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/ComparatorChain.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/comparators/ComparableComparator.html)
* [Next Class](http://docs.google.com/org/apache/commons/collections4/comparators/FixedOrderComparator.html)
* [Frames](http://docs.google.com/index.html?org/apache/commons/collections4/comparators/ComparatorChain.html)
* [No Frames](http://docs.google.com/ComparatorChain.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#1t3h5sf) |
* [Method](#lnxbz9)

org.apache.commons.collections4.comparators

## Class ComparatorChain<E>

* [java.lang.Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)
  + org.apache.commons.collections4.comparators.ComparatorChain<E>
* Type Parameters: E - the type of objects compared by this comparator All Implemented Interfaces: [Serializable](https://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html?is-external=true), [Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<E>  
    
  public class ComparatorChain<E>  
  extends [Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)  
  implements [Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<E>, [Serializable](https://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html?is-external=true)  
  A ComparatorChain is a Comparator that wraps one or more Comparators in sequence. The ComparatorChain calls each Comparator in sequence until either 1) any single Comparator returns a non-zero result (and that result is then returned), or 2) the ComparatorChain is exhausted (and zero is returned). This type of sorting is very similar to multi-column sorting in SQL, and this class allows Java classes to emulate that kind of behaviour when sorting a List.  
  To further facilitate SQL-like sorting, the order of any single Comparator in the list can be reversed.  
  Calling a method that adds new Comparators or changes the ascend/descend sort *after compare(Object, Object) has been called* will result in an UnsupportedOperationException. However, *take care* to not alter the underlying List of Comparators or the BitSet that defines the sort order.  
  Instances of ComparatorChain are not synchronized. The class is not thread-safe at construction time, but it *is* thread-safe to perform multiple comparisons after all the setup operations are complete.Since: 2.0 See Also: [Serialized Form](http://docs.google.com/serialized-form.html#org.apache.commons.collections4.comparators.ComparatorChain)

### Constructor SummaryConstructors

|  |
| --- |
| * + Constructor and Description |
| * + [ComparatorChain](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#ComparatorChain--)() Construct a ComparatorChain with no Comparators. |
| * + [ComparatorChain](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#ComparatorChain-java.util.Comparator-)([Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator) Construct a ComparatorChain with a single Comparator, sorting in the forward order |
| * + [ComparatorChain](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#ComparatorChain-java.util.Comparator-boolean-)([Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator, boolean reverse) Construct a Comparator chain with a single Comparator, sorting in the given order |
| * + [ComparatorChain](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#ComparatorChain-java.util.List-)([List](https://docs.oracle.com/javase/7/docs/api/java/util/List.html?is-external=true)<[Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)>> list) Construct a ComparatorChain from the Comparators in the List. |
| * + [ComparatorChain](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#ComparatorChain-java.util.List-java.util.BitSet-)([List](https://docs.oracle.com/javase/7/docs/api/java/util/List.html?is-external=true)<[Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)>> list, [BitSet](https://docs.oracle.com/javase/7/docs/api/java/util/BitSet.html?is-external=true) bits) Construct a ComparatorChain from the Comparators in the given List. |

### Method SummaryAll Methods Instance Methods Concrete Methods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + void | * + [addComparator](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#addComparator-java.util.Comparator-)([Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator) Add a Comparator to the end of the chain using the forward sort order |
| * + void | * + [addComparator](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#addComparator-java.util.Comparator-boolean-)([Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator, boolean reverse) Add a Comparator to the end of the chain using the given sort order |
| * + int | * + [compare](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#compare-E-E-)([E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html) o1, [E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html) o2) Perform comparisons on the Objects as per Comparator.compare(o1,o2). |
| * + boolean | * + [equals](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#equals-java.lang.Object-)([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true) object) Returns true iff *that* Object is is a [Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true) whose ordering is known to be equivalent to mine. |
| * + int | * + [hashCode](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#hashCode--)() Implement a hash code for this comparator that is consistent with [equals](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#equals-java.lang.Object-). |
| * + boolean | * + [isLocked](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#isLocked--)() Determine if modifications can still be made to the ComparatorChain. |
| * + void | * + [setComparator](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#setComparator-int-java.util.Comparator-)(int index, [Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator) Replace the Comparator at the given index, maintaining the existing sort order. |
| * + void | * + [setComparator](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#setComparator-int-java.util.Comparator-boolean-)(int index, [Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator, boolean reverse) Replace the Comparator at the given index in the ComparatorChain, using the given sort order |
| * + void | * + [setForwardSort](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#setForwardSort-int-)(int index) Change the sort order at the given index in the ComparatorChain to a forward sort. |
| * + void | * + [setReverseSort](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#setReverseSort-int-)(int index) Change the sort order at the given index in the ComparatorChain to a reverse sort. |
| * + int | * + [size](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#size--)() Number of Comparators in the current ComparatorChain. |

