| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ConcurrentNavigableMap.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/concurrent/ConcurrentMap.html)   [**NEXT CLASS**](http://docs.google.com/java/util/concurrent/ConcurrentSkipListMap.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/concurrent/ConcurrentNavigableMap.html)    [**NO FRAMES**](http://docs.google.com/ConcurrentNavigableMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#17dp8vu) |

## **java.util.concurrent**

Interface ConcurrentNavigableMap<K,V>

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

public interface **ConcurrentNavigableMap<K,V>**extends [ConcurrentMap](http://docs.google.com/java/util/concurrent/ConcurrentMap.html)<K,V>, [NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<K,V>

A [ConcurrentMap](http://docs.google.com/java/util/concurrent/ConcurrentMap.html) supporting [NavigableMap](http://docs.google.com/java/util/NavigableMap.html) operations, and recursively so for its navigable sub-maps.

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

**Since:** 1.6

| **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** | |
| --- | --- |
| [NavigableSet](http://docs.google.com/java/util/NavigableSet.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> | [**descendingKeySet**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html#descendingKeySet())()            Returns a reverse order [NavigableSet](http://docs.google.com/java/util/NavigableSet.html) view of the keys contained in this map. |
| [ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> | [**descendingMap**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html#descendingMap())()            Returns a reverse order view of the mappings contained in this map. |
| [ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> | [**headMap**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html#headMap(K))([K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html) toKey)            Returns a view of the portion of this map whose keys are strictly less than toKey. |
| [ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> | [**headMap**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html#headMap(K,%20boolean))([K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html) toKey, boolean inclusive)            Returns a view of the portion of this map whose keys are less than (or equal to, if inclusive is true) toKey. |
| [NavigableSet](http://docs.google.com/java/util/NavigableSet.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> | [**keySet**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html#keySet())()            Returns a [NavigableSet](http://docs.google.com/java/util/NavigableSet.html) view of the keys contained in this map. |
| [NavigableSet](http://docs.google.com/java/util/NavigableSet.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> | [**navigableKeySet**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html#navigableKeySet())()            Returns a [NavigableSet](http://docs.google.com/java/util/NavigableSet.html) view of the keys contained in this map. |
| [ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> | [**subMap**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html#subMap(K,%20boolean,%20K,%20boolean))([K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html) fromKey, boolean fromInclusive, [K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html) toKey, boolean toInclusive)            Returns a view of the portion of this map whose keys range from fromKey to toKey. |
| [ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> | [**subMap**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html#subMap(K,%20K))([K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html) fromKey, [K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html) toKey)            Returns a view of the portion of this map whose keys range from fromKey, inclusive, to toKey, exclusive. |
| [ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> | [**tailMap**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html#tailMap(K))([K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html) fromKey)            Returns a view of the portion of this map whose keys are greater than or equal to fromKey. |
| [ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> | [**tailMap**](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html#tailMap(K,%20boolean))([K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html) fromKey, boolean inclusive)            Returns a view of the portion of this map whose keys are greater than (or equal to, if inclusive is true) fromKey. |

| **Methods inherited from interface java.util.concurrent.**[**ConcurrentMap**](http://docs.google.com/java/util/concurrent/ConcurrentMap.html) |
| --- |
| [putIfAbsent](http://docs.google.com/java/util/concurrent/ConcurrentMap.html#putIfAbsent(K,%20V)), [remove](http://docs.google.com/java/util/concurrent/ConcurrentMap.html#remove(java.lang.Object,%20java.lang.Object)), [replace](http://docs.google.com/java/util/concurrent/ConcurrentMap.html#replace(K,%20V)), [replace](http://docs.google.com/java/util/concurrent/ConcurrentMap.html#replace(K,%20V,%20V)) |

| **Methods inherited from interface java.util.**[**NavigableMap**](http://docs.google.com/java/util/NavigableMap.html) |
| --- |
| [ceilingEntry](http://docs.google.com/java/util/NavigableMap.html#ceilingEntry(K)), [ceilingKey](http://docs.google.com/java/util/NavigableMap.html#ceilingKey(K)), [firstEntry](http://docs.google.com/java/util/NavigableMap.html#firstEntry()), [floorEntry](http://docs.google.com/java/util/NavigableMap.html#floorEntry(K)), [floorKey](http://docs.google.com/java/util/NavigableMap.html#floorKey(K)), [higherEntry](http://docs.google.com/java/util/NavigableMap.html#higherEntry(K)), [higherKey](http://docs.google.com/java/util/NavigableMap.html#higherKey(K)), [lastEntry](http://docs.google.com/java/util/NavigableMap.html#lastEntry()), [lowerEntry](http://docs.google.com/java/util/NavigableMap.html#lowerEntry(K)), [lowerKey](http://docs.google.com/java/util/NavigableMap.html#lowerKey(K)), [pollFirstEntry](http://docs.google.com/java/util/NavigableMap.html#pollFirstEntry()), [pollLastEntry](http://docs.google.com/java/util/NavigableMap.html#pollLastEntry()) |

| **Methods inherited from interface java.util.**[**SortedMap**](http://docs.google.com/java/util/SortedMap.html) |
| --- |
| [comparator](http://docs.google.com/java/util/SortedMap.html#comparator()), [entrySet](http://docs.google.com/java/util/SortedMap.html#entrySet()), [firstKey](http://docs.google.com/java/util/SortedMap.html#firstKey()), [lastKey](http://docs.google.com/java/util/SortedMap.html#lastKey()), [values](http://docs.google.com/java/util/SortedMap.html#values()) |

| **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** |
| --- |

### subMap

[ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **subMap**([K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html) fromKey,  
 boolean fromInclusive,  
 [K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html) toKey,  
 boolean toInclusive)

