Rocco Vulpis

December 14, 2023

Plotting Functions using Java and Excel

In the plotting portion of this project, a mathematical function was created and adjusted by user specified parameters. Its data points were then exported to a CSV file. From there, a graph of the data was created within Microsoft Excel displaying the results of the plotted data points. The graph of the data resembles the graph of the function based on the parameters input by the user. In this case, the function that was used was the equation of a line expressed in slope intercept form which is *y = mx + b.*

The first step was to create the slope function. This could be done by calling and passing parameters to the createSlopeFunction method. Its job is to calculate the y values of the linear function for a set of x values within a specified range and interval. The necessary parameters are m (the slope of the linear function), b (the y intercept of the linear function), xRangeBeginning (the starting value of the x range), xRangeEnd (the ending value of the x range), and numberOfPoints (the number of points to calculate in the specified range).

After x and y values were calculated and stored in their respective arrays, they would then be passed to the writeToFile method. This method takes in two parameters, the array of x values and the array of y values. It uses the BufferedWriter class to write both the x and y values to a CSV file named “data.CSV”. When the program runs, this file is generated and stored in the default workspace directory. Using excel to open the CSV, the list of x values and y values that were written are stored in columns A and B. Using Excel’s built in charts, a scatter plot can be graphed from the given data. The graph of the data should resemble the graph of the function just as it would on a graphing calculator.

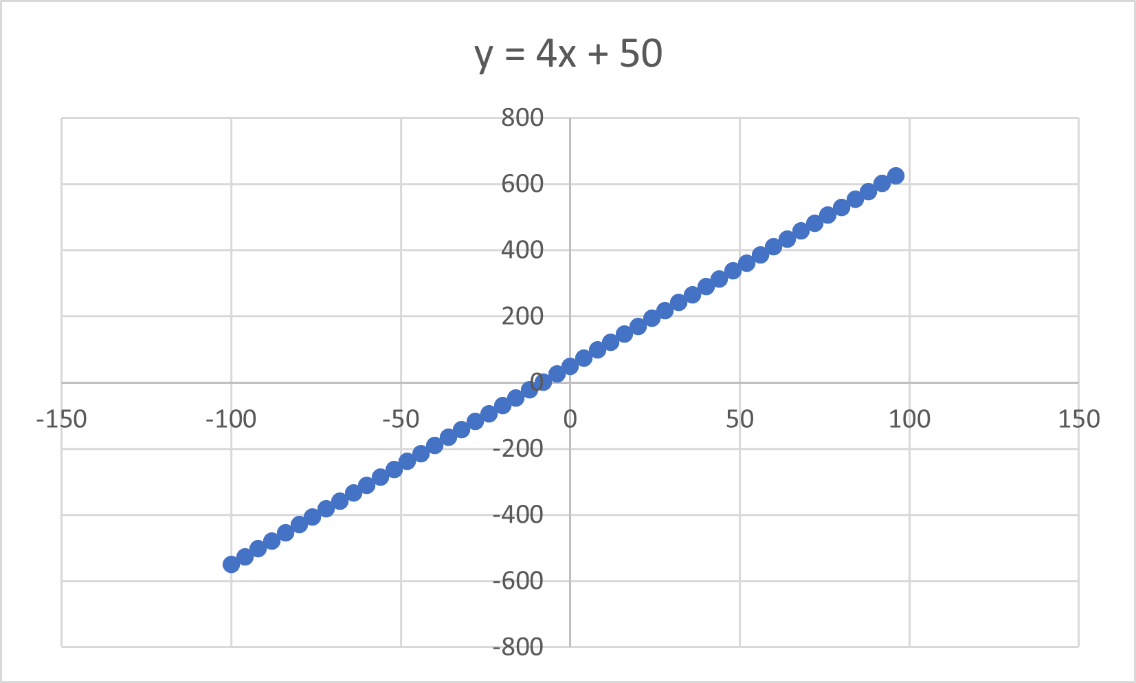
The parameters of the createSlopeFunction make it so that the graph of the function can be altered. For the sake of example, the parameters being used are y = 2, b = 3, xRangeBeginning = -100, xRangeEnd = 100, and numberOfPoints = 50. Increasing the m value makes the graph of the points appear steeper based on the calculation of the y values. Since the slope of the line is 2, each point will increase up two steps and to the right one step. Increasing the b value will adjust the y intercept. If b is set to a positive value, the y intercept will be plotted above zero on the y axis. If b is set to a negative value, the y intercept will be plotted below zero on the y axis. Because b = 3, the points will cross the y axis at y = 3. The starting x value and ending x value can be set to give a window of x values to plot between. xRangeBeginning and xRangeEnd are set to -100 and 100, so the line will be plotted between x values -100 and 100. The last parameter is numberOfPoints. Adjusting this value to 50 will plot 50 points between the given x range which in this case is -100 and 100. Below is a screenshot displaying the graph using the specified parameters created in Excel.

