

PlotHandler.java

```
1 //*****
2 //
3 // File:    PlotHandler.java
4 // Package: ---
5 // Unit:    Class PlotHandler
6 //
7 //*****
8
9 import java.io.File;
10 import java.io.IOException;
11 import java.text.DecimalFormat;
12 import edu.rit.numeric.ListXYSeries;
13 import edu.rit.numeric.plot.Plot;
14 import edu.rit.numeric.plot.Strokes;
15 import edu.rit.util.AList;
16
17 /**
18  * Class PlotHandler is the delegate for dealing with visualizing the data
19  * generated by the "number crunching" program, SimulationStation.
20  * Its purpose is to be instantiated in SimulationStation with the data to plot,
21  * where the write() method should then be called.
22  *
23  * Running this program and specifying in the command line arguments the plot
24  * files previously generated will open a graphical representation of these
25  * plots for each file.
26  *
27  * @author Jimi Ford
28  * @version 4-4-2015
29  *
30  */
31 public class PlotHandler {
32
33     // private data members
34     private final String averagePowerFile;
35     private final String probabilityFile;
36     private final AList<SimulationResult> results;
37
38     /**
39      * Construct a new plot handler that plots average distances for a fixed
40      * vertex count v, while varying the edge probability p
41      *
42      * @param plotFilePrefix prefix to be used in the name of
43      *        the plot file
44      * @param results collection of results of the finished set of
45      *        simulations.
46      */
47     public PlotHandler(String plotFilePrefix,
48         AList<SimulationResult> results) {
49         averagePowerFile = plotFilePrefix + "-average-power.dwg";
50         probabilityFile = plotFilePrefix + "-probability-connected.dwg";
51         this.results = results;
52     }
53
54     /**
55      * Save the plot information into a file to visualize by running
56      * the main method of this class
57      *
58      * @throws IOException if it can't write to the file specified
```

```

59  */
60  public void write() throws IOException {
61      ListXYSeries averagePowerSeries = new ListXYSeries();
62      ListXYSeries probabilitySeries = new ListXYSeries();
63      SimulationResult result = null;
64      for(int i = 0; i < this.results.size(); i++) {
65          result = results.get(i);
66          if(!Double.isNaN(result.averagePower))
67              averagePowerSeries.add(result.v, result.averagePower);
68          if(!Double.isNaN(result.percentConnected))
69              probabilitySeries.add(result.v, result.percentConnected);
70      }
71
72      Plot powerPlot = new Plot()
73          .plotTitle ("Average Power vs. Number of Nodes")
74          .xAxisTitle ("Number of Nodes <I>V</I>")
75          .xAxisTickFormat(new DecimalFormat("0"))
76          .yAxisTitle ("Average Power Needed")
77          .yAxisTickFormat (new DecimalFormat ("0E0"))
78          .seriesDots(null)
79          .seriesStroke (Strokes.solid(2))
80          .xySeries (averagePowerSeries);
81      Plot.write(powerPlot, new File(averagePowerFile));
82      Plot probabilityPlot = new Plot()
83          .plotTitle ("Percent Connected vs. Number of Nodes")
84          .xAxisTitle ("Number of Nodes <I>V</I>")
85          .xAxisTickFormat(new DecimalFormat("0"))
86          .yAxisTitle ("Percent Connected")
87          .yAxisTickFormat (new DecimalFormat ("0.0"))
88          .seriesDots(null)
89          .seriesStroke (Strokes.solid(2))
90          .xySeries (probabilitySeries);
91      Plot.write(probabilityPlot, new File(probabilityFile));
92  }
93
94  /**
95   * Open a GUI for each plot in order to visualize the results of a
96   * previously run set of simulations.
97   *
98   * @param args each plot file generated that you wish to visualize
99   */
100 public static void main(String args[]) {
101     if(args.length < 1) {
102         System.err.println("Must specify at least 1 plot file.");
103         usage();
104     }
105
106     for(int i = 0; i < args.length; i++) {
107         try {
108             Plot plot = Plot.read(args[i]);
109             plot.getFrame().setVisible(true);
110         } catch (ClassNotFoundException e) {
111             System.err.println("Could not deserialize " + args[i]);
112         } catch (IOException e) {
113             System.err.println("Could not open " + args[i]);
114         } catch (IllegalArgumentException e) {
115             System.err.println("Error in file " + args[i]);
116         }
117     }

```

PlotHandler.java

```
117     }
118 }
119
120 /**
121  * Print the usage message for this program and gracefully exit.
122  */
123 private static void usage() {
124     System.err.println("usage: java PlotHandler <plot-file-1> "+
125         "<plot-file-2> <plot-file-3>... etc.");
126     System.exit(1);
127 }
128 }
129
```