

```

public static void findNeedles(String haystack, String[] needles) {
    if (needles.length > 5) {
        System.err.println("Too many words!");
    } else {
        int[] countArray = new int[needles.length];
        for (int i = 0; i < needles.length; i++) {
            String[] words = haystack.split("[ \\t\\n\\b\\f\\r]", 0);
            for (int j = 0; j < words.length; j++) {
                if (words[j].compareTo(needles[i]) == 0) {
                    countArray[i]++;
                }
            }
        }
        for (int j = 0; j < needles.length; j++) {
            System.out.println(needles[j] + ": " + countArray[j]);
        }
    }
}

```

Assignment:

1. 1) Write documentation for above code (1 page)
2. 2) Write email to author of code.

Java findNeedles() Method Documentation

****Version****: 1.0

****Creation Date****: June 20, 2022

****Creator****: Jonathan Cohn

Description:

The `findNeedles()` method searches for up to five specific case-sensitive "needle" strings within a larger "haystack" string. The method identifies which needles are present in the haystack and counts the number of times each needle appears.

This function is useful for locating specific substrings within a larger string while ignoring certain delimiters such as punctuation or white spaces.

Syntax:

```
```java
```

```
findNeedles(String haystack, String[] needles)
```

```
```
```

Parameters:

| **Parameter** | **Description** |
|----------------------|------------------------|
|----------------------|------------------------|

| | |
|-------|-------|
| ----- | ----- |
|-------|-------|

| | |
|-----------------------|--|
| <code>haystack</code> | A string to search through (case-sensitive). |
|-----------------------|--|

| | |
|----------------------|--|
| <code>needles</code> | An array containing up to 5 strings to search for in <code>haystack</code> (case-sensitive). |
|----------------------|--|

Example:

```
```java
```

```
String[] needles = {"universal", "stretch", "embroidery", "sharps", "quilting"};
```

```
String haystack = "hay hay universal hay stretch hay embroidery stretchstretch hay hay stretch";
```

```
quiltinghay sharps.";
```

```
findNeedles(haystack, needles);
```

```
...
```

```
Output:
```

```
...
```

```
universal: 1
```

```
stretch: 2
```

```
embroidery: 1
```

```
...
```

## ## Technical Details:

- **\*\*Delimiters\*\***: The method splits the `haystack` string using the following delimiters: `[ '\t\b\f\r' ]`. Any needles containing these characters will not be recognized.

- The search is **\*\*case-sensitive\*\***. A needle will only be detected if its case exactly matches the corresponding text in the haystack.

- **\*\*Limitations\*\***:

- Needles must be discrete words and cannot be part of other words (e.g., "cat" will not be detected

in "catastrophe").

- The method prints the results and does not return them in an array or other format.

## ## Warnings:

1. **\*\*Delimiter Handling\*\***: Be cautious when searching for needles that contain any of the delimiters ( `[ '\t\b\f\r' ]` ), as they will not be counted.

2. **\*\*Case Sensitivity\*\***: The method does not perform a case-insensitive search, so ensure that needle strings match the case of their corresponding haystack values.

3. **\*\*Exact Matches Required\*\***: Needles that include punctuation (e.g., commas, periods) directly

after them in the haystack will not be detected.

---

# Email to the Author

---

**\*\*Subject\*\***: Questions About findNeedles() Documentation

Hi [Author's Name],

Thank you for sending over the `findNeedles()` method for documentation. I have started working on it and wanted to clarify a few points to make sure everything is explained clearly. I've listed my questions below, but I am also happy to meet and discuss them if that's easier.

1. Is there a reason for the five-needle limit? Should users be aware of this as a best practice or

constraint?

2. Why are these specific characters `["' \t\b\f\r]` chosen as delimiters? Should I warn users about including them in the needles?

3. The method doesn't detect needles that are part of other words or followed by punctuation. Is there a reason for this behavior?

4. Since `str.compareTo()` is case-sensitive, should we highlight this for users?

5. Can you explain why the method prints the results instead of returning them in an array or another format?

6. Lastly, could you clarify why `String[] words = haystack.split...` is placed in the outer loop?

Thanks again for your time and clarification.

Best,

Jonathan Cohn

Technical Writer

P.S. Adding a sixth needle in the next version might be worth considering 😊