PSS1 Report

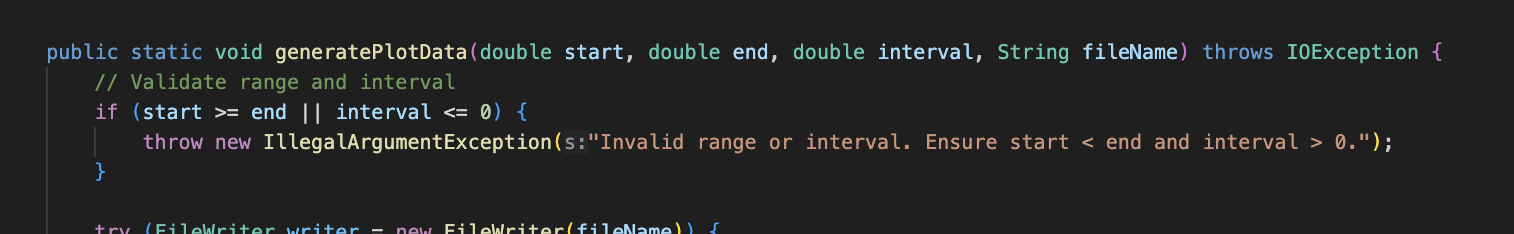
Taszid Chowdhury

**Project Overview**

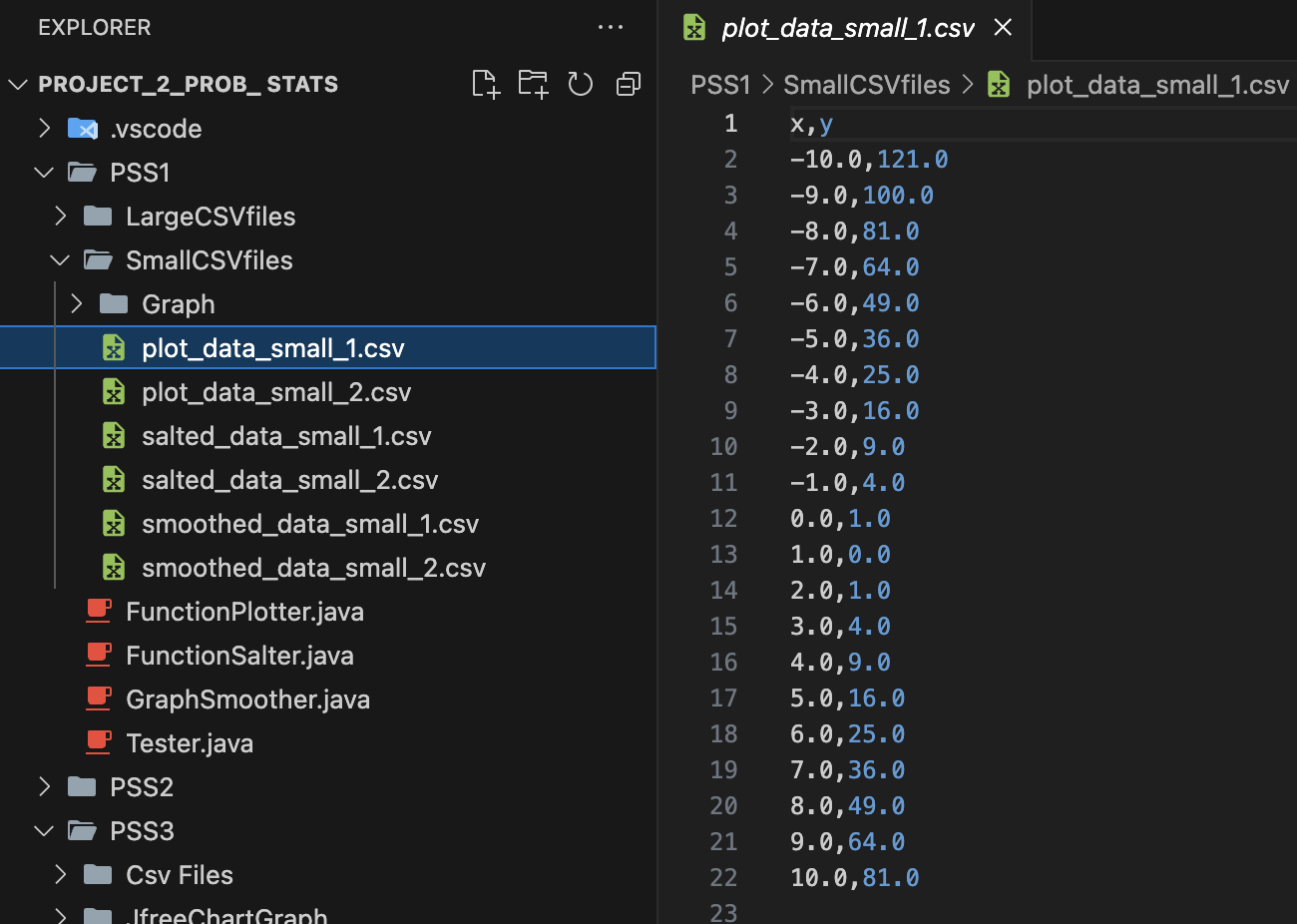
1. This project contains Java programs to:
2. Generate and save mathematical function plots as CSV files.
3. Introduce random noise ("salt") to the data in the files.
4. Smooth the noisy data using a moving average technique.
5. Test the above functionalities with different configurations.