### Methods inherited from class java.lang.[**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true)[clone](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#clone--), [finalize](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#finalize--), [getClass](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#getClass--), [notify](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#notify--), [notifyAll](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#notifyAll--), [toString](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#toString--), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#wait--), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#wait-long-), [wait](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#wait-long-int-)

### Methods inherited from interface java.util.[**Comparator**](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)[comparing](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#comparing-java.util.function.Function-), [comparing](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#comparing-java.util.function.Function-java.util.Comparator-), [comparingDouble](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#comparingDouble-java.util.function.ToDoubleFunction-), [comparingInt](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#comparingInt-java.util.function.ToIntFunction-), [comparingLong](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#comparingLong-java.util.function.ToLongFunction-), [naturalOrder](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#naturalOrder--), [nullsFirst](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#nullsFirst-java.util.Comparator-), [nullsLast](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#nullsLast-java.util.Comparator-), [reversed](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#reversed--), [reverseOrder](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#reverseOrder--), [thenComparing](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#thenComparing-java.util.Comparator-), [thenComparing](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#thenComparing-java.util.function.Function-), [thenComparing](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#thenComparing-java.util.function.Function-java.util.Comparator-), [thenComparingDouble](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#thenComparingDouble-java.util.function.ToDoubleFunction-), [thenComparingInt](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#thenComparingInt-java.util.function.ToIntFunction-), [thenComparingLong](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#thenComparingLong-java.util.function.ToLongFunction-)

### Constructor Detail

#### ComparatorChain public ComparatorChain() Construct a ComparatorChain with no Comparators. You must add at least one Comparator before calling the compare(Object,Object) method, or an UnsupportedOperationException is thrown

#### ComparatorChain public ComparatorChain([Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator) Construct a ComparatorChain with a single Comparator, sorting in the forward orderParameters: comparator - First comparator in the Comparator chain

#### ComparatorChain public ComparatorChain([Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator, boolean reverse) Construct a Comparator chain with a single Comparator, sorting in the given orderParameters: comparator - First Comparator in the ComparatorChain reverse - false = forward sort; true = reverse sort

#### ComparatorChain public ComparatorChain([List](https://docs.oracle.com/javase/7/docs/api/java/util/List.html?is-external=true)<[Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)>> list) Construct a ComparatorChain from the Comparators in the List. All Comparators will default to the forward sort order.Parameters: list - List of Comparators See Also: [ComparatorChain(List,BitSet)](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#ComparatorChain-java.util.List-java.util.BitSet-)

#### ComparatorChain public ComparatorChain([List](https://docs.oracle.com/javase/7/docs/api/java/util/List.html?is-external=true)<[Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)>> list, [BitSet](https://docs.oracle.com/javase/7/docs/api/java/util/BitSet.html?is-external=true) bits) Construct a ComparatorChain from the Comparators in the given List. The sort order of each column will be drawn from the given BitSet. When determining the sort order for Comparator at index *i* in the List, the ComparatorChain will call BitSet.get(*i*). If that method returns *false*, the forward sort order is used; a return value of *true* indicates reverse sort order.Parameters: list - List of Comparators. NOTE: This constructor does not perform a defensive copy of the list bits - Sort order for each Comparator. Extra bits are ignored, unless extra Comparators are added by another method.

