Michelle Chung
Writing Exercise #3
Summer 2018 Tech Intern Writing Exercise

Audience

This document serves programmers familiar with Java and object-oriented programming concepts.

Overview

This document describes the findNeedles() method and its parameters.

What does it do

The findNeedles() method counts and prints the number of occurrences of words (needles) in a given sentence (haystack).

Parameters

`String haystack`

Accepts a string argument.

When the string is split into its individual words, the following regex delimiters are supported:

`\"`	Double quotes
`\'`	Single quotes
`\t`	Tabs
`\n`	Newlines
`\b`	Backspaces
`\f`	Form feeds
`\r`	Carriage returns

`String[] needles`

Accepts up to and including five string arguments.

Please note that there is no error catching for invalid arguments; use this method with discretion.

Example input/output

```
String haystack = new String(“foo foo apple cherry bar”)
String[] needles = new String{“foo”, “bar”}

foo: 2
bar: 1
```

Java implementation

```java
public static void findNeedles(String haystack, String[] needles) {
      if(needles.length > 5) {
            System.err.println("Too many words!");
      }
      else {
            /* countArray is used to index the count of each word in the
            needles array. */
            int[] countArray = new int[needles.length];
            // For each word in the needles array:
            for(int i = 0; i < needles.length; i++){
                  /* The haystack string is split into its individual words and
                  stored into the words array. */
                  String[] words = haystack.split("[ \"\'\t\n\b\f\r]", 0);
                  // And for each word in the words array:
                  for(int j = 0; j < words.length; j++){
                        // If the word matches the word from the needles array,
                        if(words[j].compareTo(needles[i]) == 0) {
                              // Increment the corresponding index in countArray.
                              countArray[i]++;
                              }
                        }
            }
            // Finally, print the results.
            for (int j = 0; j < needles.length; j++) {
                  System.out.println(needles[j] + ": " + countArray[j]);
            }
      }
}
```

Suggested edits

1. Eliminate repeated work by moving

   ```
   String[] words = haystack.split("[ \"\'\t\n\b\f\r]", 0);
   ```

   outside of the first for loop.

2. Instead of using a nested for loop, consider creating a hashtable for the haystack words (keys) and their counts (values). Search the hashtable by key and return its count. This implementation will reduce search time from $O(N^2)$ to $O(1)$ and eliminate the need for a countArray.

3. Strongly consider adding exception catching for invalid arguments (null case, non string type, non-supported delimiter, etc.), as well as a more specific error statement (ex. “Needles only accepts up to and including five strings.”)