

Fetching an element from a TreeMap#

The following methods can be used to fetch elements from a **TreeMap**.

Fetching the value for a particular key#

We can fetch the value for a particular key using the `get(Object o)` method. This method will return the value of the key if the key is present. If the key is not present, it will return null.

Fetching the smallest key#

As we are aware that a **TreeMap** stores elements in sorted order, we can fetch the smallest key using the `firstKey()` method. If the **TreeMap** is empty, then `NoSuchElementException` is thrown. If we want to get the smallest entry, then the `firstEntry()` method can be used.

Fetching the largest key#

We can fetch the largest key using the `lastKey()` method. If the **TreeMap** is empty, then `NoSuchElementException` is thrown. If we want to get the largest entry, then the `lastEntry()` method can be used.

```
1 import java.util.Map.Entry;
2 import java.util.TreeMap;
3
4 public class TreeMapDemo {
5
6     public static void main(String args[]) {
7
8         TreeMap<String, Integer> map = new TreeMap<>();
9         map.put("Oracle", 43);
10        map.put("Microsoft", 56);
11        map.put("Apple", 76);
12        map.put("Novartis", 87);
13
14        //Fetching the first entry in the Map.
15        Entry<String, Integer> firstEntry = map.firstEntry();
16
17
18        System.out.println("Smallest key: " + firstEntry.getKey() + ", Value: " + firstEntry.getValue());
19
20        //Fetching the last entry in the Map.
21        Entry<String, Integer> lastEntry = map.lastEntry();
22        System.out.println("Largest key: " + lastEntry.getKey() + ", Value: " + lastEntry.getValue());
23    }
24 }
25
```

Run Save Reset

Removing an element from a TreeMap#

To remove an element from **TreeMap**, the `remove(Object o)` method can be used. This method returns the previous value for this key if the key is present. If the key is not present, then `null` is returned.

```
1 import java.util.TreeMap;
2
3 public class TreeMapDemo {
4
5     public static void main(String args[]) {
6         TreeMap<String, Integer> map = new TreeMap<>();
7
8         map.put("Oracle", 43);
9         map.put("Microsoft", 56);
10        map.put("Apple", 43);
11        map.put("Novartis", 87);
12
13        System.out.println("Removing Oracle from Map. This will return the value corresponding to Oracle:");
14        System.out.println("Removing Google from Map. This will return null as Google is not present in the Map.");
15    }
16 }
17
```

Run Save Reset

Updating values in a TreeMap#

If we put a key-value pair in a **TreeMap**, and it is already present, then the value is updated. But what if we want to update the value of a key only if it is already present in the Map? In that case, we can use the `replace()` method provided in the **TreeMap**.

It has the following two variations:

Replacing without checking the previous value#

The `replace(Key k, value v)` method is used to replace the value of the given key. If the key is present then this method replaces the old value with the new value and returns the old value. If the key is not present then `null` is returned.

Replacing after checking the previous value#

The `replace(K key, V oldValue, V newValue)` method is used to replace the value of the given key if its current value is the same as the provided value. If the value is replaced, then `true` is returned. If not, then `false` is returned.

```
1 import java.util.TreeMap;
2
3 public class TreeMapDemo {
4
5     public static void main(String args[]) {
6
7         TreeMap<String, Integer> map = new TreeMap<>();
8         map.put("Oracle", 43);
9         map.put("Microsoft", 56);
10        map.put("Apple", 76);
11        map.put("Novartis", 87);
12
13        System.out.println("Replacing the value of Oracle : " + map.replace("Oracle", 67));
14        System.out.println("Latest value of Oracle : " + map.get("Oracle"));
15
16        System.out.println("Replacing the value of Apple only if current value is 50 : " + map.replace("Apple", 50));
17        System.out.println("Latest value of Oracle : " + map.get("Apple"));
18
19        System.out.println("Replacing the value of Apple only if current value is 76 : " + map.replace("Apple", 76));
20        System.out.println("Latest value of Oracle : " + map.get("Apple"));
21    }
22 }
23
```

Run Save Reset