JavaScript is disabled on your browser.

[Skip navigation links](#1fob9te)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/Bag.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/org/apache/commons/collections4/ArrayStack.html)
* [Next Class](http://docs.google.com/org/apache/commons/collections4/BagUtils.html)
* [Frames](http://docs.google.com/index.html?org/apache/commons/collections4/Bag.html)
* [No Frames](http://docs.google.com/Bag.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#3dy6vkm)

org.apache.commons.collections4

## Interface Bag<E>

* Type Parameters: E - the type of elements in this bag All Superinterfaces: [Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<E>, [Iterable](https://docs.oracle.com/javase/7/docs/api/java/lang/Iterable.html?is-external=true)<E> All Known Subinterfaces: [SortedBag](http://docs.google.com/org/apache/commons/collections4/SortedBag.html)<E> All Known Implementing Classes: [AbstractBagDecorator](http://docs.google.com/org/apache/commons/collections4/bag/AbstractBagDecorator.html), [AbstractMapBag](http://docs.google.com/org/apache/commons/collections4/bag/AbstractMapBag.html), [AbstractSortedBagDecorator](http://docs.google.com/org/apache/commons/collections4/bag/AbstractSortedBagDecorator.html), [CollectionBag](http://docs.google.com/org/apache/commons/collections4/bag/CollectionBag.html), [CollectionSortedBag](http://docs.google.com/org/apache/commons/collections4/bag/CollectionSortedBag.html), [HashBag](http://docs.google.com/org/apache/commons/collections4/bag/HashBag.html), [PredicatedBag](http://docs.google.com/org/apache/commons/collections4/bag/PredicatedBag.html), [PredicatedSortedBag](http://docs.google.com/org/apache/commons/collections4/bag/PredicatedSortedBag.html), [SynchronizedBag](http://docs.google.com/org/apache/commons/collections4/bag/SynchronizedBag.html), [SynchronizedSortedBag](http://docs.google.com/org/apache/commons/collections4/bag/SynchronizedSortedBag.html), [TransformedBag](http://docs.google.com/org/apache/commons/collections4/bag/TransformedBag.html), [TransformedSortedBag](http://docs.google.com/org/apache/commons/collections4/bag/TransformedSortedBag.html), [TreeBag](http://docs.google.com/org/apache/commons/collections4/bag/TreeBag.html), [UnmodifiableBag](http://docs.google.com/org/apache/commons/collections4/bag/UnmodifiableBag.html), [UnmodifiableSortedBag](http://docs.google.com/org/apache/commons/collections4/bag/UnmodifiableSortedBag.html)  
    
  public interface Bag<E>  
  extends [Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<E>  
  Defines a collection that counts the number of times an object appears in the collection.  
  Suppose you have a Bag that contains {a, a, b, c}. Calling [getCount(Object)](http://docs.google.com/org/apache/commons/collections4/Bag.html#getCount-java.lang.Object-) on a would return 2, while calling [uniqueSet()](http://docs.google.com/org/apache/commons/collections4/Bag.html#uniqueSet--) would return {a, b, c}.  
  *NOTE: This interface violates the* [*Collection*](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true) *contract.* The behavior specified in many of these methods is *not* the same as the behavior specified by Collection. The noncompliant methods are clearly marked with "(Violation)". Exercise caution when using a bag as a Collection.  
  This violation resulted from the original specification of this interface. In an ideal world, the interface would be changed to fix the problems, however it has been decided to maintain backwards compatibility instead.Since: 2.0

