| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/java/util/Comparator.html) | **Use** | [**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   NEXT | [**FRAMES**](http://docs.google.com/index.html?java/util//class-useComparator.html)    [**NO FRAMES**](http://docs.google.com/Comparator.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

**Uses of Interface**

**java.util.Comparator**

| Packages that use [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| [**java.lang**](#3znysh7) | Provides classes that are fundamental to the design of the Java programming language. |
| [**java.text**](#2et92p0) | Provides classes and interfaces for handling text, dates, numbers, and messages in a manner independent of natural languages. |
| [**java.util**](#tyjcwt) | Contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (a string tokenizer, a random-number generator, and a bit array). |
| [**java.util.concurrent**](#3dy6vkm) | Utility classes commonly useful in concurrent programming. |
| [**javax.swing**](#1t3h5sf) | Provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms. |
| [**javax.swing.table**](#4d34og8) | Provides classes and interfaces for dealing with javax.swing.JTable. |

| Uses of [Comparator](http://docs.google.com/java/util/Comparator.html) in [java.lang](http://docs.google.com/java/lang/package-summary.html) | |
| --- | --- |

| Fields in [java.lang](http://docs.google.com/java/lang/package-summary.html) declared as [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| static [Comparator](http://docs.google.com/java/util/Comparator.html)<[String](http://docs.google.com/java/lang/String.html)> | **String.**[**CASE\_INSENSITIVE\_ORDER**](http://docs.google.com/java/lang/String.html#CASE_INSENSITIVE_ORDER)            A Comparator that orders String objects as by compareToIgnoreCase. |

| Uses of [Comparator](http://docs.google.com/java/util/Comparator.html) in [java.text](http://docs.google.com/java/text/package-summary.html) | |
| --- | --- |

| Classes in [java.text](http://docs.google.com/java/text/package-summary.html) that implement [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| class | [**Collator**](http://docs.google.com/java/text/Collator.html)            The Collator class performs locale-sensitive String comparison. |
| class | [**RuleBasedCollator**](http://docs.google.com/java/text/RuleBasedCollator.html)            The RuleBasedCollator class is a concrete subclass of Collator that provides a simple, data-driven, table collator. |

| Uses of [Comparator](http://docs.google.com/java/util/Comparator.html) in [java.util](http://docs.google.com/java/util/package-summary.html) | |
| --- | --- |

| Methods in [java.util](http://docs.google.com/java/util/package-summary.html) that return [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [K](http://docs.google.com/java/util/TreeMap.html)> | **TreeMap.**[**comparator**](http://docs.google.com/java/util/TreeMap.html#comparator())() |
| [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [E](http://docs.google.com/java/util/PriorityQueue.html)> | **PriorityQueue.**[**comparator**](http://docs.google.com/java/util/PriorityQueue.html#comparator())()            Returns the comparator used to order the elements in this queue, or null if this queue is sorted according to the [natural ordering](http://docs.google.com/java/lang/Comparable.html) of its elements. |
| [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [E](http://docs.google.com/java/util/TreeSet.html)> | **TreeSet.**[**comparator**](http://docs.google.com/java/util/TreeSet.html#comparator())() |
| [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [E](http://docs.google.com/java/util/SortedSet.html)> | **SortedSet.**[**comparator**](http://docs.google.com/java/util/SortedSet.html#comparator())()            Returns the comparator used to order the elements in this set, or null if this set uses the [natural ordering](http://docs.google.com/java/lang/Comparable.html) of its elements. |
| [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [K](http://docs.google.com/java/util/SortedMap.html)> | **SortedMap.**[**comparator**](http://docs.google.com/java/util/SortedMap.html#comparator())()            Returns the comparator used to order the keys in this map, or null if this map uses the [natural ordering](http://docs.google.com/java/lang/Comparable.html) of its keys. |
| static   | <T> [Comparator](http://docs.google.com/java/util/Comparator.html)<T> | | --- | | **Collections.**[**reverseOrder**](http://docs.google.com/java/util/Collections.html#reverseOrder())()            Returns a comparator that imposes the reverse of the *natural ordering* on a collection of objects that implement the Comparable interface. |
| static   | <T> [Comparator](http://docs.google.com/java/util/Comparator.html)<T> | | --- | | **Collections.**[**reverseOrder**](http://docs.google.com/java/util/Collections.html#reverseOrder(java.util.Comparator))([Comparator](http://docs.google.com/java/util/Comparator.html)<T> cmp)            Returns a comparator that imposes the reverse ordering of the specified comparator. |

