```java
import java.util.Comparator;

import components.map.Map;
import components.map.Map3;
import components.simplereader.SimpleReader;
import components.simplereader.SimpleReader1L;
import components.simplewriter.SimpleWriter;
import components.simplewriter.SimpleWriter1L;

/**
 * Simple HelloWorld program (clear of Checkstyle and SpotBugs warnings).
 *
 * @author Yunlong Zhang
 */
public final class WordCunter {

    public static void main(String[] args) {

        SimpleReader in = new SimpleReader1L();
        SimpleWriter out = new SimpleWriter1L();

        out.println("Please enter the file name: ");
        String fileName = in.nextLine();
        out.println("Please enter the output name: ");
        String outPutName = in.nextLine();
        // code above to get file name and output name from user.

        SimpleReader inFile = new SimpleReader1L(fileName);
        SimpleWriter outFile = new SimpleWriter1L(outPutName);
        // code above is about read user's file and output a new file

        Map<String, Integer> result = new Map3<>(new stringCompare());

        outPutHeader(outFile, fileName);
        // code above to print html header.
        buildMap(result, inFile);
        // code above to set a new map.
        bodyPart(result, outFile);
        // code above to print body part.
        outPutFooter(outFile);
        // code above to print footer

        in.close();
        out.close();
        inFile.close();
        outFile.close();
    }

    public static void outPutHeader(SimpleWriter out, String fileName) {

        out.println("<html>");
        out.println("<head>");
        out.println("<title>" + "Words Counted in " + fileName + "</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h2>" + "Words Counted in " + fileName + "</h2>");
        out.println("<hr />");
        out.println("<table border=\"1\">");
        out.println("<tr>");
```

```
60          out.println("<th>Words</th>");
61          out.println("<th>Counts</th>");
62          out.println("</tr>");
63          // code above to print header
64
65      }
66
67      public static void outPutFooter(SimpleWriter out) {
68          out.println("</table>");
69          out.println("</body>");
70          out.println("</html>");
71          // code above to print footer
72
73      }
74
75      public static void buildMap(Map<String, Integer> comb, SimpleReader in) {
76
77          while (!in.atEOS()) {
78              String line = " " + in.nextLine().replaceAll("\\p{Punct}", " ");
79              // code above to remove all punctuation from line.
80              String[] words = line.split(" ");
81              // code above to split line to single word based on " ".
82              for (String x : words) {
83                  if (!comb.hasKey(x)) {
84                      comb.add(x, 1);
85                  } else {
86                      comb.replaceValue(x, comb.value(x) + 1);
87                  }
88                  // code above to add word to map if the word not exist in map.
89                  // if the word is already exist in map, count +1;
90              }
91          }
92          comb.remove("");
93          // code above to remove empty.
94      }
95
96      private static class stringCompare implements Comparator<String> {
97          @Override
98          public int compare(String o1, String o2) {
99              return o1.compareToIgnoreCase(o2);
100         }
101         // code above to make strings's order in Alphabetic order.
102     }
103
104     public static void bodyPart(Map<String, Integer> comb, SimpleWriter out) {
105         for (Map.Pair<String, Integer> x : comb) {
106             out.println("<tr>");
107             out.println("<td>" + x.key() + "</td>");
108             out.println("<td>" + x.value() + "</td>");
109             out.println("</tr>");
110         }
111         // code above to print body part.
112
113     }
114
115 }
116
```