### Method SummaryAll Methods Instance Methods Abstract Methods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + boolean | * + [add](http://docs.google.com/org/apache/commons/collections4/Bag.html#add-E-)([E](http://docs.google.com/org/apache/commons/collections4/Bag.html) object) *(Violation)* Adds one copy of the specified object to the Bag. |
| * + boolean | * + [add](http://docs.google.com/org/apache/commons/collections4/Bag.html#add-E-int-)([E](http://docs.google.com/org/apache/commons/collections4/Bag.html) object, int nCopies) Adds nCopies copies of the specified object to the Bag. |
| * + boolean | * + [containsAll](http://docs.google.com/org/apache/commons/collections4/Bag.html#containsAll-java.util.Collection-)([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<?> coll) *(Violation)* Returns true if the bag contains all elements in the given collection, respecting cardinality. |
| * + int | * + [getCount](http://docs.google.com/org/apache/commons/collections4/Bag.html#getCount-java.lang.Object-)([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true) object) Returns the number of occurrences (cardinality) of the given object currently in the bag. |
| * + [Iterator](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> | * + [iterator](http://docs.google.com/org/apache/commons/collections4/Bag.html#iterator--)() Returns an [Iterator](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true) over the entire set of members, including copies due to cardinality. |
| * + boolean | * + [remove](http://docs.google.com/org/apache/commons/collections4/Bag.html#remove-java.lang.Object-)([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true) object) *(Violation)* Removes all occurrences of the given object from the bag. |
| * + boolean | * + [remove](http://docs.google.com/org/apache/commons/collections4/Bag.html#remove-java.lang.Object-int-)([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true) object, int nCopies) Removes nCopies copies of the specified object from the Bag. |
| * + boolean | * + [removeAll](http://docs.google.com/org/apache/commons/collections4/Bag.html#removeAll-java.util.Collection-)([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<?> coll) *(Violation)* Remove all elements represented in the given collection, respecting cardinality. |
| * + boolean | * + [retainAll](http://docs.google.com/org/apache/commons/collections4/Bag.html#retainAll-java.util.Collection-)([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<?> coll) *(Violation)* Remove any members of the bag that are not in the given collection, respecting cardinality. |
| * + int | * + [size](http://docs.google.com/org/apache/commons/collections4/Bag.html#size--)() Returns the total number of items in the bag across all types. |
| * + [Set](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> | * + [uniqueSet](http://docs.google.com/org/apache/commons/collections4/Bag.html#uniqueSet--)() Returns a [Set](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html?is-external=true) of unique elements in the Bag. |

### Methods inherited from interface java.util.[**Collection**](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)[addAll](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#addAll-java.util.Collection-), [clear](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#clear--), [contains](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#contains-java.lang.Object-), [equals](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#equals-java.lang.Object-), [hashCode](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#hashCode--), [isEmpty](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#isEmpty--), [parallelStream](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#parallelStream--), [removeIf](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#removeIf-java.util.function.Predicate-), [spliterator](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#spliterator--), [stream](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#stream--), [toArray](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#toArray--), [toArray](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#toArray-T:A-)

### Methods inherited from interface java.lang.[**Iterable**](https://docs.oracle.com/javase/7/docs/api/java/lang/Iterable.html?is-external=true)[forEach](https://docs.oracle.com/javase/7/docs/api/java/lang/Iterable.html?is-external=true#forEach-java.util.function.Consumer-)

### Method Detail

#### getCount int getCount([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true) object) Returns the number of occurrences (cardinality) of the given object currently in the bag. If the object does not exist in the bag, return 0.Parameters: object - the object to search for Returns: the number of occurrences of the object, zero if not found

#### add boolean add([E](http://docs.google.com/org/apache/commons/collections4/Bag.html) object) *(Violation)* Adds one copy of the specified object to the Bag. If the object is already in the [uniqueSet()](http://docs.google.com/org/apache/commons/collections4/Bag.html#uniqueSet--) then increment its count as reported by [getCount(Object)](http://docs.google.com/org/apache/commons/collections4/Bag.html#getCount-java.lang.Object-). Otherwise add it to the [uniqueSet()](http://docs.google.com/org/apache/commons/collections4/Bag.html#uniqueSet--) and report its count as 1. Since this method always increases the size of the bag, according to the [Collection.add(Object)](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#add-E-) contract, it should always return true. Since it sometimes returns false, this method violates the contract.Specified by: [add](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#add-E-) in interface [Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> Parameters: object - the object to add Returns: true if the object was not already in the uniqueSet

#### add boolean add([E](http://docs.google.com/org/apache/commons/collections4/Bag.html) object, int nCopies) Adds nCopies copies of the specified object to the Bag. If the object is already in the [uniqueSet()](http://docs.google.com/org/apache/commons/collections4/Bag.html#uniqueSet--) then increment its count as reported by [getCount(Object)](http://docs.google.com/org/apache/commons/collections4/Bag.html#getCount-java.lang.Object-). Otherwise add it to the [uniqueSet()](http://docs.google.com/org/apache/commons/collections4/Bag.html#uniqueSet--) and report its count as nCopies.Parameters: object - the object to add nCopies - the number of copies to add Returns: true if the object was not already in the uniqueSet

