| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AbstractSet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/AbstractSequentialList.html)   [**NEXT CLASS**](http://docs.google.com/java/util/ArrayDeque.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/AbstractSet.html)    [**NO FRAMES**](http://docs.google.com/AbstractSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

## **java.util**

Class AbstractSet<E>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.util.AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<E>  
 **java.util.AbstractSet<E>**

**Type Parameters:**E - the type of elements maintained by this set **All Implemented Interfaces:** [Iterable](http://docs.google.com/java/lang/Iterable.html)<E>, [Collection](http://docs.google.com/java/util/Collection.html)<E>, [Set](http://docs.google.com/java/util/Set.html)<E> **Direct Known Subclasses:** [ConcurrentSkipListSet](http://docs.google.com/java/util/concurrent/ConcurrentSkipListSet.html), [CopyOnWriteArraySet](http://docs.google.com/java/util/concurrent/CopyOnWriteArraySet.html), [EnumSet](http://docs.google.com/java/util/EnumSet.html), [HashSet](http://docs.google.com/java/util/HashSet.html), [TreeSet](http://docs.google.com/java/util/TreeSet.html)

public abstract class **AbstractSet<E>**extends [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<E>implements [Set](http://docs.google.com/java/util/Set.html)<E>

This class provides a skeletal implementation of the Set interface to minimize the effort required to implement this interface.

The process of implementing a set by extending this class is identical to that of implementing a Collection by extending AbstractCollection, except that all of the methods and constructors in subclasses of this class must obey the additional constraints imposed by the Set interface (for instance, the add method must not permit addition of multiple instances of an object to a set).

Note that this class does not override any of the implementations from the AbstractCollection class. It merely adds implementations for equals and hashCode.

This class is a member of the  [Java Collections Framework](http://docs.google.com/technotes/guides/collections/index.html).

**Since:** 1.2 **See Also:**[Collection](http://docs.google.com/java/util/Collection.html), [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html), [Set](http://docs.google.com/java/util/Set.html)

| **Constructor Summary** | |
| --- | --- |
| protected | [**AbstractSet**](http://docs.google.com/java/util/AbstractSet.html#AbstractSet())()            Sole constructor. |

| **Method Summary** | |
| --- | --- |
| boolean | [**equals**](http://docs.google.com/java/util/AbstractSet.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Compares the specified object with this set for equality. |
| int | [**hashCode**](http://docs.google.com/java/util/AbstractSet.html#hashCode())()            Returns the hash code value for this set. |
| boolean | [**removeAll**](http://docs.google.com/java/util/AbstractSet.html#removeAll(java.util.Collection))([Collection](http://docs.google.com/java/util/Collection.html)<?> c)            Removes from this set all of its elements that are contained in the specified collection (optional operation). |

| **Methods inherited from class java.util.**[**AbstractCollection**](http://docs.google.com/java/util/AbstractCollection.html) |
| --- |
| [add](http://docs.google.com/java/util/AbstractCollection.html#add(E)), [addAll](http://docs.google.com/java/util/AbstractCollection.html#addAll(java.util.Collection)), [clear](http://docs.google.com/java/util/AbstractCollection.html#clear()), [contains](http://docs.google.com/java/util/AbstractCollection.html#contains(java.lang.Object)), [containsAll](http://docs.google.com/java/util/AbstractCollection.html#containsAll(java.util.Collection)), [isEmpty](http://docs.google.com/java/util/AbstractCollection.html#isEmpty()), [iterator](http://docs.google.com/java/util/AbstractCollection.html#iterator()), [remove](http://docs.google.com/java/util/AbstractCollection.html#remove(java.lang.Object)), [retainAll](http://docs.google.com/java/util/AbstractCollection.html#retainAll(java.util.Collection)), [size](http://docs.google.com/java/util/AbstractCollection.html#size()), [toArray](http://docs.google.com/java/util/AbstractCollection.html#toArray()), [toArray](http://docs.google.com/java/util/AbstractCollection.html#toArray(T%5B%5D)), [toString](http://docs.google.com/java/util/AbstractCollection.html#toString()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Methods inherited from interface java.util.**[**Set**](http://docs.google.com/java/util/Set.html) |
| --- |
| [add](http://docs.google.com/java/util/Set.html#add(E)), [addAll](http://docs.google.com/java/util/Set.html#addAll(java.util.Collection)), [clear](http://docs.google.com/java/util/Set.html#clear()), [contains](http://docs.google.com/java/util/Set.html#contains(java.lang.Object)), [containsAll](http://docs.google.com/java/util/Set.html#containsAll(java.util.Collection)), [isEmpty](http://docs.google.com/java/util/Set.html#isEmpty()), [iterator](http://docs.google.com/java/util/Set.html#iterator()), [remove](http://docs.google.com/java/util/Set.html#remove(java.lang.Object)), [retainAll](http://docs.google.com/java/util/Set.html#retainAll(java.util.Collection)), [size](http://docs.google.com/java/util/Set.html#size()), [toArray](http://docs.google.com/java/util/Set.html#toArray()), [toArray](http://docs.google.com/java/util/Set.html#toArray(T%5B%5D)) |

| **Constructor Detail** |
| --- |

### AbstractSet

protected **AbstractSet**()

