Guava学习之TreeMultimap

TreeMultimap类是Multimap接口的实现子类，其中的key和value都是根据默认的自然排序或者用户指定的排序规则排好序的。在任何情况下，如果你想判断TreeMultimap中两个元素是否相等，都不要使用equals方法去实现，而需要用compareTo或compare函数去判断。下面探讨一下TreeMultimap类的源码实现：

TreeMultimap里面一共有两个构造函数，实现如下:

```
private transient Comparator<? super K> keyComparator;
private transient Comparator<? super V> valueComparator;

TreeMultimap(Comparator<? super K> keyComparator,
                    Comparator<? super V> valueComparator) {
          super(new TreeMap<K, Collection<V>>(keyComparator));
          this.keyComparator = keyComparator;
          this.valueComparator = valueComparator;
}

private TreeMultimap(Comparator<? super K> keyComparator,
                     Comparator<? super V> valueComparator,
                     Multimap<? extends K, ? extends V> multimap) {
          this(keyComparator, valueComparator);
          putAll(multimap);
}
```

第一个构造函数有两个参数，keyComparator是指定TreeMultimap中key是按照什么比较算法去排序的；同理valueComparator是指定TreeMultimap中同一个key中的value是按照什么算法去排序的。第二个构造函数多了一个multimap参数，代表初始化的时候就带有数据，并将multimap中的数据按照keyComparator和valueComparator排序算法复制到TreeMultimap对象中。

虽然TreeMultimap中有两个构造函数，但是用户都不能直接调用，TreeMultimap类为我们提供了以下三个create()方法来构造一个TreeMultimap对象，实现如下：

```
public static <K extends Comparable, V extends Comparable>
   TreeMultimap<K, V> create() {
      return new TreeMultimap<K, V>(Ordering.natural(), Ordering.natural());
}

public static <K, V> TreeMultimap<K, V> create(
      Comparator<? super K> keyComparator,
      Comparator<? super V> valueComparator) {
```

```
        return new TreeMultimap<K, V>(checkNotNull(keyComparator),
                checkNotNull(valueComparator));
}

public static <K extends Comparable, V extends Comparable>
TreeMultimap<K, V> create(Multimap<? extends K, ? extends V> multimap) {
        return new TreeMultimap<K, V>(Ordering.natural(), Ordering.natural(),
                multimap);
}
```

我们可以通过上面三个create()函数构造出一个TreeMultimap对象，第一个create()函数是利用自然排序对TreeMultimap对象中的key和value进行排序。其余的都是按照用户输入的进行排序。(待续)

本博客文章除特别声明，全部都是原创！
转载本文请加上：转载自过往记忆（https://www.iteblog.com/）
本文链接:【】（）