

```

1  /**
2   * EECS233 WrittenHW3
3   * Tung Ho Lin
4   * Spelling Checker: SpellChecker
5   */
6  import java.io.File;
7  import java.io.FileNotFoundException;
8  import java.util.Scanner;
9  import java.util.ArrayList;
10
11 //D:\School Documents\EECS233\HW5\input2.txt
12 public class SpellChecker {
13
14     private String inputfile;
15
16     private Dictionary dict;
17
18     public SpellChecker(String main, String personal, String input) {
19         this.inputfile = input;
20         dict = new Dictionary();
21         dict.build(main);
22         dict.build(personal);
23     }
24
25     //main method
26     public static void main(String[] args) {
27         Scanner sc = new Scanner(System.in);
28         System.out.println("Please enter the path of the main dictionary file.");
29         String main = sc.nextLine();
30         System.out.println("Please enter the path of the personal dictionary file.");
31         String personal = sc.nextLine();
32         System.out.println("Please enter the path of the text file to be spell
checked.");
33         String input = sc.nextLine();
34         SpellChecker checker = new SpellChecker(main, personal, input);
35         checker.spellCheck();
36     }
37
38     //method to spell check and print out the errors
39     public void spellCheck() {
40         try {
41             Scanner linesc = new Scanner(new File(inputfile));
42             int linenumber = 0;
43             String line;
44             String word;
45             while(linesc.hasNextLine()) {
46                 line = linesc.nextLine(); //scan a whole line of text
47                 linenumber++; //increment line number
48                 Scanner wordsc = new Scanner(line);
49                 while(wordsc.hasNext()) {
50                     word = wordsc.next().toLowerCase(); //scan a word from the line of text
51                     word = word.replaceAll("\\s*\\p{Punct}+\\s*$", "");
52                     if(dict.findWord(word) == false) { //if the word is not found in the built
dictionary
53                         System.out.println(word + " : in Line " + linenumber + " is not spelled
correctly.");
54                         System.out.println(appendSuggestions(word) + "\n");
55                     }
56                 }
57                 wordsc.close();
58             }
59             linesc.close();
60             System.out.println("Spellcheck completed!");
61         }
62         catch (FileNotFoundException e) {
63             System.err.println("File not found!");
64         }
65     }
66
67     //collect all the suggestions from the 3 methods and delete identical suggestions

```

```

68     public String appendSuggestions(String input) {
69         ArrayList<String> add = addChar(input);
70         ArrayList<String> remove = removeChar(input);
71         ArrayList<String> swap = swapChar(input);
72         ArrayList<String> suggestions = new ArrayList<String>();
73         String output = "Suggestions: ";
74         for(int i=0; i<add.size(); i++) //add all suggestions from first method
75             suggestions.add(add.get(i));
76         for(int i=0; i<remove.size(); i++) { //add non-recurrent suggestions
77             if(suggestions.contains(remove.get(i)) == false)
78                 suggestions.add(remove.get(i));
79         }
80         for(int i=0; i<swap.size(); i++) { //add non-recurrent suggestions
81             if(suggestions.contains(swap.get(i)) == false)
82                 suggestions.add(swap.get(i));
83         }
84         for(int i=0; i<suggestions.size()-1; i++) { //append a String of suggestions
85             output += suggestions.get(i) + ", "; //do not print out ", " on the last
word
86         }
87         try{
88             output += suggestions.get(suggestions.size()-1); //deal with the last word in
the list
89         }
90         catch(ArrayIndexOutOfBoundsException e){
91             System.err.println("Suggestions cannot be generated by the built-in methods,
Sorry!"); //if the misspelled word cannot be fixed by the 3 methods
92             output += "N/A";
93         }
94         return output;
95     }
96
97     //create suggestions by adding a character to anywhere in the word each time
98     public ArrayList<String> addChar(String input) {
99         ArrayList<String> suggestions = new ArrayList<String>();
100         char[] alphabets = new char[26];
101         for(int i=0; i<alphabets.length; i++) //create an array of all 26 alphabets
102             alphabets[i] = (char)('a'+ i);
103         for(char c : alphabets) { //for each alphabet to be inserted in the word
104             for(int i=0; i<=input.length(); i++) { //for each space in the word to be
inserted
105                 String suggest = input.substring(0, i) + c + input.substring(i, input.
length()); //insert the character between each adjacent characters
106                 if(dict.findWord(suggest)) //check if it is a correct word
107                     suggestions.add(suggest);
108             }
109         }
110         return suggestions;
111     }
112
113     //create suggestions by removing a character from anywhere in the word each time
114     public ArrayList<String> removeChar(String input) {
115         ArrayList<String> suggestions = new ArrayList<String>();
116         for(int i=0; i<input.length(); i++) {
117             StringBuilder builder = new StringBuilder(input);
118             builder.deleteCharAt(i); //delete one character each time
119             String suggest = builder.toString();
120             if(dict.findWord(suggest)) //compare to the dictionary if it is a correct
word
121                 suggestions.add(suggest);
122         }
123         return suggestions;
124     }
125
126     //create suggestions by swapping 2 characters in a word
127     public ArrayList<String> swapChar(String input) {
128         ArrayList<String> suggestions = new ArrayList<String>();
129         for(int i=0; i<input.length()-1; i++) {
130             char[] decon = input.toCharArray();
131             char swap = decon[i];

```

```
132         decon[i] = decon[i+1]; //swapping the characters
133         decon[i+1] = swap;
134         String suggest = new String(decon); //back to String
135         if(dict.findWord(suggest)) //check if it is a correct word
136             suggestions.add(suggest);
137     }
138     return suggestions;
139 }
140 }
141
142
143
144
145
```