

What is NavigableMap in Java 6 - Creating subMap from Map with Example

NavigableMap in Java 6 is an extension of SortedMap like [TreeMap](#) which provides convenient navigation method like `lowerKey`, `floorKey`, `ceilingKey` and `higherKey`.

NavigableMap is added on Java 1.6 and along with these popular navigation method it also provide ways to *create a Sub Map from existing Map in Java* e.g. `headMap` whose keys are less than specified key, `tailMap` whose keys are greater than specified key and a `subMap` which is strictly contains keys which falls between `toKey` and `fromKey`. All of these methods also provides a boolean to include specified key or not. [TreeMap](#) and `ConcurrentSkipListMap` are two concrete implementation of NavigableMap in Java 1.6 API. Though NavigableMap is not as popular as [HashMap](#), [ConcurrentHashMap](#) or [Hashtable](#) but given that `TreeMap` implements NavigableMap you already get all good things in a well known Map implementation.



How to use NavigableMap in Java - Example



In this Java tutorial we will explore some API methods of NavigableMap to show its functionality. This Java program shows example of `lowerKey` which returns keys less than specified, `floorKey` returns key less than or equal to, `ceilingKey` return greater than or equal to and `higherKey` which returns keys which are greater than specified key.

This Java example also demonstrate use of `headMap()`, `tailMap()` and `subMap()` method which is used to *create Map from an existing Map in Java*. `headMap` returns a [Map](#) whose keys are lower than specified keys while `tailMap` returns Map which contains keys, those are higher than specified. Here is complete code example of *How to use NavigableMap in Java*.

```
import java.util.NavigableMap;
import java.util.TreeMap;

/**
 *
 * Java program to demonstrate What is NavigableMap in Java and How to use NavigableMap
 * in Java. NavigableMap provides two important features navigation methods
 * like lowerKey(), floorKey, ceilingKey() and higherKey().
 * There Entry counterpart and methods to create subMap e.g. headMap(), tailMap()
 * and subMap().
 *
 * @author Javin Paul
 */
```

Interview Questions

[core java interview question \(161\)](#)

[data structure and algorithm \(45\)](#)

[Coding Interview Question \(32\)](#)

[SQL Interview Questions \(24\)](#)

[thread interview questions \(20\)](#)

[database interview questions \(18\)](#)

[servlet interview questions \(17\)](#)

[collections interview questions \(15\)](#)

[spring interview questions \(9\)](#)

[Programming interview question \(4\)](#)

[hibernate interview questions \(4\)](#)

Translate this blog

Select Language ▼

Powered by [Google Translate](#)

```

public class NavigableMapExample {

    public static void main(String args[]) {

        //NavigableMap extends SortedMap to provide useful navigation methods
        NavigableMap<String, String> navigableMap = new TreeMap<String, String>();

        navigableMap.put("C++", "Good programming language");
        navigableMap.put("Java", "Another good programming language");
        navigableMap.put("Scala", "Another JVM language");
        navigableMap.put("Python", "Language which Google use");

        System.out.println("SorteMap : " + navigableMap);

        //lowerKey returns key which is less than specified key
        System.out.println("lowerKey from Java : " + navigableMap.lowerKey("Java"));

        //floorKey returns key which is less than or equal to specified key
        System.out.println("floorKey from Java: " + navigableMap.floorKey("Java"));

        //ceilingKey returns key which is greater than or equal to specified key
        System.out.println("ceilingKey from Java: " + navigableMap.ceilingKey("Java"));

        //higherKey returns key which is greater specified key
        System.out.println("higherKey from Java: " + navigableMap.higherKey("Java"));

        //Apart from navigagtion methodk, it also provides useful method
        //to create subMap from existing Map e.g. tailMap, headMap and subMap

        //an example of headMap - returns NavigableMap whose key is less than specified
        NavigableMap<String, String> headMap = navigableMap.headMap("Python", false);
        System.out.println("headMap created form navigableMap : " + headMap);

        //an example of tailMap - returns NavigableMap whose key is greater than specified
        NavigableMap<String, String> tailMap = navigableMap.tailMap("Scala", false);
        System.out.println("tailMap created form navigableMap : " + tailMap);

        //an example of subMap - return NavigableMap from toKey to fromKey
        NavigableMap<String, String> subMap = navigableMap.subMap("C++", false ,
                                                                "Python", false);
        System.out.println("subMap created form navigableMap : " + subMap);
    }
}

```

Output:

```

SorteMap : {C++=Good programming language, Java=Another good programming language,
Python=Language which Google use, Scala=Another JVM language}
lowerKey from Java : C++
floorKey from Java: Java
ceilingKey from Java: Java
higherKey from Java: Python
headMap created form navigableMap : {C++=Good programming language, Java=Another good
programming language}
tailMap created form navigableMap : {}
subMap created form navigableMap : {Java=Another good programming language}

```

That's all on **What is NavigableMap in Java** and How to use NavigableMap with example. We have seen examples of popular navigation method on TreeMap e.g. floorKey. You can also use similar method like lowerEntry, floorEntry,

Java Tutorials

[date and time tutorial \(18\)](#)[FIX protocol tutorial \(16\)](#)[java collection tutorial \(53\)](#)[java IO tutorial \(25\)](#)[Java JSON tutorial \(6\)](#)[Java multithreading Tutorials \(33\)](#)[Java Programming Tutorials \(27\)](#)[Java xml tutorial \(9\)](#)

**Download 2
free audiobooks**

[Start Here](#)

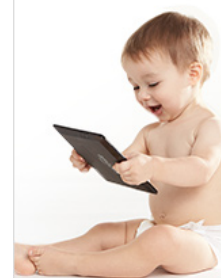


Search This Blog



The best
selection
and price
from any device

[Start your registry](#)



[Privacy](#)