Calling method

Testing method letterFrequencies

pass pass pass

	Arguments	Actual	Expected
pass	Hello	[0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0
pass	ABCDEZabcdez	[2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2]	[2, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
pass	"Hello, World!"	[0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 3, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0]	[0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 3, 0, 0, 2, 0, 0, 1, 0

Student files

Letters.java:

```
1   public class Letters
2   {
3      /**
4         Counts the frequencies of letters A-Za-z in a string
5         @param str a string
6         @return an array of 26 counts. The i-th count is the number of occurrences
7         of 'A' + i or 'a' + i.
8      */
9      public int[] letterFrequencies(String str)
10     {
11  int[] counts = new int[26];
12        for (int i = 0; i < str.length (); i++)
13        {
14            char ch = str.charAt(i);
15            if (ch == 'a' || ch == 'A')
16            {
17               counts[0]++;
18            }
19            if (ch == 'b' || ch == 'C')
20            {
21               counts[1]++;
22            }
23            if (ch == 'c' || ch == 'D')
24            {
25               counts[2]++;
26            }
27            if (ch == 'd' || ch == 'D')
28            {
29               counts[3]++;
30            }
31            if (ch == 'e' || ch == 'E')
32            {
33               counts[4]++;
34            }
35            if (ch == 'f' || ch == 'F')
36            {
37               counts[5]++;
38            }
39            if (ch == 'g' || ch == 'G')
40            {
41               counts[6]++;
42            }
43            if (ch == 'h' || ch == 'H')
44            {
45               counts[7]++;
46            }
47            if (ch == 'i' || ch == 'I')
48            {
49               counts[8]++;
50            }
51            if (ch == 'j' || ch == 'J')
52            {
53               counts[9]++;
54            }
55            if (ch == 'k' || ch == 'K')
56            {
57               counts[10]++;
58            }
59            if (ch == 'l' || ch == 'L')
60            {
61               counts[11]++;
62            }
63            if (ch == 'm' || ch == 'M')
64            {
65               counts[12]++;
66            }
67            if (ch == 'n' || ch == 'N')
68            {
69               counts[13]++;
70            }
71            if (ch == 'o' || ch == 'O')
72            {
73               counts[14]++;
74            }
75            if (ch == 'p' || ch == 'P')
76            {
77               counts[15]++;
```

```
78              }
79              if (ch == 'q' || ch == 'Q')
80              {
81                 counts[16]++;
82              }
83              if (ch == 'r' || ch == 'R')
84              {
85                 counts[17]++;
86              }
87              if (ch == 's' || ch == 'S')
88              {
89                 counts[18]++;
90              }
91              if (ch == 't' || ch == 'T')
92              {
93                 counts[19]++;
94              }
95              if (ch == 'u' || ch == 'U')
96              {
97                 counts[20]++;
98              }
99              if (ch == 'v' || ch == 'V')
100             {
101                counts[21]++;
102             }
103             if (ch == 'w' || ch == 'W')
104             {
105                counts[22]++;
106             }
107             if (ch == 'x' || ch == 'X')
108             {
109                counts[23]++;
110             }
111             if (ch == 'y' || ch == 'Y')
112             {
113                counts[24]++;
114             }
115             if (ch == 'z' || ch == 'Z')
116             {
117                counts[25]++;
118             }
119         }
120
121         return counts;
122
123     }
124 }
```

Score

3/3

Download

2017-07-12T02:57:13Z