

Java中Map根据键值(key)或者值(value)进行排序实现

我们都知道，java中的Map结构是key->value键值对存储的，而且根据Map的特性，同一个Map中不存在两个Key相同的元素，而value不存在这个限制。换句话说，在同一个Map中Key是唯一的，而value不唯一。Map是一个接口，我们不能直接声明一个Map类型的对象，在实际开发中，比较常用的Map性数据结构是HashMap和TreeMap,它们都是Map的直接子类。如果考虑到存取效率的话，建议使用HashMap数据结构，而如果需要考虑到Key的顺序，建议使用TreeMap，但是TreeMap在删除、添加过程中需要排序，性能比较差。

以Key进行排序

我们可以声明一个TreeMap对象

```
Map<Integer, Person> map = new TreeMap<Integer, Person>();
```

然后往map中添加元素，可以通过输出结果，可以发现map里面的元素都是排好序的

//遍历集合

```
for (Iterator<Integer> it = map.keySet().iterator(); it.hasNext();) {  
    Person person = map.get(it.next());  
    System.out.println(person.getId_card() + " " + person.getName());  
}
```

我们也可以声明一个HashMap对象，然后把HashMap对象赋值给TreeMap，如下：

```
Map<Integer, Person> map = new HashMap<Integer, Person>();  
TreeMap treemap = new TreeMap(map);
```

以Value进行排序

先声明一个HashMap对象：

```
Map<String, Integer> map = new HashMap<String, Integer>();
```

然后我们可以将Map集合转换成List集合中，而List使用ArrayList来实现如下：

```
List<Entry<String,Integer>> list =  
    new ArrayList<Entry<String,Integer>>(map.entrySet());
```

最后通过Collections.sort(List l, Comparator c)方法来进行排序，代码如下：

```
Collections.sort(list, new Comparator<Map.Entry<String, Integer>>() {  
    public int compare(Map.Entry<String, Integer> o1,  
        Map.Entry<String, Integer> o2) {  
        return (o2.getValue() - o1.getValue());  
    }  
});
```

上述代码是讲map中的value按照逆序排序，如果需要按照升序进行排序的话，只需要修改o2.getValue() - o1.getValue()为o1.getValue() - o2.getValue()即可

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（[过往记忆](#)）所有，未经许可不得转载。
本文链接: [【】](#)（）