| Methods in [java.util](http://docs.google.com/java/util/package-summary.html) with parameters of type [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| static   | <T> int | | --- | | **Collections.**[**binarySearch**](http://docs.google.com/java/util/Collections.html#binarySearch(java.util.List,%20T,%20java.util.Comparator))([List](http://docs.google.com/java/util/List.html)<? extends T> list, T key, [Comparator](http://docs.google.com/java/util/Comparator.html)<? super T> c)            Searches the specified list for the specified object using the binary search algorithm. |
| static   | <T> int | | --- | | **Arrays.**[**binarySearch**](http://docs.google.com/java/util/Arrays.html#binarySearch(T%5B%5D,%20int,%20int,%20T,%20java.util.Comparator))(T[] a, int fromIndex, int toIndex, T key, [Comparator](http://docs.google.com/java/util/Comparator.html)<? super T> c)            Searches a range of the specified array for the specified object using the binary search algorithm. |
| static   | <T> int | | --- | | **Arrays.**[**binarySearch**](http://docs.google.com/java/util/Arrays.html#binarySearch(T%5B%5D,%20T,%20java.util.Comparator))(T[] a, T key, [Comparator](http://docs.google.com/java/util/Comparator.html)<? super T> c)            Searches the specified array for the specified object using the binary search algorithm. |
| static   | <T> T | | --- | | **Collections.**[**max**](http://docs.google.com/java/util/Collections.html#max(java.util.Collection,%20java.util.Comparator))([Collection](http://docs.google.com/java/util/Collection.html)<? extends T> coll, [Comparator](http://docs.google.com/java/util/Comparator.html)<? super T> comp)            Returns the maximum element of the given collection, according to the order induced by the specified comparator. |
| static   | <T> T | | --- | | **Collections.**[**min**](http://docs.google.com/java/util/Collections.html#min(java.util.Collection,%20java.util.Comparator))([Collection](http://docs.google.com/java/util/Collection.html)<? extends T> coll, [Comparator](http://docs.google.com/java/util/Comparator.html)<? super T> comp)            Returns the minimum element of the given collection, according to the order induced by the specified comparator. |
| static   | <T> [Comparator](http://docs.google.com/java/util/Comparator.html)<T> | | --- | | **Collections.**[**reverseOrder**](http://docs.google.com/java/util/Collections.html#reverseOrder(java.util.Comparator))([Comparator](http://docs.google.com/java/util/Comparator.html)<T> cmp)            Returns a comparator that imposes the reverse ordering of the specified comparator. |
| static   | <T> void | | --- | | **Collections.**[**sort**](http://docs.google.com/java/util/Collections.html#sort(java.util.List,%20java.util.Comparator))([List](http://docs.google.com/java/util/List.html)<T> list, [Comparator](http://docs.google.com/java/util/Comparator.html)<? super T> c)            Sorts the specified list according to the order induced by the specified comparator. |
| static   | <T> void | | --- | | **Arrays.**[**sort**](http://docs.google.com/java/util/Arrays.html#sort(T%5B%5D,%20java.util.Comparator))(T[] a, [Comparator](http://docs.google.com/java/util/Comparator.html)<? super T> c)            Sorts the specified array of objects according to the order induced by the specified comparator. |
| static   | <T> void | | --- | | **Arrays.**[**sort**](http://docs.google.com/java/util/Arrays.html#sort(T%5B%5D,%20int,%20int,%20java.util.Comparator))(T[] a, int fromIndex, int toIndex, [Comparator](http://docs.google.com/java/util/Comparator.html)<? super T> c)            Sorts the specified range of the specified array of objects according to the order induced by the specified comparator. |

| Constructors in [java.util](http://docs.google.com/java/util/package-summary.html) with parameters of type [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| [**PriorityQueue**](http://docs.google.com/java/util/PriorityQueue.html#PriorityQueue(int,%20java.util.Comparator))(int initialCapacity, [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [E](http://docs.google.com/java/util/PriorityQueue.html)> comparator)            Creates a PriorityQueue with the specified initial capacity that orders its elements according to the specified comparator. |
| [**TreeMap**](http://docs.google.com/java/util/TreeMap.html#TreeMap(java.util.Comparator))([Comparator](http://docs.google.com/java/util/Comparator.html)<? super [K](http://docs.google.com/java/util/TreeMap.html)> comparator)            Constructs a new, empty tree map, ordered according to the given comparator. |
| [**TreeSet**](http://docs.google.com/java/util/TreeSet.html#TreeSet(java.util.Comparator))([Comparator](http://docs.google.com/java/util/Comparator.html)<? super [E](http://docs.google.com/java/util/TreeSet.html)> comparator)            Constructs a new, empty tree set, sorted according to the specified comparator. |

| Uses of [Comparator](http://docs.google.com/java/util/Comparator.html) in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) | |
| --- | --- |

