projects\01-word-counter\src\wordCounter.java

```java
import components.map.Map;
import components.map.Map2;
import components.sequence.Sequence;
import components.sequence.Sequence1L;
import components.simplereader.SimpleReader;
import components.simplereader.SimpleReader1L;
import components.simplewriter.SimpleWriter;
import components.simplewriter.SimpleWriter1L;

/*
 * @author Emerson Schnipke
 */

public class wordCounter {

    public static String[] separateWords(String fileName) {
        SimpleReader inFile = new SimpleReader1L(fileName);

        String fileContent = inFile.nextLine();

        while (!inFile.atEOS()) {
            fileContent = fileContent + inFile.nextLine() + " ";
        }

        inFile.close();

        String[] words = fileContent.split("[,\\.\\s\\;\\-]");

        return words;
    }

    public static Sequence<String> arrayToSortedSeq(String[] array) {

        Sequence<String> sortedSequence = new Sequence1L<String>();

        for (int i = 0; i < array.length; i++) {
            int j = 0;
            while (j < sortedSequence.length() && ((array[i]
                    .compareToIgnoreCase(sortedSequence.entry(j)) > 0)
                    || (array[i]
                            .compareToIgnoreCase(sortedSequence.entry(j)) == 0
                            && array[i]
                                    .compareTo(sortedSequence.entry(j)) < 0))) {
                j++;
            }
            if (!array[i].isEmpty()) {
                sortedSequence.add(j, array[i]);
            }
```

```
49          }
50
51          return sortedSequence;
52      }
53
54      public static Map<String, Integer> wordCountMap(Sequence<String> words) {
55          Map<String, Integer> wordMap = new Map2<String, Integer>();
56
57          for (int i = 0; i < words.length(); i++) {
58              String word = words.entry(i);
59              int occurs = 1;
60              if (i + 1 < words.length()) {
61                  while (word.equals(words.entry(i + 1))) {
62                      i++;
63                      occurs++;
64                  }
65              }
66
67              wordMap.add(word, occurs);
68
69          }
70
71          return wordMap;
72      }
73
74      public static void html(String fileName, Sequence<String> wordSeq,
75              Map<String, Integer> wordMap) {
76          SimpleWriter webOut = new SimpleWriter1L(
77                  "projects/01-word-counter/gettysburg.html");
78
79          webOut.println("<html> <head> <title> Words Counted in " + fileName
80                  + " </title> </head> <body> <h1> Words Counted in " + fileName
81                  + " </h1> <hr /> <main> <table border=\"1\"> <tr> <th> Words </th> <th> Counts
</th> </tr>");
82
83          int i = 0;
84          while (i < wordSeq.length()) {
85              int value = wordMap.value(wordSeq.entry(i));
86
87              webOut.println("<tr> <td>" + wordSeq.entry(i) + "</td> <td>" + value
88                      + "</td> </tr> ");
89
90              i += value;
91          }
92
93          webOut.println("</table> </main> </body> </html>");
94
95          webOut.close();
96      }
97
```

```
98      public static void main(String[] args) {
99          SimpleReader in = new SimpleReader1L();
100         SimpleWriter out = new SimpleWriter1L();
101
102         out.print("file name: ");
103         String file = in.nextLine();
104
105         out.close();
106         in.close();
107
108         String[] words = separateWords(file);
109
110         Sequence<String> sortedSeq = arrayToSortedSeq(words);
111
112         Map<String, Integer> wordMap = wordCountMap(sortedSeq);
113
114         html(file, sortedSeq, wordMap);
115
116     }
117 }
118
```