

JAVA.UTIL.ARRAYS.COPYOFRANGE METHOD

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

Copyright © tutorialspoint.com

Description

The **java.util.Arrays.copyOfRange**(U[] original, int from, int to, Class < ? extends T[] > newType) method Copies the specified range of the specified array into a new array. The initial index of the range from must lie between zero and original.length, inclusive. The value at original[from] is placed into the initial element of the copy unless from == original.length or from == to.

Values from subsequent elements in the original array are placed into subsequent elements in the copy. The final index of the range to, which must be greater than or equal to from, may be greater than original.length, in which case null is placed in all elements of the copy whose index is greater than or equal to original.length - from. The length of the returned array will be to - from. The resulting array is of the class newType.

Declaration

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

```
public static <T,U> T[] copyOfRange(U[] original, int from, int to, Class<? extends T[]> newType)
```

Parameters

- **original** -- This is the array from which a range is to be copied.
- **from** -- This is the initial index of the range to be copied, inclusive.
- **to** -- This is the final index of the range to be copied, exclusive.
- **newType** -- The class of the copy to be returned

Return Value

This method returns a new array containing the specified range from the original array, truncated or padded with nulls to obtain the required length

Exception

- **ArrayIndexOutOfBoundsException** -- If from < 0 or from > original.length
- **IllegalArgumentException** -- If from > to.
- **NullPointerException** -- If original is null.
- **ArrayStoreException** -- If an element copied from original is not of a runtime type that can be stored in an array of class newType.

Example

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

```
package com.tutorialspoint;

import java.util.Arrays;

public class ArrayDemo {

    public static void main(String[] args) {

        // intializing an array arr1
        Short arr1[] = new Short[]{15, 10, 45};
```

```
// printing the array
System.out.println("Printing 1st array:");
for (int i = 0; i < arr1.length; i++) {
System.out.println(arr1[i]);
}

// copying array arr1 to arr2 as Number with range of index from 1 to 3
Number[] arr2 = Arrays.copyOfRange(arr1, 1, 3, Number[].class);

// printing the array arr2, which is a Number array
System.out.println("Printing new array:");
for (int i = 0; i < arr2.length; i++) {
System.out.println(arr2[i]);
}
}
```

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

```
Printing 1st array:
15
10
45
Printing new array:
10
45
```

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