

JAVA.UTIL.ARRAYS.BINARYSEARCH METHOD

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

Copyright © tutorialspoint.com

Description

The **java.util.Arrays.binarySearch**`char[]a, intfromIndex, inttoIndex, charkey` method searches a range of the specified array of chars for the specified value using the binary search algorithm. The range must be sorted before making this call. If it is not sorted, the results are undefined.

Declaration

Following is the declaration for **java.util.Arrays.binarySearch** method

```
public static int binarySearch(char[] a, int fromIndex, int toIndex, char key)
```

Parameters

- **a** -- This is the array to be searched.
- **fromIndex** -- This is the index of the first element *inclusive* to be searched.
- **toIndex** -- This is the index of the last element *exclusive* to be searched.
- **key** -- This is the value to be searched for.

Return Value

This method returns index of the search key, if it is contained in the array, else it returns `-(insertionpoint - 1)`. The insertion point is the point at which the key would be inserted into the array; the index of the first element in the range greater than the key, or `toIndex` if all elements in the range are less than the specified key.

Exception

- **IllegalArgumentException** -- if `fromIndex > toIndex`
- **ArrayIndexOutOfBoundsException** -- if `fromIndex < 0` or `toIndex > a.length`

Example

The following example shows the usage of `java.util.Arrays.binarySearch` method.

```
package com.tutorialspoint;

import java.util.Arrays;

public class ArrayDemo {

    public static void main(String[] args) {

        // initializing unsorted char array
        char charArr[] = {'a', 'c', 'b', 'e', 'd'};

        // sorting array
        Arrays.sort(charArr);

        // let us print all the elements available in list
        System.out.println("The sorted char array is:");
        for (char number : charArr) {
            System.out.println("Number = " + number);
        }

        // entering the value to be searched
        char searchVal = 'e';
```

```
// entering the range of index
int retVal = Arrays.binarySearch(charArr,1,5,searchVal);

System.out.println("The index of e is : " + retVal);
}
}
```

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

```
The sorted char array is:
Number = a
Number = b
Number = c
Number = d
Number = e
The index of e is : 4
```

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