### Method Detail

#### addComparator public void addComparator([Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator) Add a Comparator to the end of the chain using the forward sort orderParameters: comparator - Comparator with the forward sort order

#### addComparator public void addComparator([Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator, boolean reverse) Add a Comparator to the end of the chain using the given sort orderParameters: comparator - Comparator to add to the end of the chain reverse - false = forward sort order; true = reverse sort order

#### setComparator public void setComparator(int index, [Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator) throws [IndexOutOfBoundsException](https://docs.oracle.com/javase/7/docs/api/java/lang/IndexOutOfBoundsException.html?is-external=true) Replace the Comparator at the given index, maintaining the existing sort order.Parameters: index - index of the Comparator to replace comparator - Comparator to place at the given index Throws: [IndexOutOfBoundsException](https://docs.oracle.com/javase/7/docs/api/java/lang/IndexOutOfBoundsException.html?is-external=true) - if index < 0 or index >= size()

#### setComparator public void setComparator(int index, [Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> comparator, boolean reverse) Replace the Comparator at the given index in the ComparatorChain, using the given sort orderParameters: index - index of the Comparator to replace comparator - Comparator to set reverse - false = forward sort order; true = reverse sort order

#### setForwardSort public void setForwardSort(int index) Change the sort order at the given index in the ComparatorChain to a forward sort.Parameters: index - Index of the ComparatorChain

#### setReverseSort public void setReverseSort(int index) Change the sort order at the given index in the ComparatorChain to a reverse sort.Parameters: index - Index of the ComparatorChain

#### size public int size() Number of Comparators in the current ComparatorChain.Returns: Comparator count

#### isLocked public boolean isLocked() Determine if modifications can still be made to the ComparatorChain. ComparatorChains cannot be modified once they have performed a comparison.Returns: true = ComparatorChain cannot be modified; false = ComparatorChain can still be modified.

#### compare public int compare([E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html) o1, [E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html) o2) throws [UnsupportedOperationException](https://docs.oracle.com/javase/7/docs/api/java/lang/UnsupportedOperationException.html?is-external=true) Perform comparisons on the Objects as per Comparator.compare(o1,o2).Specified by: [compare](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#compare-T-T-) in interface [Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> Parameters: o1 - the first object to compare o2 - the second object to compare Returns: -1, 0, or 1 Throws: [UnsupportedOperationException](https://docs.oracle.com/javase/7/docs/api/java/lang/UnsupportedOperationException.html?is-external=true) - if the ComparatorChain does not contain at least one Comparator

#### hashCode public int hashCode() Implement a hash code for this comparator that is consistent with [equals](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html#equals-java.lang.Object-).Overrides: [hashCode](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#hashCode--) in class [Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true) Returns: a suitable hash code Since: 3.0

#### equals public boolean equals([Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true) object) Returns true iff *that* Object is is a [Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true) whose ordering is known to be equivalent to mine. This implementation returns true iff *object*.[getClass()](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#getClass--) equals this.getClass(), and the underlying comparators and order bits are equal. Subclasses may want to override this behavior to remain consistent with the [Comparator.equals(Object)](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#equals-java.lang.Object-) contract.Specified by: [equals](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true#equals-java.lang.Object-) in interface [Comparator](https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html?is-external=true)<[E](http://docs.google.com/org/apache/commons/collections4/comparators/ComparatorChain.html)> Overrides: [equals](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true#equals-java.lang.Object-) in class [Object](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html?is-external=true) Parameters: object - the object to compare with Returns: true if equal Since: 3.0

[Skip navigation links](#1pxezwc)

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/ComparatorChain.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/comparators/ComparableComparator.html)
* [Next Class](http://docs.google.com/org/apache/commons/collections4/comparators/FixedOrderComparator.html)
* [Frames](http://docs.google.com/index.html?org/apache/commons/collections4/comparators/ComparatorChain.html)
* [No Frames](http://docs.google.com/ComparatorChain.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* Field |
* [Constr](#3znysh7) |
* [Method](#2et92p0)
* Detail:
* Field |
* [Constr](#1t3h5sf) |
* [Method](#lnxbz9)

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