Sole constructor. (For invocation by subclass constructors, typically implicit.)

| **Method Detail** |
| --- |

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) o)

Compares the specified object with this set for equality. Returns true if the given object is also a set, the two sets have the same size, and every member of the given set is contained in this set. This ensures that the equals method works properly across different implementations of the Set interface.

This implementation first checks if the specified object is this set; if so it returns true. Then, it checks if the specified object is a set whose size is identical to the size of this set; if not, it returns false. If so, it returns containsAll((Collection) o).

**Specified by:**[equals](http://docs.google.com/java/util/Collection.html#equals(java.lang.Object)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/AbstractSet.html)>**Specified by:**[equals](http://docs.google.com/java/util/Set.html#equals(java.lang.Object)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/AbstractSet.html)>**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**o - object to be compared for equality with this set **Returns:**true if the specified object is equal to this set**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Returns the hash code value for this set. The hash code of a set is defined to be the sum of the hash codes of the elements in the set, where the hash code of a null element is defined to be zero. This ensures that s1.equals(s2) implies that s1.hashCode()==s2.hashCode() for any two sets s1 and s2, as required by the general contract of [Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()).

This implementation iterates over the set, calling the hashCode method on each element in the set, and adding up the results.

**Specified by:**[hashCode](http://docs.google.com/java/util/Collection.html#hashCode()) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/AbstractSet.html)>**Specified by:**[hashCode](http://docs.google.com/java/util/Set.html#hashCode()) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/AbstractSet.html)>**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**the hash code value for this set**See Also:**[Object.equals(Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Set.equals(Object)](http://docs.google.com/java/util/Set.html#equals(java.lang.Object))

### removeAll

public boolean **removeAll**([Collection](http://docs.google.com/java/util/Collection.html)<?> c)

Removes from this set all of its elements that are contained in the specified collection (optional operation). If the specified collection is also a set, this operation effectively modifies this set so that its value is the *asymmetric set difference* of the two sets.

This implementation determines which is the smaller of this set and the specified collection, by invoking the size method on each. If this set has fewer elements, then the implementation iterates over this set, checking each element returned by the iterator in turn to see if it is contained in the specified collection. If it is so contained, it is removed from this set with the iterator's remove method. If the specified collection has fewer elements, then the implementation iterates over the specified collection, removing from this set each element returned by the iterator, using this set's remove method.

Note that this implementation will throw an UnsupportedOperationException if the iterator returned by the iterator method does not implement the remove method.

**Specified by:**[removeAll](http://docs.google.com/java/util/Collection.html#removeAll(java.util.Collection)) in interface [Collection](http://docs.google.com/java/util/Collection.html)<[E](http://docs.google.com/java/util/AbstractSet.html)>**Specified by:**[removeAll](http://docs.google.com/java/util/Set.html#removeAll(java.util.Collection)) in interface [Set](http://docs.google.com/java/util/Set.html)<[E](http://docs.google.com/java/util/AbstractSet.html)>**Overrides:**[removeAll](http://docs.google.com/java/util/AbstractCollection.html#removeAll(java.util.Collection)) in class [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html)<[E](http://docs.google.com/java/util/AbstractSet.html)> **Parameters:**c - collection containing elements to be removed from this set **Returns:**true if this set changed as a result of the call **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the removeAll operation is not supported by this set [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the class of an element of this set is incompatible with the specified collection (optional) [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if this set contains a null element and the specified collection does not permit null elements (optional), or if the specified collection is null**See Also:**[AbstractCollection.remove(Object)](http://docs.google.com/java/util/AbstractCollection.html#remove(java.lang.Object)), [AbstractCollection.contains(Object)](http://docs.google.com/java/util/AbstractCollection.html#contains(java.lang.Object))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AbstractSet.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/util/AbstractSequentialList.html)   [**NEXT CLASS**](http://docs.google.com/java/util/ArrayDeque.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/AbstractSet.html)    [**NO FRAMES**](http://docs.google.com/AbstractSet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).