Java Salter Smooth Documentation ☺

The Java salter smoother can be found in the folder “Plotter Salter Smoother” and each part of it is contained within its own folder. Each folder describes what it is, like for example, “Plotter” which would contain the main code as well as the tester main method code.

This is just the code documentation with both text, screenshots and java doc comments. However, in the “Data reports” folder, you can find a report with all the generated data graphs from the .csv files.

If you would like to see the data and graphs within a spreadsheet, you can navigate to the “graphs w data (Java PSS)” folder.

If you haven’t already, please check out: https://byronproject2.carrd.co/

It serves as a fun little artistic project to present my strengths and weaknesses on this part of the project as well as organize everything into one place.

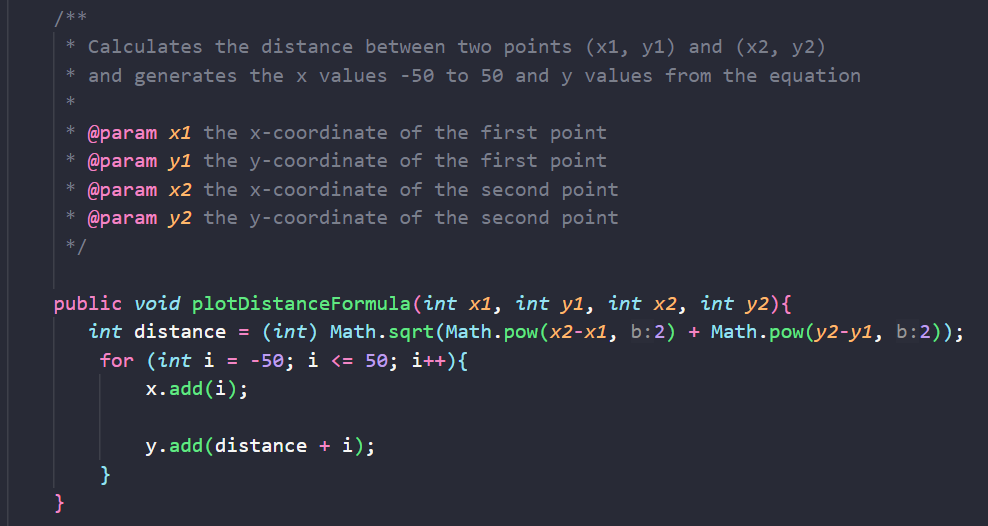
Within the plotter folder is plotter.java.

A screenshot of a computer program

Description automatically generated

So, the first method that was written was a simple method to plot the quadratic formula: “”

The for loop provides x values from -50 all the way to 50 (hence the ≤ sign), adding them all to the x value array list. All the values for y are created off the x values and the formula provided above. Math.pow was used here to take a value and put to an exponent. The rest of the formula is self-explanatory. This formula was simple enough to not need any parameters or extra fancy qualities.



The next method is called plotDistanceFormula(int , int , int , int )

The purpose is to plot the distance formula:

The math part was stored into an integer variable called distance. It takes both x values subtracted in order, squaring them and adds them to the squared y values. After this, Math.sqrt is used to square root the outcome. Once again, -50 to 50 were chosen here as the range for the x values. Why? Really high values were tried at first (-10,000 - 10,000…) and the graph did not graph. So, -50 to 50 looked better. Somehow, it was a lesson learned. The formula is used to generate the y values and i is added to the distance value.

A screenshot of a computer program

Description automatically generated

The next method is called PlotAbsoluteValue(int a).

The purpose is to plot the absolute value equation:

The same x value range is created here from -50 to 50 and added to the arraylist for it. Math.abs is used to generate y values from the formula shown above using the x values from i and the parameter a. The parameter a is a chosen value that can change how the graph looks, making it either stretched or compressed.

All the classes use two array lists for x and y values.

A group of colorful text

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated

The method name is writeToFile(String filename).

This method serves its purpose to write both the x and y values to file. It accepts the filename as a parameter (Example: “hello.csv”). Here we create a FileWriter object called file, letting it accept the filename parameter. There is also a printWriter object created to write to a new file. The size of the x array list is stored in a variable called numOfPoints. There’s a loop that will cycle through all the x values and then print them along with the y values. The printWriter is closed. There’s an IOException that works in case an error occurs.

Then there is the Salter.java class.

The salter method is as below.

A computer screen shot of a code

Description automatically generated

The salter method name is saltFormulas(int bound).

This salted method is used to salt all three equations which were plotted by the plotter methods above. The parameter is a bound that will be chosen 0 – that number, randomly. Then, that number will be subtracted by 100. The y values are then changed to these values which are produced.

A computer screen shot of a program code

Description automatically generated

The method name is readToFile(String filename)

The salter has the first instance of the readToFile method. It has the parameter filename (ex: “file.csv”). The purpose is to read data from files. BufferedReader and FileReader are used here together, to efficiently read the lines of the file. The while loop allows for the readers to understand where the x and y values are and the commas as well as adding the x and y values to variables, allowing them to be read by the file reader. The BufferedReader is then closed for efficiency and another IOException added in case of an error while reading the file.

The write to file method is used again to create salted csv files, but it can be seen above as it is the same as the one used to plotter.

Then we have the Smoother.java class.

A screenshot of a computer program

Description automatically generated

Here is the smoothData(int range) method.

It accepts the parameter range. The first for loop iterates through all the y values. The second for loop goes through indices within the range stated. It works to find the average for each element. The if statement works to make sure an index out of bounds error is not provoked. This method works to smooth the salted data.

The readtoFile and writeToFile method are reused again here in order to read the salted data and write the smoothed data to a csv file.