### **Files and Functionalities**

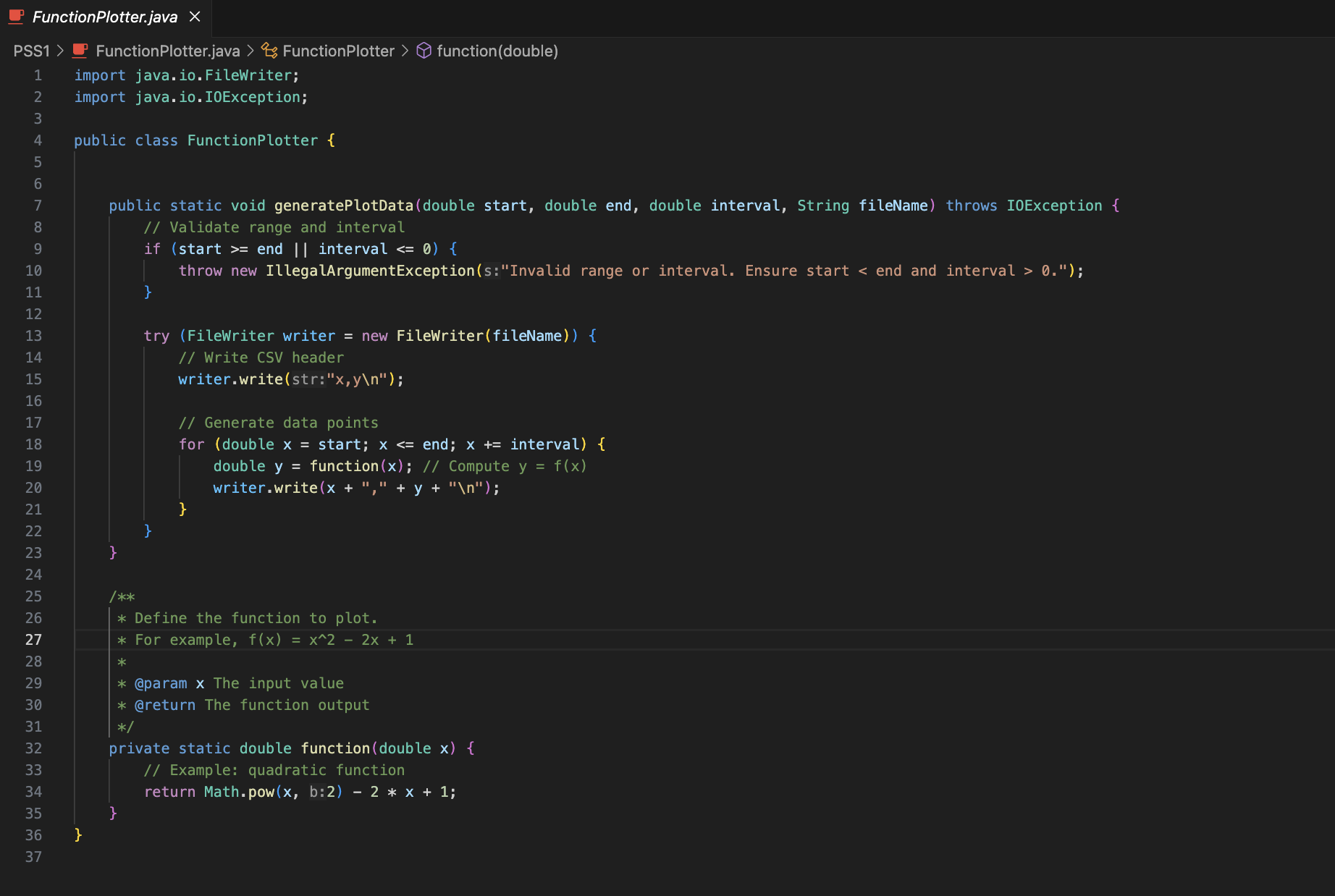
1. FunctionPlotter.java
   1. **Purpose:** Generates CSV data for a mathematical function, with configurable range and interval
   2. **Key Features:**
      1. Method generatePlotData:
         1. Accepts a start value, end value, interval, and output file name.
         2. Validates input to ensure:
            1. start < end
            2. interval > 0
         3. Generates function data points using a predefined function.
         4. Writes output in CSV format with headers: x, y.



