

# Set

`java.util.Set` 不包含重复元素的集合、不能保证存储的顺序、只允许有一个 `null`

```
1 public interface Set<E>
2 extends Collection<E>
```

抽象方法，都是继承自 `java.util.Collection` 接口，在此不做重复描述。

`Set` 集合的实现类有很多，在此我们重点了解 `HashSet`、`TreeSet`、`LinkedHashSet`

## TreeSet

```
1 public class TreeSet<E>
2 extends AbstractSet<E>
3 implements NavigableSet<E>, Cloneable, Serializable
```

- 非线程安全
- 值必须可比较(元素实现 `Comparable` 接口、传递 比较器 `Comparator` 对象)
- 不能存 `null`
- 判断是否是重复元素，是按照自然比较/比较器进行比较

就是说`a.compareTo(b) == 0`,如果是 `true`，那么 `add(a)` 之后的 `add(b)` 将会返回 `false`，也就是添加失败

常用的构造方法

方法名	描述
<code>TreeSet()</code>	构造一个新的空 <code>TreeSet</code> 集合，根据其元素的自然顺序进行排序
<code>TreeSet(Comparator&lt;? super E&gt; comparator)</code>	构造一个新的空 <code>TreeSet</code> 集合，根据指定的比较器进行排序
<code>TreeSet(Collection&lt;? extends E&gt; c)</code>	构造一个新的 <code>TreeSet</code> 集合，，该 <code>TreeSet</code> 集合包含指定集合中的元素，并根据其元素的自然顺序进行排序。

常用方法

( `Collection` 接口的方法不在此赘述)

方法名	返回值	描述
<code>ceiling(E e)</code>	<code>E</code>	返回此集合中大于或等于给定元素的最小元素，如果没有这样的元素，则返回 <code>null</code> 。
<code>first()</code>	<code>E</code>	返回当前在此集合中的第一个（最低的）元素。
<code>floor(E e)</code>	<code>E</code>	返回此集合中小于或等于给定元素的最大元素，如果没有这样的元素，则返回 <code>null</code> 。
<code>headSet(E toElement)</code>	<code>SortedSet&lt;E&gt;</code>	返回此集合中元素严格小于 <code>toElement</code> 的部分的视图。
<code>higher(e)</code>	<code>E</code>	返回此集合中严格大于给定元素的最小元素，如果没有这样的元素，则返回 <code>null</code> 。
<code>last()</code>	<code>E</code>	返回当前在此集合中的最后一个（最高的）元素。
<code>lower(e)</code>	<code>E</code>	返回此集合中严格小于给定元素的最大元素，如果没有这样的元素，则返回 <code>null</code> 。
<code>pollFirst()</code>	<code>E</code>	检索并删除第一个（最低的）元素，如果此集合为空，则返回 <code>null</code>
<code>pollLast()</code>	<code>E</code>	检索并删除最后一个（最高）元素，如果此集合为空，则返回 <code>null</code>
<code>tailSet(E fromElement)</code>	<code>SortedSet&lt;E&gt;</code>	返回此集合中元素严格大于或等于 <code>fromElement</code> 的部分的视图。

```
1 TreeSet<String> set = new TreeSet(List.of("null", "a", "a", "b", "c", "e", "f", "g"));
2
3 System.out.println(set); // [a, b, c, e, f, g, null]
4
5 // 返回此集合中大于或等于给定元素的最小元素，如果没有这样的元素，则返回null。
6 String ceiling = set.ceiling("d");
7 System.out.println(ceiling); // e
8
9 // 返回当前在此集合中的第一个（最低的）元素。
10 String first = set.first();
11 System.out.println(first); // a
12
13 // 返回此集合中小于或等于给定元素的最大元素，如果没有这样的元素，则返回null。
```

```

14 String floor = set.floor("d");
15 System.out.println(floor); // c
16
17 // 返回此集合中元素严格小于toElement的部分的视图。
18 SortedSet<String> headSet = set.headSet("c");
19 System.out.println(headSet); // a, b
20
21 // 返回此集合中严格大于给定元素的最小元素，如果没有这样的元素，则返回null。
22 String higher = set.higher("c");
23 System.out.println(higher); // e
24
25 // 返回此集合中元素严格大于或等于fromElement的部分的视图。
26 SortedSet<String> tailSet = set.tailSet("c");
27 System.out.println(tailSet); // c, e, f, g, null
28
29 // 迭代
30 for (Object obj : set){
31     System.out.println(obj);
32 }

```

## HashSet

```

1 public class HashSet<E>
2     extends AbstractSet<E>
3     implements Set<E>, Cloneable, Serializable

```

实现了 `Set` 接口，底层实现是 `HashMap`。不保证迭代顺序，允许 `null` 元素

- 非线程安全的
- 如果 `add` 的值已存在( `equals` 方法返回 `true` ,基本数据类型自动装箱)返回 `false`
- 如果 `HashSet` 中存的是对象，需要重写此对象类中的 `equals` 和 `hashCode()` 方法

构造方法

方法名	描述
<code>HashSet()</code>	构造一个新的空集合; 底层实现 <code>HashMap</code> 实例具有默认初始容量（16）和负载因子（0.75）。
<code>HashSet(Collection&lt;? extends E&gt; c)</code>	构造一个包含指定集合中的元素的新集合

方法名	描述
<code>HashSet(int initialCapacity)</code>	构造一个新的空集合,默认初始容量(initialCapacity)和负载因子(0.75)
<code>HashSet(int initialCapacity, float loadFactor)</code>	构造一个新的空集合; 底层HashMap实例具有指定的初始容量和指定的负载因子

常用方法:

略 (自己补充)

## LinkedHashSet

```
1 public class LinkedHashSet<E>  
2     extends HashSet<E>  
3     implements Set<E>, Cloneable, java.io.Serializable
```

- 哈希表和双向链表实现的 `Set` 接口
- 具有可预测的迭代次序(有序)
- 内部实现是 `LinkedHashMap` , 顺序是插入顺序