A graph of a line

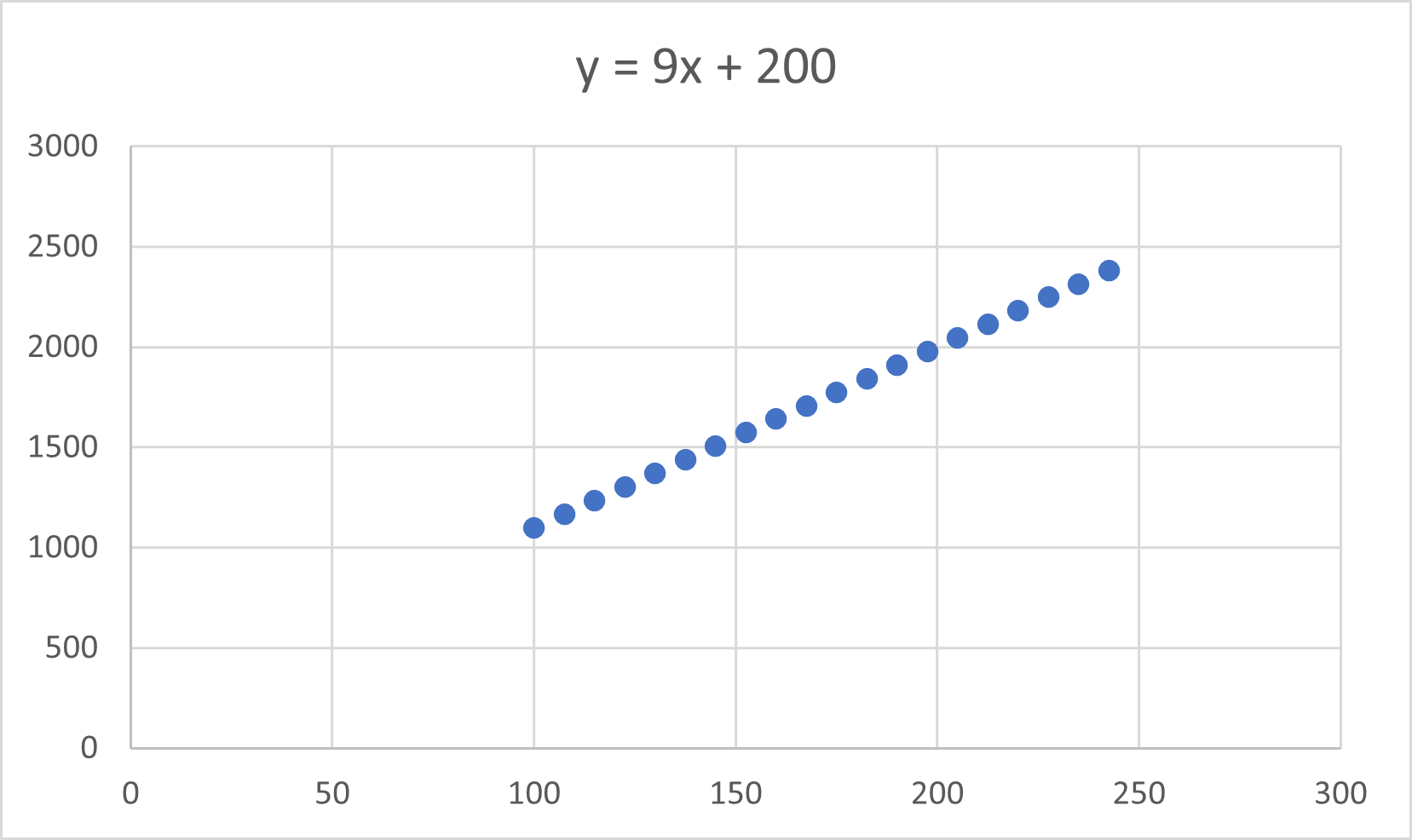
Description automatically generated

y = 2x + 3 where x is bounded between -100 and 100 with 50 points plotted.

Some more examples of graphs using different parameters.



y = 2x + 50 where x is bounded between -100 and 100 with 50 plotted points.



y = 9x + 200 where x is bounded between 100 and 250 with 20 plotted points.