

SimulationStation.java

```

1 //*****
2 //
3 // File:    SimulationStation.java
4 // Package: ---
5 // Unit:    Class SimulationStation
6 //
7 //*****
8
9 import java.io.IOException;
10 import edu.rit.pj2.Task;
11 import edu.rit.util.AList;
12
13 /**
14  * Class runs a number of trials simulating a network of space stations
15  *
16  * @author Jimi Ford (jhf3617)
17  * @version 4-2-2015
18  */
19 public class SimulationStation extends Task {
20
21     // java pj2 SimulationStation
22     // [0] = <lower_bound_stations>
23     // [1] = <upper_bound_stations>
24     // [2] = <increment_stations>
25     // [3] = <seed>
26     // [4] = <file_prefix>
27
28     private static final int
29         LOWER_INDEX = 0,
30         UPPER_INDEX = 1,
31         INCREMENT_INDEX = 2,
32         TRIALS_INDEX = 3,
33         SEED_INDEX = 4,
34         FILE_INDEX = 5;
35
36     /**
37      * main method
38      * @param args command line arguments
39      */
40     public void main(String[] args) {
41         if(args.length != 6) {
42             usage();
43         }
44         int lowerBound = 0, upperBound = 0, increment = 0, trials = 0;
45         long seed = 0;
46         String filePrefix = null;
47         try {
48             lowerBound = Integer.parseInt(args[LOWER_INDEX]);
49             upperBound = Integer.parseInt(args[UPPER_INDEX]);
50             increment = Integer.parseInt(args[INCREMENT_INDEX]);
51             trials = Integer.parseInt(args[TRIALS_INDEX]);
52             seed = Long.parseLong(args[SEED_INDEX]);
53             filePrefix = args[FILE_INDEX];
54         } catch (NumberFormatException e) {
55             error("Detected non-numeric input where expected numeric value");
56         }
57         AList<SimulationResult> results = new AList<SimulationResult>();
58         for(int vertices = lowerBound; vertices <= upperBound;

```

SimulationStation.java

```

59         vertices += increment) {
60         results.addLast(
61             new Simulator(this, vertices, trials, seed).simulate());
62     }
63     try {
64         new PlotHandler(filePrefix, results).write();
65         new TableHandler(filePrefix, results).write();
66     } catch (IOException e) {
67         error("Error writing results file(s) using prefix: " + filePrefix);
68     }
69 }
70
71 /**
72  * print usage statement and gracefully exit
73  */
74 private static void usage() {
75     System.err.println("java pj2 SimulationStation "
76         + "<lower_bound_stations> "
77         + "<upper_bound_stations> "
78         + "<increment_stations> "
79         + "<num_trials> "
80         + "<seed> "
81         + "<file_prefix>");
82     System.exit(1);
83 }
84
85 /**
86  * print an error message and call the usage() method
87  * @param msg the error message to print
88  */
89 private static void error(String msg) {
90     System.err.println(msg);
91     usage();
92 }
93 }
94

```