

```
1 import java.io.File;
10
11
12 public class Frequencies {
13
14     /**
15      * This method organizes the frequencies based on the current keyboard layout and
16      * then returns a static
17      * array with the organized digraphs. These frequencies are in percentage form, not
18      * decimal form.
19      * @return the frequency array, organized to match the keyboard layout.
20      * @throws IOException if file cannot be found.
21      */
22     public static double[][] getFrequencies() throws IOException {
23         double[][] frequencyArray = new double[26][26];
24
25         FileInputStream file = new FileInputStream(
26             new File("C:\\Users\\apple\\eclipse-workspace\\Keyboards\\
27             \\digraphs.xlsx"));
28         Workbook workbook = new XSSFWorkbook(file);
29         Sheet sheet = workbook.getSheetAt(1);
30         workbook.close();
31
32         char cellLetter = 'a';
33         char rowLetter = 'a';
34         for (int i = 0; i < 26; i++) {
35             Row row = sheet.getRow(i);
36             for (int j = 0; j < 26; j++) {
37                 Cell cell = row.getCell(j);
38                 frequencyArray[search(rowLetter)][search(cellLetter)] = (double)
39                 cell.getNumericCellValue();
40                 cellLetter++;
41             }
42             rowLetter = 'a';
43             rowLetter++;
44         }
45
46         return frequencyArray;
47     }
48
49     /**
50      * This method searches the keyboard layout from the keyboard arranger to determine
51      * at what index the desired letter is located. It then returns the index so the
52      * frequency can be
53      * placed correspondingly in the array.
54      * @param letter is the desired letter being searched for.
55      * @return the index at which the desired letter is located in the keyboard array.
56      */
57     private static int search(char letter) {
58         for (int i = 0; i < 26; i++) {
59             if (KeyboardArranger.referenceKeyboard[i] == letter) {
60                 return i;
61             }
62         }
63         return -1;
64     }
65 }
```