

# Lớp TreeMap trong Java

Lớp TreeMap trong Java triển khai Map Interface bởi sử dụng một tree. Một TreeMap cung cấp các phương thức hiệu quả để lưu giữ các cặp key/value trong thứ tự được sắp xếp, và cho phép thu hồi nhanh chóng.

Bạn nên chú ý rằng, không giống một hash map, một tree map bảo đảm rằng các phần tử của nó sẽ được xếp thứ tự theo thứ tự key tăng dần.

Để hiểu sâu hơn các khái niệm được trình bày trong chương này, mời bạn tham khảo loạt bài: **Ví dụ về Collection trong Java**.

Lớp TreeMap trong Java hỗ trợ 4 constructor. Mẫu đầu tiên xây dựng một tree map trống mà sẽ được xếp thứ tự bởi sử dụng thứ tự tự nhiên cho các key.

```
TreeMap( )
```

Constructor thứ hai xây dựng một tree map trống mà sẽ được xếp thứ tự bởi sử dụng Comparator comp:

```
TreeMap(Comparator comp)
```

Constructor thứ ba khởi tạo một tree map với các entry từ m, mà sẽ được xếp thứ tự tự nhiên các key của nó.

```
TreeMap(Map m)
```

Constructor thứ 4 khởi tạo một tree map với các entry từ m, mà sẽ được xếp thứ tự giống như sm:

```
TreeMap(SortedMap sm)
```

Ngoài các phương thức được hỗ trợ bởi các lớp cha, lớp TreeMap cũng định nghĩa các phương thức sau:

STT	Phương thức và Miêu tả
1	<b>void clear()</b>  Gỡ bỏ tất cả ánh xạ (mapping) từ TreeMap này

2	<b>Object clone()</b>  Trả về một shallow copy của TreeMap này
3	<b>Comparator comparator()</b>  Trả về comparator được sử dụng để xếp thứ tự map này, hoặc trả về null nếu map này sử dụng cách sắp xếp theo thứ tự tự nhiên các key của nó
4	<b>boolean containsKey(Object key)</b>  Trả về true nếu map này chứa một ánh xạ cho key đã xác định
5	<b>boolean containsValue(Object value)</b>  Trả về true nếu map này so khớp một hoặc nhiều key tới value đã cho
6	<b>Set entrySet()</b>  Trả về một tập hợp các ánh xạ được chứa trong map này
7	<b>Object firstKey()</b>  Trả về key đầu tiên (thấp nhất) trong sorted map này
8	<b>Object get(Object key)</b>  Trả về value từ đó map này so khớp với key đã cho
9	<b>SortedMap headMap(Object toKey)</b>  Trả về một phần map này có các key nhỏ hơn toKey
10	<b>Set keySet()</b>  Trả về một Set các key được chứa trong map này
11	<b>Object lastKey()</b>

	Trả về key cuối cùng (cao nhất) trong sorted map này
12	<b>Object put(Object key, Object value)</b> Liên kết value đã cho với key đã xác định trong map này
13	<b>void putAll(Map map)</b> Sao chép tất cả ánh xạ từ map đã cho tới map này
14	<b>Object remove(Object key)</b> Gỡ bỏ ánh xạ cho key này từ TreeMap nếu có mặt
15	<b>int size()</b> Trả về số ánh xạ key-value trong map này
16	<b>SortedMap subMap(Object fromKey, Object toKey)</b> Trả về một phần map này mà có các key từ fromKey tới toKey
17	<b>SortedMap tailMap(Object fromKey)</b> Trả về một phần map này mà có key lớn hơn hoặc bằng fromKey
18	<b>Collection values()</b> Trả về một tập hợp các value được chứa trong map này

## Ví dụ

Chương trình sau minh họa các phương thức được hỗ trợ bởi lớp TreeMap trong Java:

```
import java.util.*;  
  
public class TreeMapDemo {
```

```
public static void main(String args[]) {  
    // Create a hash map  
    TreeMap tm = new TreeMap();  
  
    // Put elements to the map  
    tm.put("Zara", new Double(3434.34));  
    tm.put("Mahnaz", new Double(123.22));  
    tm.put("Ayan", new Double(1378.00));  
    tm.put("Daisy", new Double(99.22));  
    tm.put("Qadir", new Double(-19.08));  
  
    // Get a set of the entries  
    Set set = tm.entrySet();  
  
    // Get an iterator  
    Iterator i = set.iterator();  
  
    // Display elements  
    while(i.hasNext()) {  
        Map.Entry me = (Map.Entry)i.next();  
        System.out.print(me.getKey() + ": ");  
        System.out.println(me.getValue());  
    }  
    System.out.println();  
  
    // Deposit 1000 into Zara's account  
    double balance = ((Double)tm.get("Zara")).doubleValue();  
    tm.put("Zara", new Double(balance + 1000));  
    System.out.println("Zara's new balance: " +  
        tm.get("Zara"));  
}  
}
```

Nó sẽ cho kết quả sau:

```
Ayan: 1378.0  
Daisy 99.22  
Mahnaz: 123.22
```

Qadir: -19.08

Zara: 3434.34

Zara's current balance: 4434.34