

JAVA.UTIL.SCANNER.FINDWITHINHORIZON METHOD

http://www.tutorialspoint.com/java/util/scanner_findwithinhorizon_pattern.htm

Copyright © tutorialspoint.com

Description

The **java.util.Scanner.findWithinHorizon** method Attempts to find the next occurrence of the specified pattern. This method searches through the input up to the specified search horizon, ignoring delimiters. If the pattern is found the scanner advances past the input that matched and returns the string that matched the pattern. If no such pattern is detected then the null is returned and the scanner's position remains unchanged. This method may block waiting for input that matches the pattern. A scanner will never search more than horizon code points beyond its current position. Note that a match may be clipped by the horizon; that is, an arbitrary match result may have been different if the horizon had been larger.

Declaration

Following is the declaration for **java.util.Scanner.findWithinHorizon** method

```
public String findWithinHorizon(Pattern pattern, int horizon)
```

Parameters

- **pattern** -- a string specifying the pattern to search for

Return Value

This method returns the text that matched the specified pattern.

Exception

- **IllegalStateException** -- if this scanner is closed
- **IllegalArgumentException** -- if horizon is negative

Example

The following example shows the usage of java.util.Scanner.findWithinHorizon method.

```
package com.tutorialspoint;

import java.util.*;
import java.util.regex.Pattern;

public class ScannerDemo {

    public static void main(String[] args) {

        String s = "Hello World! 3+3.0=6";

        // create a new scanner with the specified String Object
        Scanner scanner = new Scanner(s);

        // find a pattern of 2 letters before rld, with horizon of 10
        System.out.println("
+ scanner.findWithinHorizon(Pattern.compile("..rld"), 10));

        // find a pattern of 2 letters before rld, with horizon of 20
        System.out.println("
+ scanner.findWithinHorizon(Pattern.compile("..rld"), 20));

        // print the rest of the string
        System.out.println("
+ scanner.nextLine());

        // close the scanner
```

```
scanner.close();  
}  
}
```

Let us compile and run the above program, this will produce the following result:

```
null  
World  
! 3+3.0=6
```

Loading [Mathjax]/jax/output/HTML-CSS/jax.js