

# TreeSet

概念:

有序(但得实现Comparable重写compareTo方法), TreeSet主要依靠compareTo实现的有序

若不实现Comparable则无法正常使用.

其他特点:

唯一, 可以存储单个null, 底层为TreeMap[红黑树实现]

常用方法:

```
public SortedSet<E> subSet(E fromElement, E toElement)
```

截取集合中指定元素到指定元素之间的数据

```
TreeSet<Integer> hashSet = new TreeSet();
```

```
for (int i = 0; i < 100; i++) {
```

```
    hashSet.add(i);
```

```
}
```

```
System.out.println(hashSet.subSet(1,5)); //结果为 1 2 3 4
```

```
public SortedSet<E> tailSet(E fromElement)
```

返回此集合的部分的视图, 其元素大于或等于fromElement

```
TreeSet<Integer> hashSet = new TreeSet();
```

```
for (int i = 0; i < 100; i++) {
```

```
    hashSet.add(i);
```

```
}
```

```
System.out.println(hashSet.tailSet(90)); //结果为大于90的打印
```

```
public SortedSet<E> headSet(E toElement)
```

返回此集合的部分的视图, 其元素严格小于toElement

```
TreeSet<Integer> hashSet = new TreeSet();
```

```
for (int i = 0; i < 100; i++) {
```

```
    hashSet.add(i);
```

```
}
```

```
System.out.println(hashSet.tailSet(10)); //结果为小于10的打印
```