#### remove boolean remove([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true) object) *(Violation)* Removes all occurrences of the given object from the bag. This will also remove the object from the [uniqueSet()](http://docs.google.com/org/apache/commons/collections4/Bag.html#uniqueSet--). According to the [Collection.remove(Object)](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#remove-java.lang.Object-) method, this method should only remove the *first* occurrence of the given object, not *all* occurrences.Specified by: [remove](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#remove-java.lang.Object-) in interface [Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> Parameters: object - the object to remove Returns: true if this call changed the collection

#### remove boolean remove([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true) object, int nCopies) Removes nCopies copies of the specified object from the Bag. If the number of copies to remove is greater than the actual number of copies in the Bag, no error is thrown.Parameters: object - the object to remove nCopies - the number of copies to remove Returns: true if this call changed the collection

#### uniqueSet [Set](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> uniqueSet() Returns a [Set](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html?is-external=true) of unique elements in the Bag. Uniqueness constraints are the same as those in [Set](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html?is-external=true).Returns: the Set of unique Bag elements

#### size int size() Returns the total number of items in the bag across all types.Specified by: [size](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#size--) in interface [Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> Returns: the total size of the Bag

#### containsAll boolean containsAll([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<?> coll) *(Violation)* Returns true if the bag contains all elements in the given collection, respecting cardinality. That is, if the given collection coll contains n copies of a given object, calling [getCount(Object)](http://docs.google.com/org/apache/commons/collections4/Bag.html#getCount-java.lang.Object-) on that object must be >= n for all n in coll. The [Collection.containsAll(Collection)](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#containsAll-java.util.Collection-) method specifies that cardinality should *not* be respected; this method should return true if the bag contains at least one of every object contained in the given collection.Specified by: [containsAll](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#containsAll-java.util.Collection-) in interface [Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> Parameters: coll - the collection to check against Returns: true if the Bag contains all the collection

#### removeAll boolean removeAll([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<?> coll) *(Violation)* Remove all elements represented in the given collection, respecting cardinality. That is, if the given collection coll contains n copies of a given object, the bag will have n fewer copies, assuming the bag had at least n copies to begin with. The [Collection.removeAll(Collection)](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#removeAll-java.util.Collection-) method specifies that cardinality should *not* be respected; this method should remove *all* occurrences of every object contained in the given collection.Specified by: [removeAll](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#removeAll-java.util.Collection-) in interface [Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> Parameters: coll - the collection to remove Returns: true if this call changed the collection

#### retainAll boolean retainAll([Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<?> coll) *(Violation)* Remove any members of the bag that are not in the given collection, respecting cardinality. That is, if the given collection coll contains n copies of a given object and the bag has m > n copies, then delete m - n copies from the bag. In addition, if e is an object in the bag but !coll.contains(e), then remove e and any of its copies. The [Collection.retainAll(Collection)](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#retainAll-java.util.Collection-) method specifies that cardinality should *not* be respected; this method should keep *all* occurrences of every object contained in the given collection.Specified by: [retainAll](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#retainAll-java.util.Collection-) in interface [Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> Parameters: coll - the collection to retain Returns: true if this call changed the collection

#### iterator [Iterator](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> iterator() Returns an [Iterator](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html?is-external=true) over the entire set of members, including copies due to cardinality. This iterator is fail-fast and will not tolerate concurrent modifications.Specified by: [iterator](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true#iterator--) in interface [Collection](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> Specified by: [iterator](https://docs.oracle.com/javase/7/docs/api/java/lang/Iterable.html?is-external=true#iterator--) in interface [Iterable](https://docs.oracle.com/javase/7/docs/api/java/lang/Iterable.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/Bag.html)> Returns: iterator over all elements in the Bag

[Skip navigation links](#2xcytpi)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/Bag.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/org/apache/commons/collections4/ArrayStack.html)
* [Next Class](http://docs.google.com/org/apache/commons/collections4/BagUtils.html)
* [Frames](http://docs.google.com/index.html?org/apache/commons/collections4/Bag.html)
* [No Frames](http://docs.google.com/Bag.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* Constr |
* [Method](#3znysh7)
* Detail:
* Field |
* Constr |
* [Method](#3dy6vkm)

Copyright © 2001–2019 [The Apache Software Foundation](https://www.apache.org/). All rights reserved.