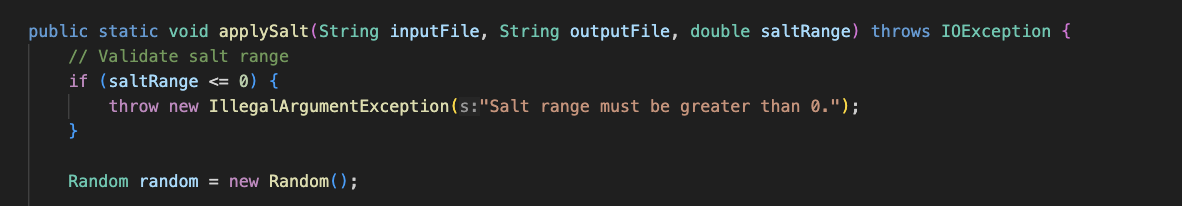
* 1. **Function Definition:**
     1. Currently defined as (a quadratic function).
  2. **Example Output File:** A CSV file with rows representing xxx and yyy values.



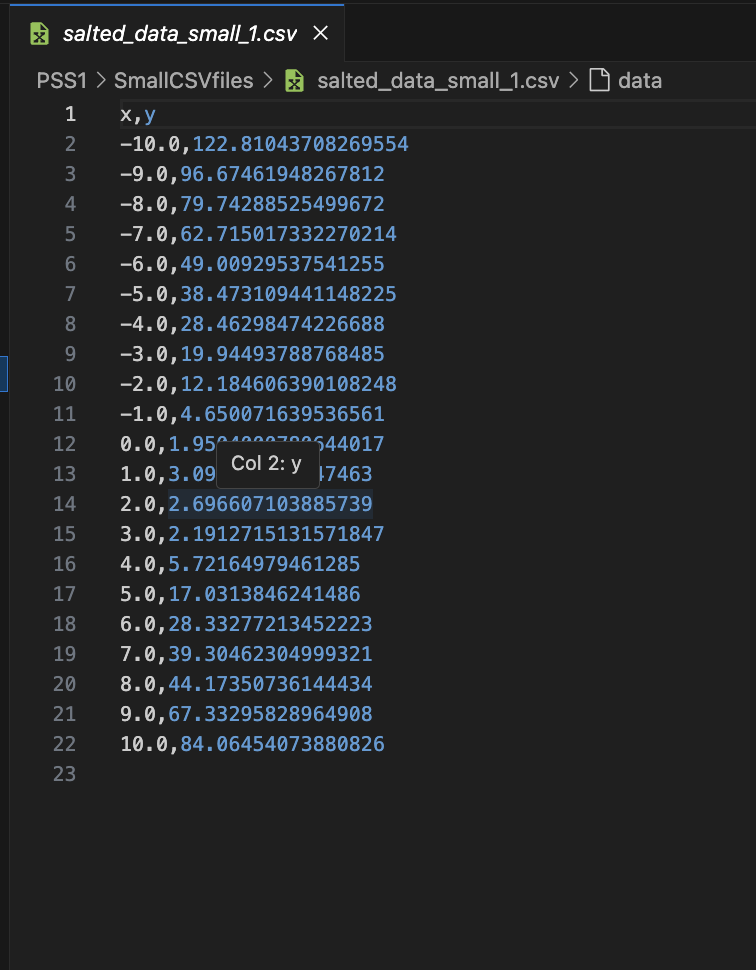
Code ScreenShot



