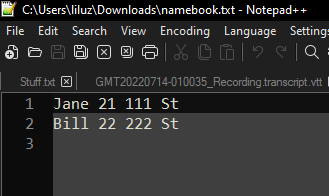


**Challenge Exercise #1:** continuing project #1, create a String array for addresses. Use the following addresses for the array >> 111 st and 222 st. Then have the output transfer all the data from the arrays into a text file.

**#1 print screen the output with code below here.**

****

**#2 open your text file and print screen it below here.**



**Project #2 (Using methods and arguments to pass arrays)**

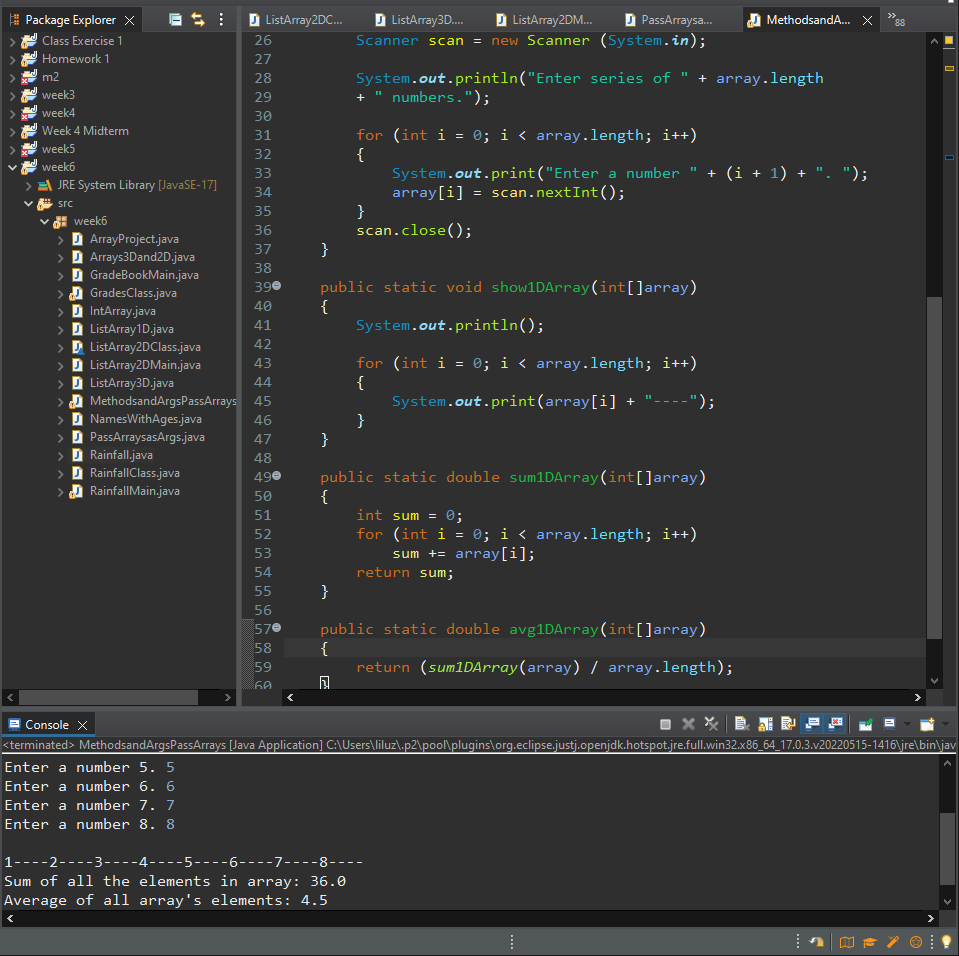
Graphical user interface, text, application

Description automatically generatedTimeline

Description automatically generated with low confidence

**Challenge Exercise #2:** add 4 more numbers into the array, with a total of 8 numbers. Then sum and average the numbers of the array.

