| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SortedMap.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/SimpleTimeZone.html)   [**NEXT CLASS**](http://docs.google.com/java/util/SortedSet.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/SortedMap.html)    [**NO FRAMES**](http://docs.google.com/SortedMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#1t3h5sf) |

## **java.util**

Interface SortedMap<K,V>

**Type Parameters:**K - the type of keys maintained by this mapV - the type of mapped values **All Superinterfaces:** [Map](http://docs.google.com/java/util/Map.html)<K,V> **All Known Subinterfaces:** [ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<K,V>, [NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<K,V> **All Known Implementing Classes:** [ConcurrentSkipListMap](http://docs.google.com/java/util/concurrent/ConcurrentSkipListMap.html), [TreeMap](http://docs.google.com/java/util/TreeMap.html)

public interface **SortedMap<K,V>**extends [Map](http://docs.google.com/java/util/Map.html)<K,V>

A [Map](http://docs.google.com/java/util/Map.html) that further provides a *total ordering* on its keys. The map is ordered according to the [natural ordering](http://docs.google.com/java/lang/Comparable.html) of its keys, or by a [Comparator](http://docs.google.com/java/util/Comparator.html) typically provided at sorted map creation time. This order is reflected when iterating over the sorted map's collection views (returned by the entrySet, keySet and values methods). Several additional operations are provided to take advantage of the ordering. (This interface is the map analogue of [SortedSet](http://docs.google.com/java/util/SortedSet.html).)

All keys inserted into a sorted map must implement the Comparable interface (or be accepted by the specified comparator). Furthermore, all such keys must be *mutually comparable*: k1.compareTo(k2) (or comparator.compare(k1, k2)) must not throw a ClassCastException for any keys k1 and k2 in the sorted map. Attempts to violate this restriction will cause the offending method or constructor invocation to throw a ClassCastException.

Note that the ordering maintained by a sorted map (whether or not an explicit comparator is provided) must be *consistent with equals* if the sorted map is to correctly implement the Map interface. (See the Comparable interface or Comparator interface for a precise definition of *consistent with equals*.) This is so because the Map interface is defined in terms of the equals operation, but a sorted map performs all key comparisons using its compareTo (or compare) method, so two keys that are deemed equal by this method are, from the standpoint of the sorted map, equal. The behavior of a tree map *is* well-defined even if its ordering is inconsistent with equals; it just fails to obey the general contract of the Map interface.

All general-purpose sorted map implementation classes should provide four "standard" constructors: 1) A void (no arguments) constructor, which creates an empty sorted map sorted according to the natural ordering of its keys. 2) A constructor with a single argument of type Comparator, which creates an empty sorted map sorted according to the specified comparator. 3) A constructor with a single argument of type Map, which creates a new map with the same key-value mappings as its argument, sorted according to the keys' natural ordering. 4) A constructor with a single argument of type SortedMap, which creates a new sorted map with the same key-value mappings and the same ordering as the input sorted map. There is no way to enforce this recommendation, as interfaces cannot contain constructors.

Note: several methods return submaps with restricted key ranges. Such ranges are *half-open*, that is, they include their low endpoint but not their high endpoint (where applicable). If you need a *closed range* (which includes both endpoints), and the key type allows for calculation of the successor of a given key, merely request the subrange from lowEndpoint to successor(highEndpoint). For example, suppose that m is a map whose keys are strings. The following idiom obtains a view containing all of the key-value mappings in m whose keys are between low and high, inclusive:

SortedMap<String, V> sub = m.subMap(low, high+"\0");

A similar technique can be used to generate an *open range* (which contains neither endpoint). The following idiom obtains a view containing all of the key-value mappings in m whose keys are between low and high, exclusive:

SortedMap<String, V> sub = m.subMap(low+"\0", high);

