java.util

类 HashSet<E>

java.lang.Object

继承者 java.util.AbstractCollection<E>

继承者 java.util.AbstractSet<E>

继承者 java.util.HashSet<E>

类型参数：

E - 此 set 所维护的元素的类型

所有已实现的接口：

Serializable, Cloneable, Iterable<E>, Collection<E>, Set<E>

直接已知子类：

JobStateReasons, LinkedHashSet

--------------------------------------------------------------------------------

public class HashSet<E>extends AbstractSet<E>implements Set<E>, Cloneable, Serializable此类实现 Set 接口，由哈希表（实际上是一个 HashMap 实例）支持。它不保证 set 的迭代顺序；特别是它不保证该顺序恒久不变。此类允许使用 null 元素。

此类为基本操作提供了稳定性能，这些基本操作包括 add、remove、contains 和 size，假定哈希函数将这些元素正确地分布在桶中。对此 set 进行迭代所需的时间与 HashSet 实例的大小（元素的数量）和底层 HashMap 实例（桶的数量）的“容量”的和成比例。因此，如果迭代性能很重要，则不要将初始容量设置得太高（或将加载因子设置得太低）。

注意，此实现不是同步的。如果多个线程同时访问一个哈希 set，而其中至少一个线程修改了该 set，那么它必须 保持外部同步。这通常是通过对自然封装该 set 的对象执行同步操作来完成的。如果不存在这样的对象，则应该使用 Collections.synchronizedSet 方法来“包装” set。最好在创建时完成这一操作，以防止对该 set 进行意外的不同步访问：

Set s = Collections.synchronizedSet(new HashSet(...));此类的 iterator 方法返回的迭代器是快速失败 的：在创建迭代器之后，如果对 set 进行修改，除非通过迭代器自身的 remove 方法，否则在任何时间以任何方式对其进行修改，Iterator 都将抛出 ConcurrentModificationException。因此，面对并发的修改，迭代器很快就会完全失败，而不冒将来在某个不确定时间发生任意不确定行为的风险。

注意，迭代器的快速失败行为无法得到保证，因为一般来说，不可能对是否出现不同步并发修改做出任何硬性保证。快速失败迭代器在尽最大努力抛出 ConcurrentModificationException。因此，为提高这类迭代器的正确性而编写一个依赖于此异常的程序是错误做法：迭代器的快速失败行为应该仅用于检测 bug。

此类是 Java Collections Framework 的成员。

从以下版本开始：

1.2

另请参见：

Collection, Set, TreeSet, HashMap, 序列化表格

--------------------------------------------------------------------------------

构造方法摘要

HashSet()

构造一个新的空 set，其底层 HashMap 实例的默认初始容量是 16，加载因子是 0.75。

HashSet(Collection<? extends E> c)

构造一个包含指定 collection 中的元素的新 set。

HashSet(int initialCapacity)

构造一个新的空 set，其底层 HashMap 实例具有指定的初始容量和默认的加载因子（0.75）。

HashSet(int initialCapacity, float loadFactor)

构造一个新的空 set，其底层 HashMap 实例具有指定的初始容量和指定的加载因子。

方法摘要

boolean add(E e)

如果此 set 中尚未包含指定元素，则添加指定元素。

void clear()

从此 set 中移除所有元素。

Object clone()

返回此 HashSet 实例的浅表副本：并没有复制这些元素本身。

boolean contains(Object o)

如果此 set 包含指定元素，则返回 true。

boolean isEmpty()

如果此 set 不包含任何元素，则返回 true。

Iterator<E> iterator()

返回对此 set 中元素进行迭代的迭代器。

boolean remove(Object o)

如果指定元素存在于此 set 中，则将其移除。

int size()

返回此 set 中的元素的数量（set 的容量）。

从类 java.util.AbstractSet 继承的方法

equals, hashCode, removeAll

从类 java.util.AbstractCollection 继承的方法

addAll, containsAll, retainAll, toArray, toArray, toString

从类 java.lang.Object 继承的方法

finalize, getClass, notify, notifyAll, wait, wait, wait

从接口 java.util.Set 继承的方法

addAll, containsAll, equals, hashCode, removeAll, retainAll, toArray, toArray

类 TreeSet<E>

java.lang.Object

继承者 java.util.AbstractCollection<E>

继承者 java.util.AbstractSet<E>

继承者 java.util.TreeSet<E>

类型参数：

E - 此 set 维护的元素的类型

所有已实现的接口：

Serializable, Cloneable, Iterable<E>, Collection<E>, NavigableSet<E>, Set<E>, SortedSet<E>

--------------------------------------------------------------------------------

public class TreeSet<E>extends AbstractSet<E>implements NavigableSet<E>, Cloneable, Serializable基于 TreeMap 的 NavigableSet 实现。使用元素的自然顺序对元素进行排序，或者根据创建 set 时提供的 Comparator 进行排序，具体取决于使用的构造方法。

此实现为基本操作（add、remove 和 contains）提供受保证的 log(n) 时间开销。

注意，如果要正确实现 Set 接口，则 set 维护的顺序（无论是否提供了显式比较器）必须与 equals 一致。（关于与 equals 一致 的精确定义，请参阅 Comparable 或 Comparator。）这是因为 Set 接口是按照 equals 操作定义的，但 TreeSet 实例使用它的 compareTo（或 compare）方法对所有元素进行比较，因此从 set 的观点来看，此方法认为相等的两个元素就是相等的。即使 set 的顺序与 equals 不一致，其行为也是 定义良好的；它只是违背了 Set 接口的常规协定。

