**Plotting, Salting, Smoothing 3 (Java Libraries)**

Tutorials used: <https://www.youtube.com/watch?v=9VlZh0B6DE4>

<https://www.codejava.net/java-se/graphics/using-jfreechart-to-draw-xy-line-chart-with-xydataset>

**Plotting, Salting, and Smoothing Functions:**

The functions in this program for plotting, salting, and smoothing operate using the same logic as the ones from part 1, however instead of the x and y values being stored in ArrayLists, and then those values being outputted to a csv file, the x and y values are stored in XYDatasets, and those datasets are used with JFreeChart to create JPEG images of the graphs

**XYDatasets:**

A black background with green and yellow text

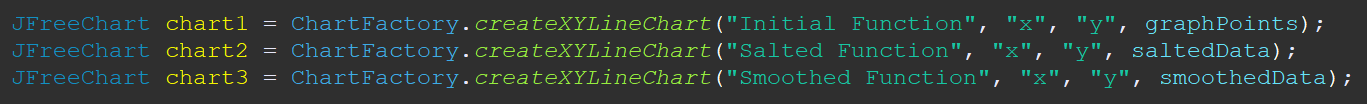
Description automatically generatedAn XYDataset is an interface which can store values in the form of (x, y). Each dataset can also store multiple series of (x, y) values for plotting multiple function on one graph, which are called XYSeries. For the purposes of this program however, my XYDatasets only contain one series. To add the data to my XYDatasets, I create an XYSeries where I add each individual point, then I add that XYSeries to the corresponding XYDataset instance variable.

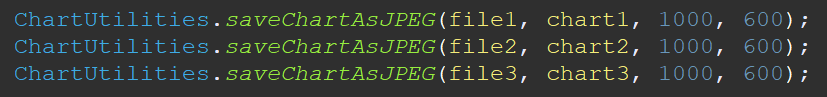
A black background with white text

Description automatically generated Used in my calculateFunction method ^

**JFreeCharts:**

In order to create the graphs using JFreeChart, I imported ChartFactory and ChartUtilities. From ChartFactory I called the createXYLineChart method, which creates the chart using an XYDataset as one of the parameters. Then from ChartUtilities I called the saveChartAsJPEG method, which saves the chart as a JPEG image.





**Final Output:**

**A graph of a function

Description automatically generated**

A graph showing a function

Description automatically generated

A graph showing a graph of a function

Description automatically generated with medium confidence

**Note:** I did import Apache Stats Library into the project, but I didn’t use it for any of my functions because 1) the only place where I thought it could really be used was in calculating the average for the smoothing method, and 2) the way that Apache has the getMean() function implemented would’ve taken more time to use than just calculating it normally. From what I was able to gather, inside of the descriptive statistics library there is a class called “DescriptiveStatistics” that has the getMean() method, but it doesn’t take any parameters. What you need to do is create an object of DescriptiveStatistics using a constructor that passes an Array as a parameter, and then getMean() will calculate the mean of that Array.