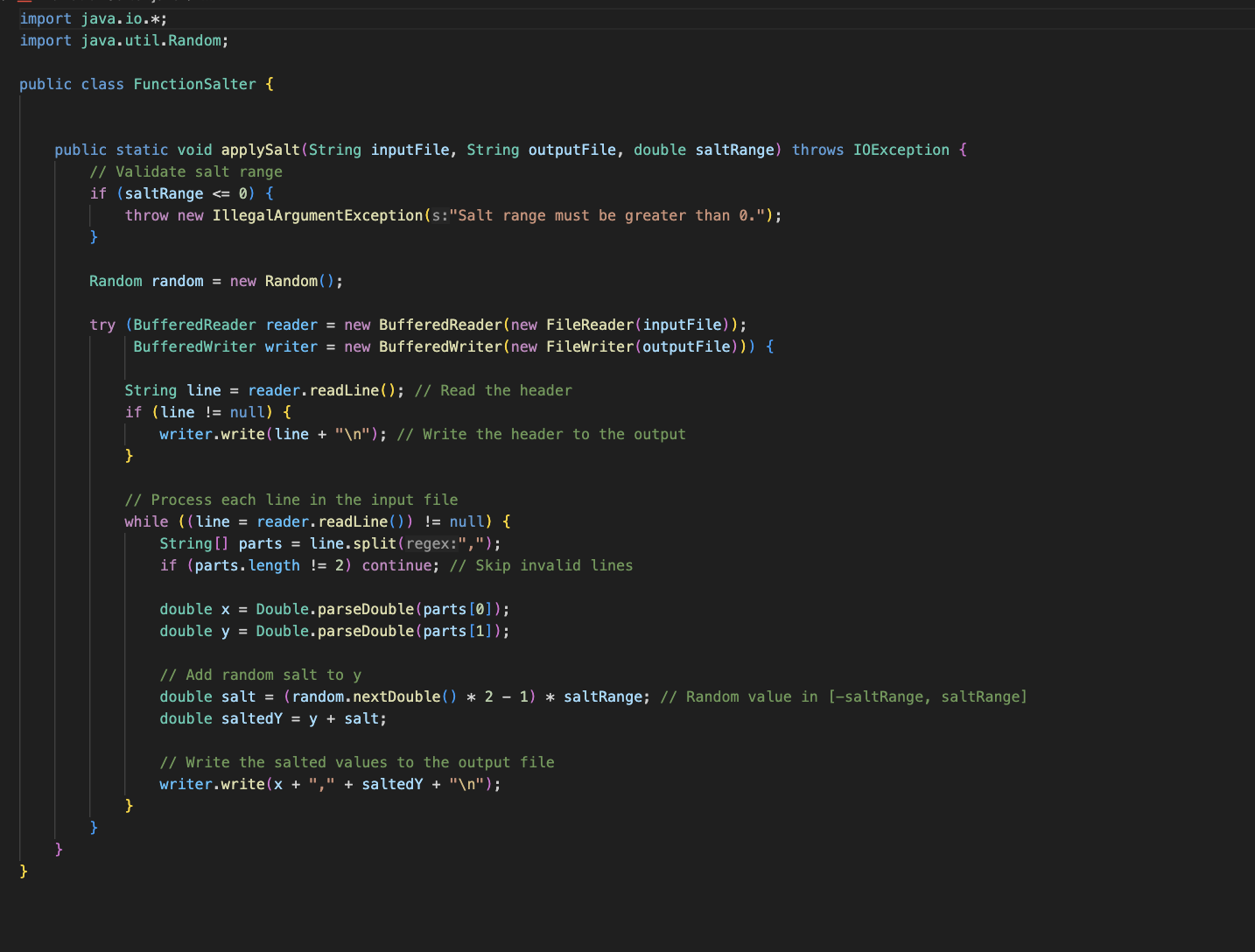
1. **FunctionSalter.java**
   1. **Purpose:** Applies random noise (salt) to the y-values in a given CSV file.
   2. **Key Features:**
      1. Method applySalt:
         1. Reads an input file and adds random noise to y-values within a specified range (saltRange).