注意，此实现不是同步的。如果多个线程同时访问一个 TreeSet，而其中至少一个线程修改了该 set，那么它必须 外部同步。这一般是通过对自然封装该 set 的对象执行同步操作来完成的。如果不存在这样的对象，则应该使用 Collections.synchronizedSortedSet 方法来“包装”该 set。此操作最好在创建时进行，以防止对 set 的意外非同步访问：

SortedSet s = Collections.synchronizedSortedSet(new TreeSet(...));此类的 iterator 方法返回的迭代器是快速失败 的：在创建迭代器之后，如果从结构上对 set 进行修改，除非通过迭代器自身的 remove 方法，否则在其他任何时间以任何方式进行修改都将导致迭代器抛出 ConcurrentModificationException。因此，对于并发的修改，迭代器很快就完全失败，而不会冒着在将来不确定的时间发生不确定行为的风险。

注意，迭代器的快速失败行为无法得到保证，一般来说，存在不同步的并发修改时，不可能作出任何肯定的保证。快速失败迭代器尽最大努力抛出 ConcurrentModificationException。因此，编写依赖于此异常的程序的做法是错误的，正确做法是：迭代器的快速失败行为应该仅用于检测 bug。

此类是 Java Collections Framework 的成员。

从以下版本开始：

1.2

另请参见：

Collection, Set, HashSet, Comparable, Comparator, TreeMap, 序列化表格

--------------------------------------------------------------------------------

构造方法摘要

TreeSet()

构造一个新的空 set，该 set 根据其元素的自然顺序进行排序。

TreeSet(Collection<? extends E> c)

构造一个包含指定 collection 元素的新 TreeSet，它按照其元素的自然顺序进行排序。

TreeSet(Comparator<? super E> comparator)

构造一个新的空 TreeSet，它根据指定比较器进行排序。

TreeSet(SortedSet<E> s)

构造一个与指定有序 set 具有相同映射关系和相同排序的新 TreeSet。

方法摘要

boolean add(E e)

将指定的元素添加到此 set（如果该元素尚未存在于 set 中）。

boolean addAll(Collection<? extends E> c)

将指定 collection 中的所有元素添加到此 set 中。

E ceiling(E e)

返回此 set 中大于等于给定元素的最小元素；如果不存在这样的元素，则返回 null。

void clear()

移除此 set 中的所有元素。

Object clone()

返回 TreeSet 实例的浅表副本。

Comparator<? super E> comparator()

返回对此 set 中的元素进行排序的比较器；如果此 set 使用其元素的自然顺序，则返回 null。

boolean contains(Object o)

如果此 set 包含指定的元素，则返回 true。

Iterator<E> descendingIterator()

返回在此 set 元素上按降序进行迭代的迭代器。

NavigableSet<E> descendingSet()

返回此 set 中所包含元素的逆序视图。

E first()

返回此 set 中当前第一个（最低）元素。

E floor(E e)

返回此 set 中小于等于给定元素的最大元素；如果不存在这样的元素，则返回 null。

SortedSet<E> headSet(E toElement)

返回此 set 的部分视图，其元素严格小于 toElement。

NavigableSet<E> headSet(E toElement, boolean inclusive)

返回此 set 的部分视图，其元素小于（或等于，如果 inclusive 为 true）toElement。

E higher(E e)

返回此 set 中严格大于给定元素的最小元素；如果不存在这样的元素，则返回 null。

boolean isEmpty()

如果此 set 不包含任何元素，则返回 true。

Iterator<E> iterator()

返回在此 set 中的元素上按升序进行迭代的迭代器。

E last()

返回此 set 中当前最后一个（最高）元素。

E lower(E e)

返回此 set 中严格小于给定元素的最大元素；如果不存在这样的元素，则返回 null。

E pollFirst()

获取并移除第一个（最低）元素；如果此 set 为空，则返回 null。

E pollLast()

获取并移除最后一个（最高）元素；如果此 set 为空，则返回 null。

boolean remove(Object o)

将指定的元素从 set 中移除（如果该元素存在于此 set 中）。

int size()

返回 set 中的元素数（set 的容量）。

NavigableSet<E> subSet(E fromElement, boolean fromInclusive, E toElement, boolean toInclusive)

返回此 set 的部分视图，其元素范围从 fromElement 到 toElement。

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

返回此 set 的部分视图，其元素从 fromElement（包括）到 toElement（不包括）。

SortedSet<E> tailSet(E fromElement)

返回此 set 的部分视图，其元素大于等于 fromElement。

NavigableSet<E> tailSet(E fromElement, boolean inclusive)

返回此 set 的部分视图，其元素大于（或等于，如果 inclusive 为 true）fromElement。

从类 java.util.AbstractSet 继承的方法

equals, hashCode, removeAll

从类 java.util.AbstractCollection 继承的方法

containsAll, retainAll, toArray, toArray, toString

从类 java.lang.Object 继承的方法

finalize, getClass, notify, notifyAll, wait, wait, wait

从接口 java.util.Set 继承的方法

containsAll, equals, hashCode, removeAll, retainAll, toArray, toArray