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

**Since:** 1.2 **See Also:**[Map](http://docs.google.com/java/util/Map.html), [TreeMap](http://docs.google.com/java/util/TreeMap.html), [SortedSet](http://docs.google.com/java/util/SortedSet.html), [Comparator](http://docs.google.com/java/util/Comparator.html), [Comparable](http://docs.google.com/java/lang/Comparable.html), [Collection](http://docs.google.com/java/util/Collection.html), [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html)

| **Nested Class Summary** | |
| --- | --- |

| **Nested classes/interfaces inherited from interface java.util.**[**Map**](http://docs.google.com/java/util/Map.html) |
| --- |
| [Map.Entry](http://docs.google.com/java/util/Map.Entry.html)<[K](http://docs.google.com/java/util/Map.Entry.html),[V](http://docs.google.com/java/util/Map.Entry.html)> |

| **Method Summary** | |
| --- | --- |
| [Comparator](http://docs.google.com/java/util/Comparator.html)<? super [K](http://docs.google.com/java/util/SortedMap.html)> | [**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. |
| [Set](http://docs.google.com/java/util/Set.html)<[Map.Entry](http://docs.google.com/java/util/Map.Entry.html)<[K](http://docs.google.com/java/util/SortedMap.html),[V](http://docs.google.com/java/util/SortedMap.html)>> | [**entrySet**](http://docs.google.com/java/util/SortedMap.html#entrySet())()            Returns a [Set](http://docs.google.com/java/util/Set.html) view of the mappings contained in this map. |
| [K](http://docs.google.com/java/util/SortedMap.html) | [**firstKey**](http://docs.google.com/java/util/SortedMap.html#firstKey())()            Returns the first (lowest) key currently in this map. |
| [SortedMap](http://docs.google.com/java/util/SortedMap.html)<[K](http://docs.google.com/java/util/SortedMap.html),[V](http://docs.google.com/java/util/SortedMap.html)> | [**headMap**](http://docs.google.com/java/util/SortedMap.html#headMap(K))([K](http://docs.google.com/java/util/SortedMap.html) toKey)            Returns a view of the portion of this map whose keys are strictly less than toKey. |
| [Set](http://docs.google.com/java/util/Set.html)<[K](http://docs.google.com/java/util/SortedMap.html)> | [**keySet**](http://docs.google.com/java/util/SortedMap.html#keySet())()            Returns a [Set](http://docs.google.com/java/util/Set.html) view of the keys contained in this map. |
| [K](http://docs.google.com/java/util/SortedMap.html) | [**lastKey**](http://docs.google.com/java/util/SortedMap.html#lastKey())()            Returns the last (highest) key currently in this map. |
| [SortedMap](http://docs.google.com/java/util/SortedMap.html)<[K](http://docs.google.com/java/util/SortedMap.html),[V](http://docs.google.com/java/util/SortedMap.html)> | [**subMap**](http://docs.google.com/java/util/SortedMap.html#subMap(K,%20K))([K](http://docs.google.com/java/util/SortedMap.html) fromKey, [K](http://docs.google.com/java/util/SortedMap.html) toKey)            Returns a view of the portion of this map whose keys range from fromKey, inclusive, to toKey, exclusive. |
| [SortedMap](http://docs.google.com/java/util/SortedMap.html)<[K](http://docs.google.com/java/util/SortedMap.html),[V](http://docs.google.com/java/util/SortedMap.html)> | [**tailMap**](http://docs.google.com/java/util/SortedMap.html#tailMap(K))([K](http://docs.google.com/java/util/SortedMap.html) fromKey)            Returns a view of the portion of this map whose keys are greater than or equal to fromKey. |
| [Collection](http://docs.google.com/java/util/Collection.html)<[V](http://docs.google.com/java/util/SortedMap.html)> | [**values**](http://docs.google.com/java/util/SortedMap.html#values())()            Returns a [Collection](http://docs.google.com/java/util/Collection.html) view of the values contained in this map. |

