

## Simulator.java

```
1 //*****
2 //
3 // File:    Simulator.java
4 // Package: ---
5 // Unit:    Class Simulator
6 //
7 //*****
8
9 import edu.rit.pj2.Loop;
10 import edu.rit.pj2.Task;
11 import edu.rit.pj2.vbl.DoubleVbl;
12 import edu.rit.pj2.vbl.IntVbl;
13 import edu.rit.util.Random;
14
15 /**
16  * Class is responsible for the majority of the runtime of the program. It
17  * generates the given number of networks in parallel by utilizing Prof. Alan
18  * Kaminsky's PJ2 library.
19  *
20  * @author Jimi Ford (jhf3617)
21  * @version 4-4-2015
22  */
23 public class Simulator {
24
25     private Task ref;
26     private int v;
27     private int trials;
28     private long seed;
29     private IntVbl.Sum countConnected;
30     private DoubleVbl.Mean averagePower;
31
32     /**
33      * Construct a Simulator
34      * @param ref the reference to the main task - necessary for utilizing the
35      *         class's parallelFor method
36      * @param v number of space stations (or nodes in the graph)
37      * @param trials the number of random networks to generate
38      * @param seed seed value for the PRNG used in instantiated classes
39      */
40     public Simulator(Task ref, int v, int trials, long seed) {
41         this.ref = ref;
42         this.v = v;
43         this.trials = trials;
44         this.seed = seed;
45         countConnected = new IntVbl.Sum();
46         averagePower = new DoubleVbl.Mean();
47     }
48
49     /**
50      * Run all <TT>trials<TT>
51      * @return a SimulationResult containing the findings of the given number of
52      *         simulations
53      */
54     public SimulationResult simulate() {
55         ref.parallelFor(0, trials - 1).exec(new Loop() {
56
57             Random prng;
58             DoubleVbl.Mean thrAverage;
```

```

59     IntVbl.Sum thrCount;
60
61     // (Non-Javadoc)
62     public void start() {
63         prng = new Random(seed + rank());
64         thrAverage = threadLocal(averagePower);
65         thrCount = threadLocal(countConnected);
66     }
67
68     // (Non-Javadoc)
69     public void run(int i) throws Exception {
70         SpaceNetwork sn = new SpaceNetwork(prng, v);
71         if(sn.isConnected()) {
72             thrCount.item++;
73         }
74         sn.accumulatePower(thrAverage);
75     }
76 });
77 return new SimulationResult(
78     v,
79     trials,
80     countConnected.intValue(),
81     averagePower.doubleValue());
82 }
83 }
84

```