**#3 print screen the output with code below here.**

****

**Project #3 (Comparing arrays)**

Text

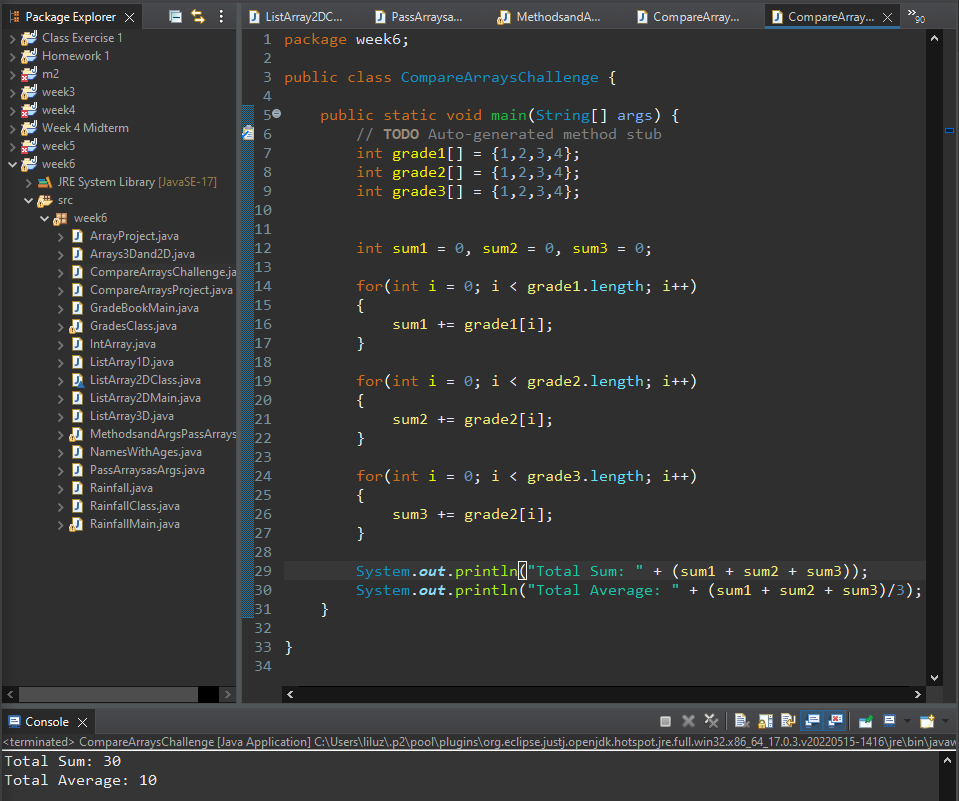
Description automatically generated with medium confidence

A picture containing table

Description automatically generated

**Challenge Exercise #3:** continuing with project #3, create a third array, with 4 numbers >> use the numbers 1,2,3 and 4 for all three arrays. Then total all the numbers from the three arrays, you should get the sum of **30** and the average from the three arrays. You should get the average of **10**.

**#4 print screen the output with the code below here.**

****

**Project #4 (Using Parallel Arrays)**

Text

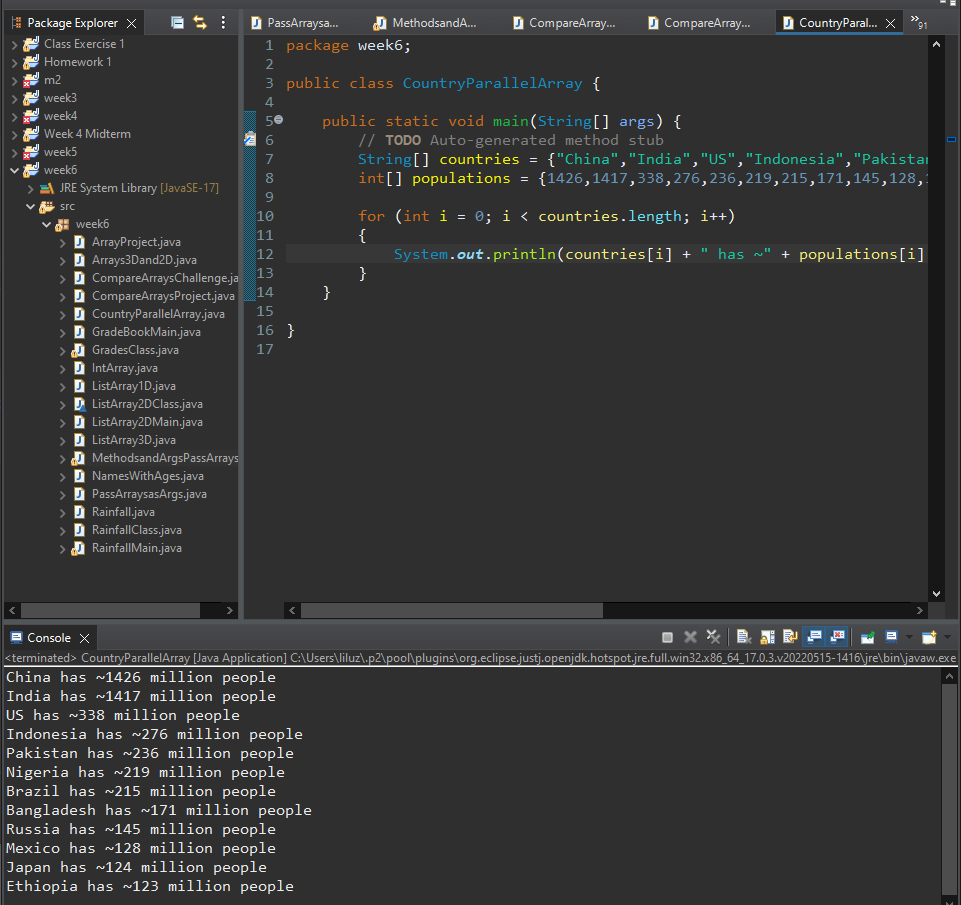
Description automatically generated with medium confidence