| **Methods inherited from interface java.util.**[**Map**](http://docs.google.com/java/util/Map.html) |
| --- |
| [clear](http://docs.google.com/java/util/Map.html#clear()), [containsKey](http://docs.google.com/java/util/Map.html#containsKey(java.lang.Object)), [containsValue](http://docs.google.com/java/util/Map.html#containsValue(java.lang.Object)), [equals](http://docs.google.com/java/util/Map.html#equals(java.lang.Object)), [get](http://docs.google.com/java/util/Map.html#get(java.lang.Object)), [hashCode](http://docs.google.com/java/util/Map.html#hashCode()), [isEmpty](http://docs.google.com/java/util/Map.html#isEmpty()), [put](http://docs.google.com/java/util/Map.html#put(K,%20V)), [putAll](http://docs.google.com/java/util/Map.html#putAll(java.util.Map)), [remove](http://docs.google.com/java/util/Map.html#remove(java.lang.Object)), [size](http://docs.google.com/java/util/Map.html#size()) |

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

### comparator

[Comparator](http://docs.google.com/java/util/Comparator.html)<? super [K](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.

**Returns:**the comparator used to order the keys in this map, or null if this map uses the natural ordering of its keys

### subMap

[SortedMap](http://docs.google.com/java/util/SortedMap.html)<[K](http://docs.google.com/java/util/SortedMap.html),[V](http://docs.google.com/java/util/SortedMap.html)> **subMap**([K](http://docs.google.com/java/util/SortedMap.html) fromKey,  
 [K](http://docs.google.com/java/util/SortedMap.html) toKey)

Returns a view of the portion of this map whose keys range from fromKey, inclusive, to toKey, exclusive. (If fromKey and toKey are equal, the returned map is empty.) The returned map is backed by this map, so changes in the returned map are reflected in this map, and vice-versa. The returned map supports all optional map operations that this map supports.

The returned map will throw an IllegalArgumentException on an attempt to insert a key outside its range.

**Parameters:**fromKey - low endpoint (inclusive) of the keys in the returned maptoKey - high endpoint (exclusive) of the keys in the returned map **Returns:**a view of the portion of this map whose keys range from fromKey, inclusive, to toKey, exclusive **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if fromKey and toKey cannot be compared to one another using this map's comparator (or, if the map has no comparator, using natural ordering). Implementations may, but are not required to, throw this exception if fromKey or toKey cannot be compared to keys currently in the map. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if fromKey or toKey is null and this map does not permit null keys [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if fromKey is greater than toKey; or if this map itself has a restricted range, and fromKey or toKey lies outside the bounds of the range

### headMap

[SortedMap](http://docs.google.com/java/util/SortedMap.html)<[K](http://docs.google.com/java/util/SortedMap.html),[V](http://docs.google.com/java/util/SortedMap.html)> **headMap**([K](http://docs.google.com/java/util/SortedMap.html) toKey)

Returns a view of the portion of this map whose keys are strictly less than toKey. The returned map is backed by this map, so changes in the returned map are reflected in this map, and vice-versa. The returned map supports all optional map operations that this map supports.

The returned map will throw an IllegalArgumentException on an attempt to insert a key outside its range.

**Parameters:**toKey - high endpoint (exclusive) of the keys in the returned map **Returns:**a view of the portion of this map whose keys are strictly less than toKey **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if toKey is not compatible with this map's comparator (or, if the map has no comparator, if toKey does not implement [Comparable](http://docs.google.com/java/lang/Comparable.html)). Implementations may, but are not required to, throw this exception if toKey cannot be compared to keys currently in the map. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if toKey is null and this map does not permit null keys [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if this map itself has a restricted range, and toKey lies outside the bounds of the range

### tailMap

[SortedMap](http://docs.google.com/java/util/SortedMap.html)<[K](http://docs.google.com/java/util/SortedMap.html),[V](http://docs.google.com/java/util/SortedMap.html)> **tailMap**([K](http://docs.google.com/java/util/SortedMap.html) fromKey)

Returns a view of the portion of this map whose keys are greater than or equal to fromKey. The returned map is backed by this map, so changes in the returned map are reflected in this map, and vice-versa. The returned map supports all optional map operations that this map supports.

The returned map will throw an IllegalArgumentException on an attempt to insert a key outside its range.