| Methods in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) that return [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [E](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html)> | **PriorityBlockingQueue.**[**comparator**](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html#comparator())()            Returns the comparator used to order the elements in this queue, or null if this queue uses the [natural ordering](http://docs.google.com/java/lang/Comparable.html) of its elements. |
| [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [E](http://docs.google.com/java/util/concurrent/ConcurrentSkipListSet.html)> | **ConcurrentSkipListSet.**[**comparator**](http://docs.google.com/java/util/concurrent/ConcurrentSkipListSet.html#comparator())() |
| [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [K](http://docs.google.com/java/util/concurrent/ConcurrentSkipListMap.html)> | **ConcurrentSkipListMap.**[**comparator**](http://docs.google.com/java/util/concurrent/ConcurrentSkipListMap.html#comparator())() |

| Constructors in [java.util.concurrent](http://docs.google.com/java/util/concurrent/package-summary.html) with parameters of type [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| [**ConcurrentSkipListMap**](http://docs.google.com/java/util/concurrent/ConcurrentSkipListMap.html#ConcurrentSkipListMap(java.util.Comparator))([Comparator](http://docs.google.com/java/util/Comparator.html)<? super [K](http://docs.google.com/java/util/concurrent/ConcurrentSkipListMap.html)> comparator)            Constructs a new, empty map, sorted according to the specified comparator. |
| [**ConcurrentSkipListSet**](http://docs.google.com/java/util/concurrent/ConcurrentSkipListSet.html#ConcurrentSkipListSet(java.util.Comparator))([Comparator](http://docs.google.com/java/util/Comparator.html)<? super [E](http://docs.google.com/java/util/concurrent/ConcurrentSkipListSet.html)> comparator)            Constructs a new, empty set that orders its elements according to the specified comparator. |
| [**PriorityBlockingQueue**](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html#PriorityBlockingQueue(int,%20java.util.Comparator))(int initialCapacity, [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [E](http://docs.google.com/java/util/concurrent/PriorityBlockingQueue.html)> comparator)            Creates a PriorityBlockingQueue with the specified initial capacity that orders its elements according to the specified comparator. |

| Uses of [Comparator](http://docs.google.com/java/util/Comparator.html) in [javax.swing](http://docs.google.com/javax/swing/package-summary.html) | |
| --- | --- |

| Methods in [javax.swing](http://docs.google.com/javax/swing/package-summary.html) that return [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| protected  [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [Component](http://docs.google.com/java/awt/Component.html)> | **SortingFocusTraversalPolicy.**[**getComparator**](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#getComparator())()            Returns the Comparator which will be used to sort the Components in a focus traversal cycle. |
| [Comparator](http://docs.google.com/java/util/Comparator.html)<?> | **DefaultRowSorter.**[**getComparator**](http://docs.google.com/javax/swing/DefaultRowSorter.html#getComparator(int))(int column)            Returns the Comparator for the specified column. |

| Methods in [javax.swing](http://docs.google.com/javax/swing/package-summary.html) with parameters of type [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| protected  void | **SortingFocusTraversalPolicy.**[**setComparator**](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#setComparator(java.util.Comparator))([Comparator](http://docs.google.com/java/util/Comparator.html)<? super [Component](http://docs.google.com/java/awt/Component.html)> comparator)            Sets the Comparator which will be used to sort the Components in a focus traversal cycle. |
| void | **DefaultRowSorter.**[**setComparator**](http://docs.google.com/javax/swing/DefaultRowSorter.html#setComparator(int,%20java.util.Comparator))(int column, [Comparator](http://docs.google.com/java/util/Comparator.html)<?> comparator)            Sets the Comparator to use when sorting the specified column. |

| Constructors in [javax.swing](http://docs.google.com/javax/swing/package-summary.html) with parameters of type [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| [**SortingFocusTraversalPolicy**](http://docs.google.com/javax/swing/SortingFocusTraversalPolicy.html#SortingFocusTraversalPolicy(java.util.Comparator))([Comparator](http://docs.google.com/java/util/Comparator.html)<? super [Component](http://docs.google.com/java/awt/Component.html)> comparator)            Constructs a SortingFocusTraversalPolicy with the specified Comparator. |

| Uses of [Comparator](http://docs.google.com/java/util/Comparator.html) in [javax.swing.table](http://docs.google.com/javax/swing/table/package-summary.html) | |
| --- | --- |

| Methods in [javax.swing.table](http://docs.google.com/javax/swing/table/package-summary.html) that return [Comparator](http://docs.google.com/java/util/Comparator.html) | |
| --- | --- |
| [Comparator](http://docs.google.com/java/util/Comparator.html)<?> | **TableRowSorter.**[**getComparator**](http://docs.google.com/javax/swing/table/TableRowSorter.html#getComparator(int))(int column)            Returns the Comparator for the specified column. |

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/java/util/Comparator.html) | **Use** | [**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   NEXT | [**FRAMES**](http://docs.google.com/index.html?java/util//class-useComparator.html)    [**NO FRAMES**](http://docs.google.com/Comparator.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).