**Challenge Exercise #4:** complete the following parallel array program.

A picture containing bird

Description automatically generated

**#5 print screen the output with the code below here.**

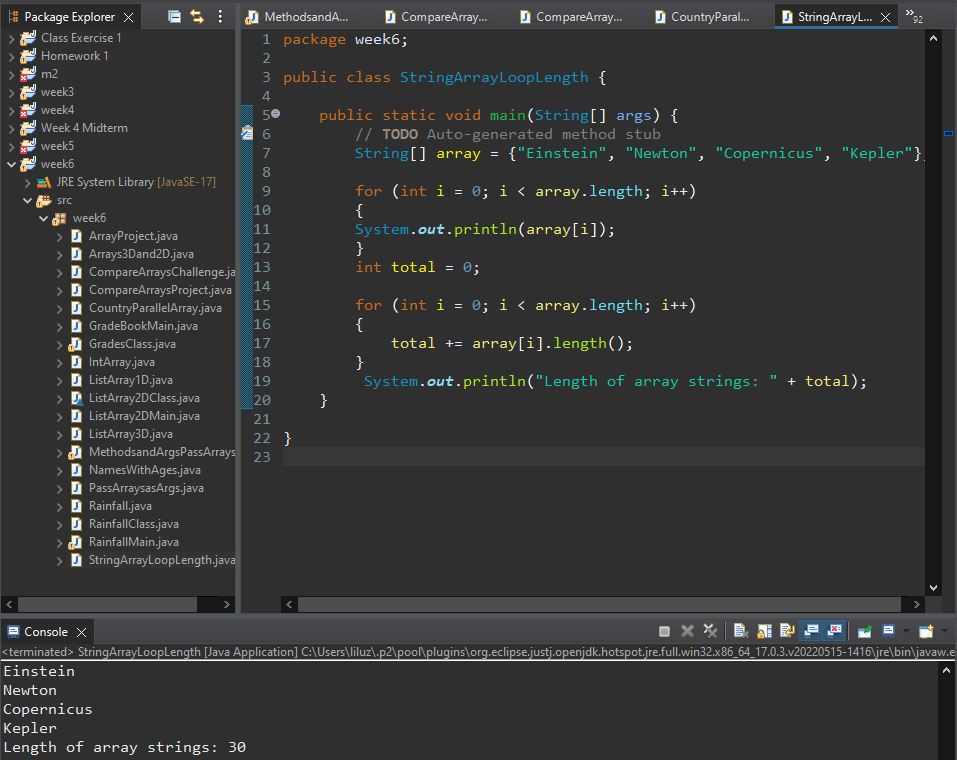


**Challenge Exercise #5:** Complete the following array program.

A close up of a sign

Description automatically generated

**#6 print screen the output with the code below here.**



**Challenge Exercise #6:** The following program will use two classes with methods, write the code from the two classes, and get the total and average of two numbers in the array. *Note: the program is not working, your task is to fix the program so it will output the correct total and average.*

Class #1 (SalesData)

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Class #2 (Main Method)**

Graphical user interface, text, application, email

Description automatically generated

**#7 print screen the output with the code below here.**

**Submit this document to module 6 class exercise.**