**Parameters:**fromKey - low endpoint (inclusive) of the keys in the returned map **Returns:**a view of the portion of this map whose keys are greater than or equal to fromKey **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if fromKey is not compatible with this map's comparator (or, if the map has no comparator, if fromKey does not implement [Comparable](http://docs.google.com/java/lang/Comparable.html)). Implementations may, but are not required to, throw this exception if fromKey cannot be compared to keys currently in the map. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if fromKey is null and this map does not permit null keys [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if this map itself has a restricted range, and fromKey lies outside the bounds of the range

### firstKey

[K](http://docs.google.com/java/util/SortedMap.html) **firstKey**()

Returns the first (lowest) key currently in this map.

**Returns:**the first (lowest) key currently in this map **Throws:** [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - if this map is empty

### lastKey

[K](http://docs.google.com/java/util/SortedMap.html) **lastKey**()

Returns the last (highest) key currently in this map.

**Returns:**the last (highest) key currently in this map **Throws:** [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - if this map is empty

### keySet

[Set](http://docs.google.com/java/util/Set.html)<[K](http://docs.google.com/java/util/SortedMap.html)> **keySet**()

Returns a [Set](http://docs.google.com/java/util/Set.html) view of the keys contained in this map. The set's iterator returns the keys in ascending order. The set is backed by the map, so changes to the map are reflected in the set, and vice-versa. If the map is modified while an iteration over the set is in progress (except through the iterator's own remove operation), the results of the iteration are undefined. The set supports element removal, which removes the corresponding mapping from the map, via the Iterator.remove, Set.remove, removeAll, retainAll, and clear operations. It does not support the add or addAll operations.

**Specified by:**[keySet](http://docs.google.com/java/util/Map.html#keySet()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/SortedMap.html),[V](http://docs.google.com/java/util/SortedMap.html)> **Returns:**a set view of the keys contained in this map, sorted in ascending order

### values

[Collection](http://docs.google.com/java/util/Collection.html)<[V](http://docs.google.com/java/util/SortedMap.html)> **values**()

Returns a [Collection](http://docs.google.com/java/util/Collection.html) view of the values contained in this map. The collection's iterator returns the values in ascending order of the corresponding keys. The collection is backed by the map, so changes to the map are reflected in the collection, and vice-versa. If the map is modified while an iteration over the collection is in progress (except through the iterator's own remove operation), the results of the iteration are undefined. The collection supports element removal, which removes the corresponding mapping from the map, via the Iterator.remove, Collection.remove, removeAll, retainAll and clear operations. It does not support the add or addAll operations.

**Specified by:**[values](http://docs.google.com/java/util/Map.html#values()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/SortedMap.html),[V](http://docs.google.com/java/util/SortedMap.html)> **Returns:**a collection view of the values contained in this map, sorted in ascending key order

### entrySet

[Set](http://docs.google.com/java/util/Set.html)<[Map.Entry](http://docs.google.com/java/util/Map.Entry.html)<[K](http://docs.google.com/java/util/SortedMap.html),[V](http://docs.google.com/java/util/SortedMap.html)>> **entrySet**()

Returns a [Set](http://docs.google.com/java/util/Set.html) view of the mappings contained in this map. The set's iterator returns the entries in ascending key order. The set is backed by the map, so changes to the map are reflected in the set, and vice-versa. If the map is modified while an iteration over the set is in progress (except through the iterator's own remove operation, or through the setValue operation on a map entry returned by the iterator) the results of the iteration are undefined. The set supports element removal, which removes the corresponding mapping from the map, via the Iterator.remove, Set.remove, removeAll, retainAll and clear operations. It does not support the add or addAll operations.

**Specified by:**[entrySet](http://docs.google.com/java/util/Map.html#entrySet()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/SortedMap.html),[V](http://docs.google.com/java/util/SortedMap.html)> **Returns:**a set view of the mappings contained in this map, sorted in ascending key order

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SortedMap.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/SimpleTimeZone.html)   [**NEXT CLASS**](http://docs.google.com/java/util/SortedSet.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/SortedMap.html)    [**NO FRAMES**](http://docs.google.com/SortedMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#1t3h5sf) |

[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).