

```

1 package xyz.amtstl.soup.output;
2
3 import java.io.FileWriter;
10
11 public class HTMLGen {
12     /**
13      * Total outputs cache
14      */
15     public static List<Float> totalOutputs = new ArrayList<Float>();
16
17     /**
18      * Total Question Strings cache
19      */
20     public static List<String> questionStrings = new ArrayList<String>();
21
22     /**
23      * Generates HTML documentation for the Soup session
24      * @param title title on the window
25      * @param description the header of the page
26      * @throws IOException throws if it cannot store
27      */
28     public static void generateOutputDocumentation(String title, String description) throws
IOException {
29         try {
30             FileWriter f = new FileWriter(System.getProperty("user.dir") + "\\\" +
"SoupNoodle.html");
31
32             f.write("<style>h1 {    color: blue;}p {    color: red;}div {    display:
flex;    align-items: center;    justify-content: center;    flex-direction: column;}
</style><title>");
33             f.write(title);
34             f.write("</title><center><h1>");
35             f.write(description + " [" + LocalTime.now().getHour() + ":" +
LocalTime.now().getMinute() + "]");
36             f.write("</h1></center><div><p>Local Vars</p><ul>");
37             writeVars(f);
38             f.write("</ul><p>Outputs</p><ul>");
39             writeOutputs(f);
40             f.write("</ul></div><div><p>Questions Asked</p><ul>");
41             writeQuestions(f);
42             f.write("</ul></div>");
43             f.close();
44             System.out.println("Output Generated at " + System.getProperty("user.dir") + "\\\" +
"SoupNoodle.html");
45         }
46         catch (IOException ex) {
47             System.out.println("Error storing output documentation at " +
System.getProperty("user.dir") + "\\\" + "SoupNoodle.html");
48         }
49     }
50
51     /**
52      * Writes variables from cache
53      * @param j current file writer instance
54      * @throws IOException throws if error storing file
55      */
56     private static void writeVars(FileWriter j) throws IOException {

```

HTMLGen.java

```
57     for (float a : VariableHandler.intVars) {
58         if (a != 0.0)
59             j.write("<li>" + String.valueOf(a) + "</li>");
60     }
61 }
62
63 /**
64  * Writes all the outputs
65  * @param j current file writer instance
66  * @throws IOException throws if error storing file
67  */
68 private static void writeOutputs(FileWriter j) throws IOException {
69     for (float a : totalOutputs) {
70         j.write("<li>" + String.valueOf(a) + "</li>");
71     }
72 }
73
74 /**
75  * Writes all the session's questions
76  * @param j current file writer instance
77  * @throws IOException throws if error storing file
78  */
79 private static void writeQuestions(FileWriter j) throws IOException {
80     for (String a : questionStrings) {
81         j.write("<li>" + a + "</li>");
82     }
83 }
84 }
```