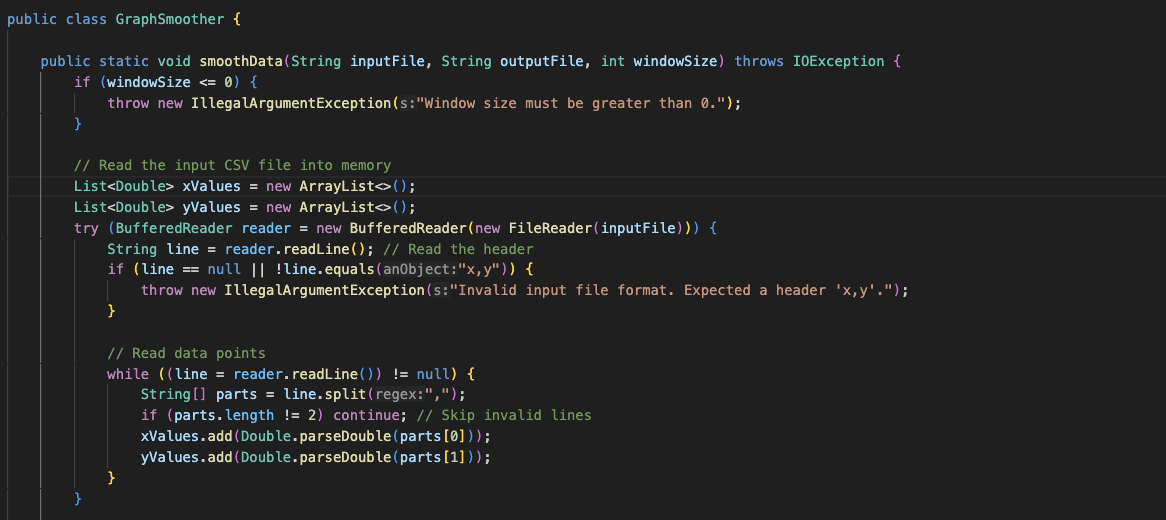
* + 1. Validates input to ensure
       1. saltRange > 0
       2. Input file contains correctly formatted rows (x, y).
    2. Writes salted data to a new output file.
    3. Randomization Logic:
       1. Noise is a random value in the rang [−saltRange,+saltRange][-saltRange, +saltRange][−saltRange,+saltRange].



