ImplementTrie.java

```
1   package com.example.trie;
2
3   class Node {
4       Node links[] = new Node[26];
5       int cntEndWith = 0;
6       int cntPrefix = 0;
7
8       public Node() {
9       }
10
11      boolean containsKey(char ch) {
12  3       return (links[ch - 'a'] != null);
13      }
14
15      Node get(char ch) {
16  2       return links[ch - 'a'];
17      }
18
19      void put(char ch, Node node) {
20  1       links[ch - 'a'] = node;
21
22      }
23
24      void increaseEnd() {
25  1       cntEndWith++;
26      }
27
28      void increasePrefix() {
29  1       cntPrefix++;
30      }
31
32      void deleteEnd() {
33  1       cntEndWith--;
34      }
35
36      void reducePrefix() {
37  1       cntPrefix--;
38      }
39
40      int getEnd() {
41  1       return cntEndWith;
42      }
43
44      int getPrefix() {
45  1       return cntPrefix;
46      }
47  };
48
49  public class ImplementTrie {
50
```

```
51      private Node root;
52
53      // Initialize your data structure here
54
55      ImplementTrie() {
56          root = new Node();
57      }
58
59      // Inserts a word into the trie
60
61      public void insert(String word) {
62          Node node = root;
63  2       for (int i = 0; i < word.length(); i++) {
64  1           if (!node.containsKey(word.charAt(i))) {
65  1               node.put(word.charAt(i), new Node());
66              }
67              node = node.get(word.charAt(i));
68  1           node.increasePrefix();
69          }
70  1       node.increaseEnd();
71      }
72
73      public int countWordsEqualTo(String word) {
74          Node node = root;
75  2       for (int i = 0; i < word.length(); i++) {
76  1           if (node.containsKey(word.charAt(i))) {
77                  node = node.get(word.charAt(i));
78              } else {
79                  return 0;
80              }
81          }
82  1       return node.getEnd();
83      }
84
85      public int countWordsStartingWith(String word) {
86          Node node = root;
87  2       for (int i = 0; i < word.length(); i++) {
88  1           if (node.containsKey(word.charAt(i))) {
89                  node = node.get(word.charAt(i));
90              } else {
91                  return 0;
92              }
93          }
94  1       return node.getPrefix();
95      }
96
97      public void erase(String word) {
98          Node node = root;
99  2       for (int i = 0; i < word.length(); i++) {
100 1           if (node.containsKey(word.charAt(i))) {
101                 node = node.get(word.charAt(i));
102 1               node.reducePrefix();
103             } else {
104                 return;
```

```
105                 }
106             }
107 1           node.deleteEnd();
108         }
109 }
```

Mutations

12	1. negated conditional → KILLED 2. replaced boolean return with true for com/example/trie/Node::containsKey → KILLED 3. Replaced integer subtraction with addition → KILLED
16	1. Replaced integer subtraction with addition → KILLED 2. replaced return value with null for com/example/trie/Node::get → KILLED
20	1. Replaced integer subtraction with addition → KILLED
25	1. Replaced integer addition with subtraction → KILLED
29	1. Replaced integer addition with subtraction → KILLED
33	1. Replaced integer subtraction with addition → KILLED
37	1. Replaced integer subtraction with addition → SURVIVED
41	1. replaced int return with 0 for com/example/trie/Node::getEnd → KILLED
45	1. replaced int return with 0 for com/example/trie/Node::getPrefix → KILLED
63	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
64	1. negated conditional → KILLED
65	1. removed call to com/example/trie/Node::put → KILLED
68	1. removed call to com/example/trie/Node::increasePrefix → KILLED
70	1. removed call to com/example/trie/Node::increaseEnd → KILLED
75	1. negated conditional → KILLED 2. changed conditional boundary → KILLED
76	1. negated conditional → KILLED
82	1. replaced int return with 0 for com/example/trie/ImplementTrie::countWordsEqualTo → KILLED
87	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
88	1. negated conditional → KILLED
94	1. replaced int return with 0 for com/example/trie/ImplementTrie::countWordsStartingWith → KILLED
99	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
100	1. negated conditional → KILLED
102	1. removed call to com/example/trie/Node::reducePrefix → SURVIVED
107	1. removed call to com/example/trie/Node::deleteEnd → KILLED

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS

- VOID_METHOD_CALLS

Tests examined

- com.example.trie.NumberofDistinctSubstringTest.main(com.example.trie.NumberofDistinctSubstringTest) (0 ms)
- com.example.trie.ImplementTrieTest.implementTrieTest(com.example.trie.ImplementTrieTest) (0 ms)

Report generated by PIT 1.15.0