

```

1  /*
2  CSE 3342
3  Seung Ki Lee
4  HW03
5  */
6
7  package hw03;
8  import java.net.*;
9  import java.io.*;
10 import java.util.*;
11
12 public class HW03J {
13
14     static List<String> strToList = new ArrayList<String>();
15
16     //return the whole string
17     public static String f1URLReader(String inputURL) throws Exception
18     {
19         URL webpage = new URL(inputURL); // Set the URL to my index.html
20         BufferedReader in = new BufferedReader(new InputStreamReader(webpage.openStream())); // Dynamically allocate the BufferedReader
21         String inputLine = null; // Create a String to hold the readLine address
22         String holdLine = null;
23         while((holdLine = in.readLine()) != null) // While the read in is not empty, print the line to check
24         {
25             //add to list for processing -> call function on strToList
26             inputLine += holdLine;
27         }
28         //close the file
29         in.close();
30
31         return inputLine;
32     }
33 }

```

```

34     public static List<String> f2CleanString(String inputString)
35     {
36         List<String> strinList = new ArrayList<String>();
37         String[] cleanedString = null;
38         String[] splittedString = inputString.replaceAll("[a-zA-Z]", " ").toLowerCase().split("\\W+");
39
40         //for debugging
41         for(int i=0; i<splittedString.length; i++) {
42             //set rid of known html header
43             if(!"null".equals(splittedString[i]) && !"div".equals(splittedString[i]) && !"h".equals(splittedString[i]) && !"charset".equals(splittedString[i]) && !"utf".equals(splittedString[i]) && !"meta".equals(splittedString[i])
44             {
45                 strinList.add(splittedString[i]);
46             }
47         }
48         //return the list
49         return strinList;
50     }
51
52     public static Map<String, Integer> f3ListToMap(List<String> inputList) throws Exception {
53
54         //Read in the list of ignorewords from the URL
55         List<String> ignoreList = new ArrayList<String>();
56
57         URL ignoreWordsTxt = new URL("http://vls.snu.edu/~cavin/3342/hw3/ignoreWords.txt"); // Set the URL to my index.html
58         BufferedReader in = new BufferedReader(new InputStreamReader(ignoreWordsTxt.openStream())); // Dynamically allocate the BufferedReader
59         String ignoreWords = null; // Create a String to hold the readLine address
60
61         //Store the ignorewords into list for comparison
62         while((ignoreWords = in.readLine()) != null) {
63             //add to list for processing
64             ignoreList.add(ignoreWords);
65         }
66         //close the file
67         in.close();
68
69         //Compare inputList vs ignoreList and erase if inputList has anything with ignoreList
70         //Have the words in Either List by putting the inputList and adding ignoreLists
71         List<String> inEitherList = new ArrayList<String>(inputList);
72         inEitherList.addAll(ignoreList);
73
74         //Have the words in Both Lists by putting the inputList, retaining everything in ignoreList, and remove everything that's in both lists from eitherlist
75         List<String> inBothList = new ArrayList<String>(inputList);
76         inBothList.retainAll(ignoreList);
77         inEitherList.removeAll(inBothList);
78
79         //build map from the checked and trimmed list
80         Map<String, Integer> pairedMap = new HashMap<String, Integer>();
81
82         //if the function does not hold a key, add it. If it does, add the count
83         for(int i=0; i<inEitherList.size(); i++) {
84             if(!pairedMap.containsKey(inEitherList.get(i)))
85                 pairedMap.put(inEitherList.get(i), 1);
86             else {
87                 Integer eachWordCount = pairedMap.get(inEitherList.get(i));
88                 pairedMap.put(inEitherList.get(i), eachWordCount+1);
89             }
90         }
91         return pairedMap;
92     }
93
94     public static String f4JsonOutput(Map<String, Integer> inputMap, Integer nOfWords) throws Exception {
95
96         //initialize the string for usage
97         String jsonString = "";
98         Map<String, Integer> newMap = new HashMap<String, Integer>();

```

```

99
100         Map<String, Integer> newMap = new HashMap<String, Integer>();
101
102         for(int i=0; i<nOfWords; i++) {
103             int max = 0;
104             String word = null;
105
106             Iterator iterat = inputMap.keySet().iterator();
107
108             while(iterat.hasNext()) {
109                 Object key = iterat.next();
110                 Object value = inputMap.get(key);
111
112                 //pop off the biggest key value pair
113                 if((int)value > max) {
114                     max = (int)value;
115                     word = (String) key;
116                 }
117                 newMap.put(word, max);
118                 iterat.remove();
119             }
120
121             //Get the top N words in order
122
123             for(int i=0; i<i; i++) {
124                 int max = 0;
125                 String word = "";
126
127                 Iterator iterat = newMap.keySet().iterator();
128
129                 while(iterat.hasNext()) {
130                     Object key = iterat.next();
131                     Object value = newMap.get(key);

```

```

132
133             Map<String, Integer> newMap = new HashMap<String, Integer>();
134
135             for(int i=0; i<nOfWords; i++) {
136                 int max = 0;
137                 String word = null;
138
139                 Iterator iterat = inputMap.keySet().iterator();
140
141                 while(iterat.hasNext()) {
142                     Object key = iterat.next();
143                     Object value = inputMap.get(key);
144
145                     //pop off the biggest key value pair
146                     if((int)value > max) {
147                         max = (int)value;
148                         word = (String) key;
149                     }
150                     newMap.put(word, max);
151                     iterat.remove();
152                 }
153
154             //Get the top N words in order
155
156             for(int i=0; i<i; i++) {
157                 int max = 0;
158                 String word = "";
159
160                 Iterator iterat = newMap.keySet().iterator();
161
162                 while(iterat.hasNext()) {
163                     Object key = iterat.next();
164                     Object value = newMap.get(key);

```

```

128         while(iterat.hasNext()) {
129             Object key = iterat.next();
130             Object value = newMap.get(key);
131
132             // pop off the biggest key value pair
133             if((int)value > max) {
134                 max = (int)value;
135                 word = (String) key;
136             }
137             jsonString += "{W}" + word + "W:" + max + ";";
138             iterat.remove();
139         }
140     }
141
142     System.out.println(jsonString);
143
144     return jsonString;
145 }
146
147 /**
148  * @param args the command line arguments
149  */
150 public static void main(String[] args) throws Exception {
151
152     String strinedURL = f1URLReader("https://yle.snu.edu/~seungh/3342/index.html");
153     List<String> listOfStrings = f2CleanString(strinedURL);
154     Map<String, Integer> filteredMap = f3ListToMap(listOfStrings);
155     f4JsonOutput(filteredMap,3);
156 }
157
158 }
159

```

Output - HW03J (run) X

```

run:
{"mostly":21{"inter":7}{"interpreter":3}
BUILD SUCCESSFUL (total time: 0 seconds)

```