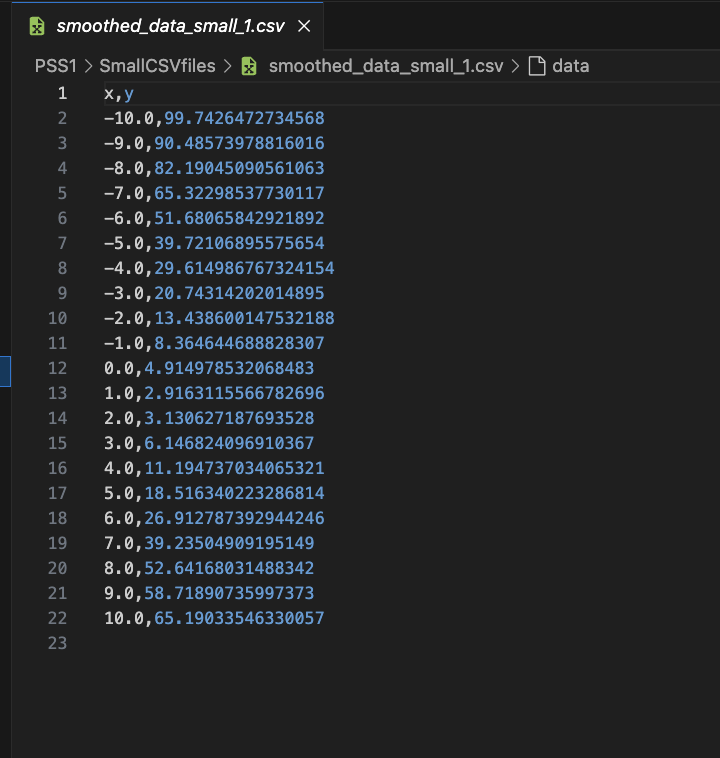
SALTED code screenshot

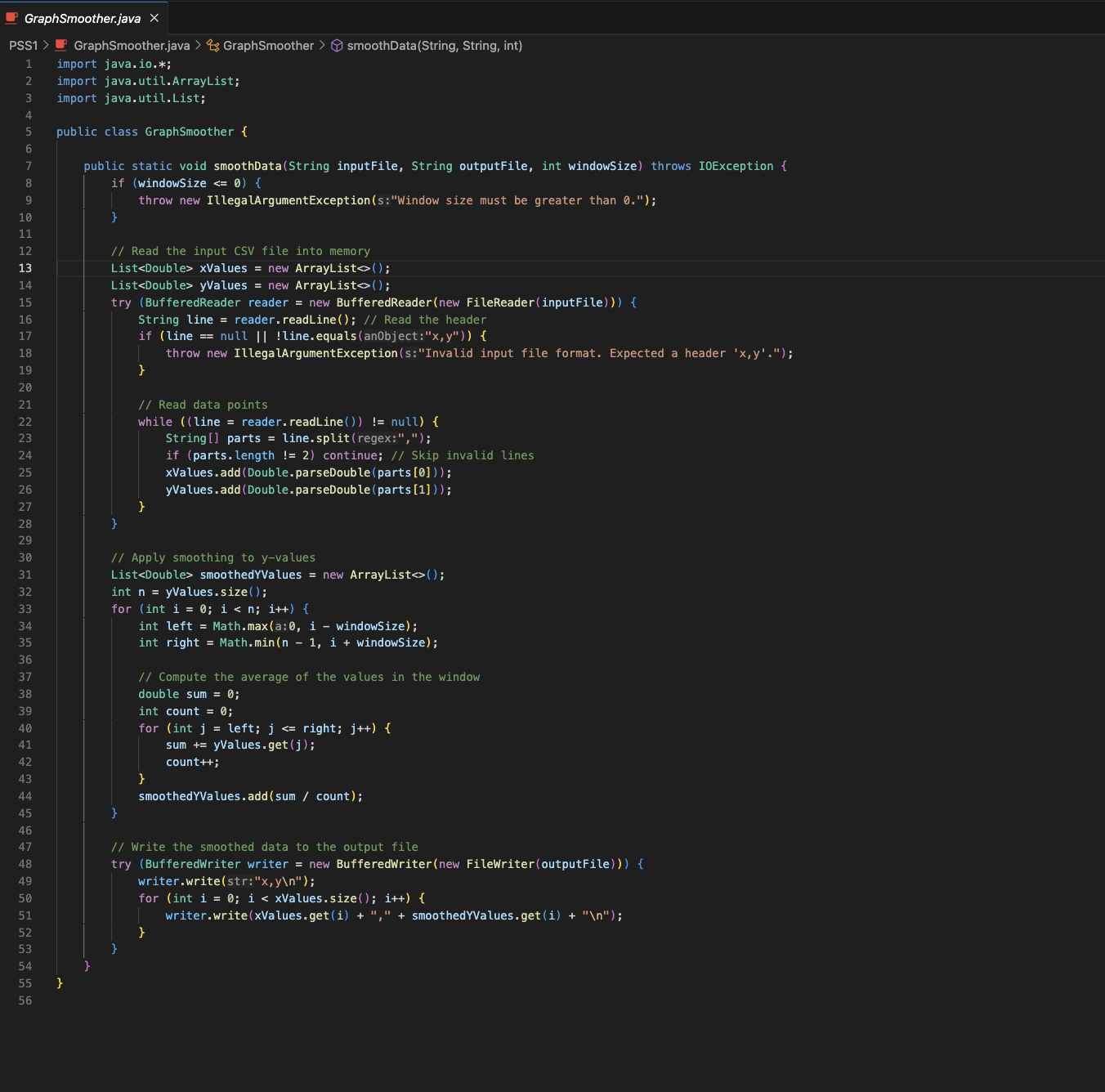


1. **GraphSmoother.java**
   1. **Purpose:** Smoothens the noisy y-values using a moving average technique.
   2. **Key Features:**
      1. Method smoothData:
         1. Reads an input CSV file and performs smoothing using a window of size windowSize.



* + - 1. Validates input:
         1. File format should include a header x, y.
         2. windowSize > 0.
      2. Computes average y-values in a sliding window around each data point.
      3. Outputs smoothed data to a new CSV file.
      4. Example Use Case: Creating a clearer trend line from noisy data.





SMOOTHER code screenshot

1. **Tester.java** 
   1. Purpose: Tests the functionalities of FunctionPlotter, FunctionSalter, and GraphSmoother with various configurations.
   2. Key Features:
      1. Defines specific test cases with different ranges, intervals, salt levels, and smoothing window sizes.
      2. Logs success or error messages to the console.
   3. Testing Scenarios:
      1. Plot data for small and large ranges with varying densities.
      2. Apply salt with low, medium, and high volatility to generated datasets.
      3. Smooth datasets using small to large window sizes.

Tester Code



Tester Output

