ImplementTrie.java

```
   package com.example;

   class Node {
       Node links[] = new Node[26];
       int cntEndWith = 0;
       int cntPrefix = 0;

       public Node() {
       }

       boolean containsKey(char ch) {
3          return (links[ch - 'a'] != null);
       }
       Node get(char ch) {
2          return links[ch-'a'];
       }
        void put(char ch, Node node) {
1          links[ch-'a'] = node;

       }
        void increaseEnd() {
1          cntEndWith++;
       }
        void increasePrefix() {
1          cntPrefix++;
       }
        void deleteEnd() {
1          cntEndWith--;
       }
        void reducePrefix() {
1          cntPrefix--;
       }
        int getEnd() {
1          return cntEndWith;
       }
        int getPrefix() {
1          return cntPrefix;
       }
   };
   public class ImplementTrie {



                 private Node root;

                 //Initialize your data structure here

                 ImplementTrie() {
                     root = new Node();
                 }

```

```
      //Inserts a word into the trie

      public void insert(String word) {
          Node node = root;
2         for(int i = 0;i<word.length();i++) {
1             if(!node.containsKey(word.charAt(i))) {
1                 node.put(word.charAt(i), new Node());
              }
              node = node.get(word.charAt(i));
1             node.increasePrefix();
          }
1         node.increaseEnd();
      }


      public int countWordsEqualTo(String word) {
          Node node = root;
2         for(int i = 0;i<word.length();i++) {
1             if(node.containsKey(word.charAt(i))) {
                  node = node.get(word.charAt(i));
              }
              else {
                  return 0;
              }
          }
1         return node.getEnd();
      }

      public int countWordsStartingWith(String word) {
          Node node = root;
2         for(int i = 0;i<word.length();i++) {
1             if(node.containsKey(word.charAt(i))) {
                  node = node.get(word.charAt(i));
              }
              else {
                  return 0;
              }
          }
1         return node.getPrefix();
      }

      public void erase(String word) {
          Node node = root;
2         for(int i = 0;i<word.length();i++) {
1             if(node.containsKey(word.charAt(i))) {
                  node = node.get(word.charAt(i));
1                 node.reducePrefix();
              }
              else {
                  return;
              }
          }
1         node.deleteEnd();
      }
```

```
107 }
```

Mutations

12	1. replaced boolean return with true for com/example/Node::containsKey → KILLED 2. Replaced integer subtraction with addition → KILLED 3. negated conditional → KILLED
15	1. Replaced integer subtraction with addition → KILLED 2. replaced return value with null for com/example/Node::get → KILLED
18	1. Replaced integer subtraction with addition → KILLED
22	1. Replaced integer addition with subtraction → KILLED
25	1. Replaced integer addition with subtraction → KILLED
28	1. Replaced integer subtraction with addition → KILLED
31	1. Replaced integer subtraction with addition → SURVIVED
34	1. replaced int return with 0 for com/example/Node::getEnd → KILLED
37	1. replaced int return with 0 for com/example/Node::getPrefix → KILLED
57	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
58	1. negated conditional → KILLED
59	1. removed call to com/example/Node::put → KILLED
62	1. removed call to com/example/Node::increasePrefix → KILLED
64	1. removed call to com/example/Node::increaseEnd → KILLED
70	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
71	1. negated conditional → KILLED
78	1. replaced int return with 0 for com/example/ImplementTrie::countWordsEqualTo → KILLED
83	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
84	1. negated conditional → KILLED
91	1. replaced int return with 0 for com/example/ImplementTrie::countWordsStartingWith → KILLED
96	1. changed conditional boundary → KILLED 2. negated conditional → KILLED
97	1. negated conditional → KILLED
99	1. removed call to com/example/Node::reducePrefix → SURVIVED
105	1. removed call to com/example/Node::deleteEnd → KILLED

Active mutators

- BOOLEAN_FALSE_RETURN
- BOOLEAN_TRUE_RETURN
- CONDITIONALS_BOUNDARY_MUTATOR
- EMPTY_RETURN_VALUES
- INCREMENTS_MUTATOR
- INVERT_NEGS_MUTATOR
- MATH_MUTATOR
- NEGATE_CONDITIONALS_MUTATOR
- NULL_RETURN_VALUES
- PRIMITIVE_RETURN_VALS_MUTATOR
- VOID_METHOD_CALL_MUTATOR

Tests examined

- com.example.ImplementTrieTest.implementTrieTest(com.example.ImplementTrieTest) (0 ms)