JAVA.UTIL.TREEMAP.SUBMAP METHOD

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

Description

The **subMap***KfromKey, KtoKey* method is used to return a view of the portion of this map whose keys range from fromKey, inclusive, to toKey, exclusive. *IffromKeyandtoKeyareequal, thereturnedmapisempty*. The returned map is backed by this map, so changes in the returned map are reflected in this map, and vice-versa.

Declaration

Following is the declaration for **java.util.TreeMap.subMap** method.

```
public SortedMap<K,V> subMap(K fromKey,K toKey)
```

Parameters

- **fromKey** -- This is the low endpoint *inclusive* of the keys in the returned map.
- **toKey** -- This is the high endpoint *exclusive* of the keys in the returned map.

Return Value

The method call returns a view of the portion of this map whose keys range from fromKey, inclusive, to toKey, exclusive.

Exception

- **ClassCastException** -- This exception is thrown if fromKey and toKey cannot be compared to one another using this map's comparator.
- **NullPointerException** -- This exception is thrown if fromKey or toKey is null and this map uses natural ordering, or its comparator does not permit null keys.
- **IllegalArgumentException** -- This exception is thrown if fromKey is greater than toKey; or if this map itself has a restricted range, and fromKey or toKey lies outside the bounds of the range.

Example

The following example shows the usage of java.util.TreeMap.subMap

```
package com.tutorialspoint;

import java.util.*;

public class TreeMapDemo {
   public static void main(String[] args) {
   // creating maps
   TreeMap<Integer, String> treemap = new TreeMap<Integer, String>();
   SortedMap<Integer, String> treemapincl = new TreeMap<Integer, String>();

   // populating tree map
   treemap.put(2, "two");
   treemap.put(1, "one");
   treemap.put(3, "three");
   treemap.put(6, "six");
   treemap.put(5, "five");

   System.out.println("Getting a portion of the map");
   treemapincl=treemap.subMap(1,5);
   System.out.println("Sub map values: "+treemapincl);
```

```
    }
}
```

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

```
Getting a portion of the map
Sub map values: {1=one, 2=two, 3=three}
```