NavigableMap Interface in Java with Example

NavigableMap is an extension of SortedMap which provides convenient navigation method like lowerKey, floorKey, ceilingKey and higherKey, 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.

An example class that implements NavigableMap is TreeMap.

Methods of NavigableMap:

1. **lowerKey(Object key)** : Returns the greatest key strictly less than the given key, or if there is no such key.
2. **floorKey(Object key)** : Returns the greatest key less than or equal to the given key, or if there is no such key.
3. **ceilingKey(Object key)** : Returns the least key greater than or equal to the given key, or if there is no such key.
4. **higherKey(Object key)** : Returns the least key strictly greater than the given key, or if there is no such key.
5. **descendingMap()** : Returns a reverse order view of the mappings contained in this map.
6. **headMap(object toKey, boolean inclusive)** : Returns a view of the portion of this map whose keys are less than (or equal to, if inclusive is true) toKey.
7. **subMap(object fromKey, boolean fromInclusive, object toKey, boolean toInclusive)** : Returns a view of the portion of this map whose keys range from fromKey to toKey.
8. **tailMap(object fromKey, boolean inclusive)** : Returns a view of the portion of this map whose keys are greater than (or equal to, if inclusive is true) fromKey.

```java
// Java program to demonstrate NavigableMap
import java.util.NavigableMap;
import java.util.TreeMap;
```

```java
public class Example
{
    public static void main(String[] args)
    {
        NavigableMap<String, Integer> nm =
                                  new TreeMap<String, Integer>();
        nm.put("C", 888);
        nm.put("Y", 999);
        nm.put("A", 444);
        nm.put("T", 555);
        nm.put("B", 666);
        nm.put("A", 555);

        System.out.printf("Descending Set  : %s%n",
                                nm.descendingKeySet());
        System.out.printf("Floor Entry  : %s%n",
                                nm.floorEntry("L"));
        System.out.printf("First Entry  : %s%n",
                                nm.firstEntry());
        System.out.printf("Last Key : %s%n",
                                nm.lastKey());
        System.out.printf("First Key : %s%n",
                                nm.firstKey());
        System.out.printf("Original Map : %s%n", nm);
        System.out.printf("Reverse Map : %s%n",
                                 nm.descendingMap());
    }
}
```

Run on IDE

Output:

```
Descending Set : [Y, T, C, B, A]
Floor Entry : C=888
First Entry : A=555
Last Key : Y
First Key : A
Original Map : {A=555, B=666, C=888, T=555, Y=999}
Reverse Map : {Y=999, T=555, C=888, B=666, A=555}
```

▲

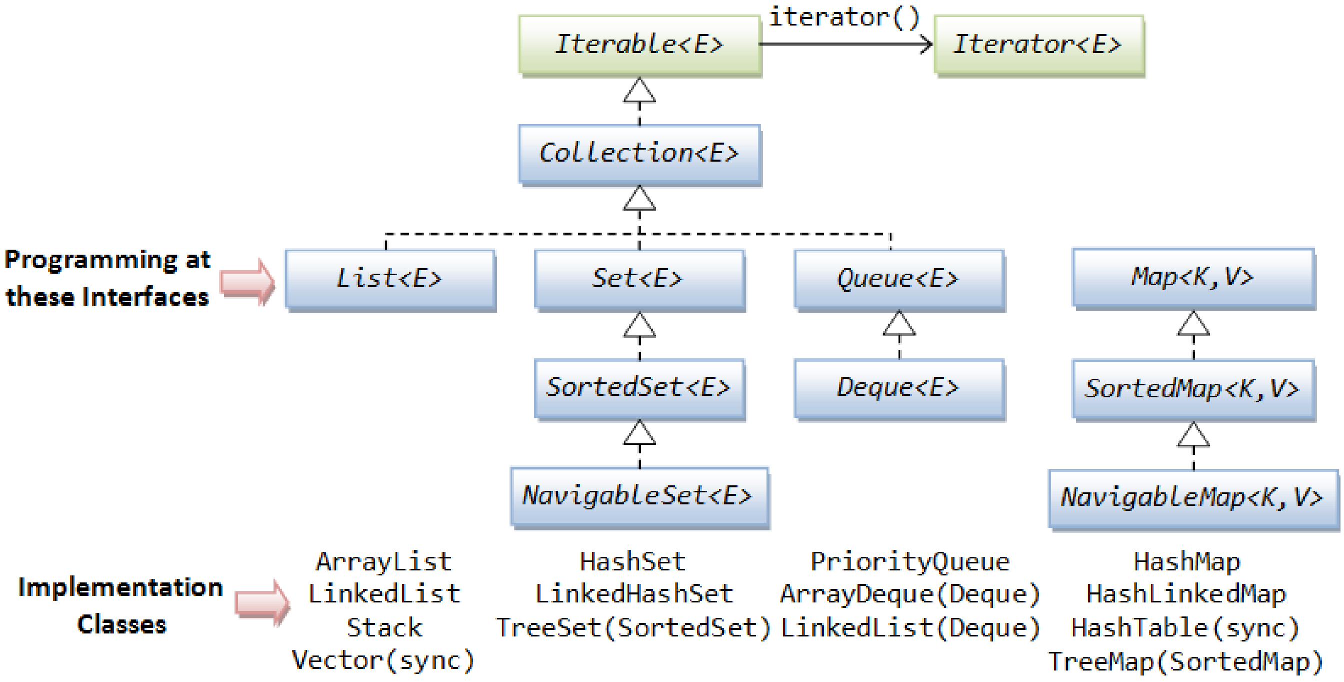

Image Source : https://www.ntu.edu.sg/home/ehchua/programming/java/J5c_Collection.html

This article is contributed by **Pratik Agarwal**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Java Java-Collections

Recommended Posts:

NavigableSet in Java with Examples
Java.util.Dictionary Class in Java
Map interface in Java with examples
SortedMap Interface in Java with Examples
Data types in Java
Java.lang.String.compareTo()
Templates in C++ vs Generics in Java
Converting Text to Speech in Java
Difference between x++ and x=x+1 in Java