JAVA.UTIL.ARRAYS.BINARYSEARCH METHOD

 Copyright © tutorialspoint.com

Description

The **java.util.Arrays.binarySearch***Object*[]*a, intfromIndex, inttoIndex, Objectkey* method searches a range of the specified array for the specified object using the binary search algorithm.The range must be sorted into ascending order according to the natural ordering of its elements 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(Object[] a, int fromIndex, int toIndex, Object 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

- **ClassCastException** -- if the search key is not comparable to the elements of the array within the specified range.
- **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 array
   Object arr[] = {10,2,22,69};

   // sorting array
   Arrays.sort(arr);

   // let us print all the elements available
   System.out.println("The sorted array is:");
   for (Object number : arr) {
   System.out.println("Number = " + number);
```

```
   }

   // entering the value to be searched
   int searchVal = 22;

   // entering range of index
   int retVal = Arrays.binarySearch(arr,2,4,searchVal);

   System.out.println("The index of element 22 is : " + retVal);
   }
}
```

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

```
The sorted array is:
Number = 2
Number = 10
Number = 22
Number = 69
The index of element 22 is : 2
```

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