TreeMap in Java with Example

BY CHAITANYA SINGH | FILED UNDER: JAVA COLLECTIONS

TreeMap is Red-Black tree based NavigableMap implementation. It is sorted according to the natural ordering of its keys.
TreeMap class implements Map interface similar to HashMap class. The main difference between them is that HashMap is an unordered collection while TreeMap is sorted in the ascending order of its keys. TreeMap is unsynchronized collection class which means it is not suitable for thread-safe operations until unless synchronized explicitly.

TreeMap Example

In this example we are storing the key and value mappings into the TreeMap and we are getting a sorted key-value mapping upon fetching the data from TreeMap.

```java
import java.util.TreeMap;
import java.util.Set;
import java.util.Iterator;
import java.util.Map;

public class Details {

   public static void main(String args[]) {

      /* This is how to declare TreeMap */
      TreeMap<Integer, String> tmap = 
             new TreeMap<Integer, String>();

      /*Adding elements to TreeMap*/
      tmap.put(1, "Data1");
      tmap.put(23, "Data2");
      tmap.put(70, "Data3");
      tmap.put(4, "Data4");
      tmap.put(2, "Data5");

      /* Display content using Iterator*/
      Set set = tmap.entrySet();
      Iterator iterator = set.iterator();
      while(iterator.hasNext()) {
         Map.Entry mentry = (Map.Entry)iterator.next();
         System.out.print("key is: "+ mentry.getKey() + " & Value is: ");
         System.out.println(mentry.getValue());
      }

   }
}
```

Output:

```
key is: 1 & Value is: Data1
key is: 2 & Value is: Data5
key is: 4 & Value is: Data4
key is: 23 & Value is: Data2
key is: 70 & Value is: Data3
```

As you can see that we have inserted the data in random order however when we displayed the TreeMap content we got the sorted result in the ascending order of keys.