**Description copied from interface:** [**NavigableMap**](http://docs.google.com/java/util/NavigableMap.html#subMap(K,%20boolean,%20K,%20boolean)) Returns a view of the portion of this map whose keys range from fromKey to toKey. If fromKey and toKey are equal, the returned map is empty unless fromExclusive and toExclusive are both true. 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 of its range, or to construct a submap either of whose endpoints lie outside its range.

**Specified by:**[subMap](http://docs.google.com/java/util/NavigableMap.html#subMap(K,%20boolean,%20K,%20boolean)) in interface [NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **Parameters:**fromKey - low endpoint of the keys in the returned mapfromInclusive - true if the low endpoint is to be included in the returned viewtoKey - high endpoint of the keys in the returned maptoInclusive - true if the high endpoint is to be included in the returned view **Returns:**a view of the portion of this map whose keys range from fromKey to toKey **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

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

**Description copied from interface:** [**NavigableMap**](http://docs.google.com/java/util/NavigableMap.html#headMap(K,%20boolean)) Returns a view of the portion of this map whose keys are less than (or equal to, if inclusive is true) 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.

**Specified by:**[headMap](http://docs.google.com/java/util/NavigableMap.html#headMap(K,%20boolean)) in interface [NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **Parameters:**toKey - high endpoint of the keys in the returned mapinclusive - true if the high endpoint is to be included in the returned view **Returns:**a view of the portion of this map whose keys are less than (or equal to, if inclusive is true) 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

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

**Description copied from interface:** [**NavigableMap**](http://docs.google.com/java/util/NavigableMap.html#tailMap(K,%20boolean)) Returns a view of the portion of this map whose keys are greater than (or equal to, if inclusive is true) 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.

**Specified by:**[tailMap](http://docs.google.com/java/util/NavigableMap.html#tailMap(K,%20boolean)) in interface [NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **Parameters:**fromKey - low endpoint of the keys in the returned mapinclusive - true if the low endpoint is to be included in the returned view **Returns:**a view of the portion of this map whose keys are greater than (or equal to, if inclusive is true) 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

### subMap

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

**Description copied from interface:** [**NavigableMap**](http://docs.google.com/java/util/NavigableMap.html#subMap(K,%20K)) 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.

Equivalent to subMap(fromKey, true, toKey, false).

**Specified by:**[subMap](http://docs.google.com/java/util/NavigableMap.html#subMap(K,%20K)) in interface [NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)>**Specified by:**[subMap](http://docs.google.com/java/util/SortedMap.html#subMap(K,%20K)) in interface [SortedMap](http://docs.google.com/java/util/SortedMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **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

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

**Description copied from interface:** [**NavigableMap**](http://docs.google.com/java/util/NavigableMap.html#headMap(K)) 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.

Equivalent to headMap(toKey, false).

**Specified by:**[headMap](http://docs.google.com/java/util/NavigableMap.html#headMap(K)) in interface [NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)>**Specified by:**[headMap](http://docs.google.com/java/util/SortedMap.html#headMap(K)) in interface [SortedMap](http://docs.google.com/java/util/SortedMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **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

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

**Description copied from interface:** [**NavigableMap**](http://docs.google.com/java/util/NavigableMap.html#tailMap(K)) 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.

Equivalent to tailMap(fromKey, true).

**Specified by:**[tailMap](http://docs.google.com/java/util/NavigableMap.html#tailMap(K)) in interface [NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)>**Specified by:**[tailMap](http://docs.google.com/java/util/SortedMap.html#tailMap(K)) in interface [SortedMap](http://docs.google.com/java/util/SortedMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **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

### descendingMap

[ConcurrentNavigableMap](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **descendingMap**()

Returns a reverse order view of the mappings contained in this map. The descending map is backed by this map, so changes to the map are reflected in the descending map, and vice-versa.

The returned map has an ordering equivalent to [Collections.reverseOrder](http://docs.google.com/java/util/Collections.html#reverseOrder(java.util.Comparator))(comparator()). The expression m.descendingMap().descendingMap() returns a view of m essentially equivalent to m.

**Specified by:**[descendingMap](http://docs.google.com/java/util/NavigableMap.html#descendingMap()) in interface [NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **Returns:**a reverse order view of this map

### navigableKeySet

[NavigableSet](http://docs.google.com/java/util/NavigableSet.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **navigableKeySet**()

Returns a [NavigableSet](http://docs.google.com/java/util/NavigableSet.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. 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.

The view's iterator is a "weakly consistent" iterator that will never throw [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html), and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

**Specified by:**[navigableKeySet](http://docs.google.com/java/util/NavigableMap.html#navigableKeySet()) in interface [NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **Returns:**a navigable set view of the keys in this map

### keySet

[NavigableSet](http://docs.google.com/java/util/NavigableSet.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **keySet**()

Returns a [NavigableSet](http://docs.google.com/java/util/NavigableSet.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. 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.

The view's iterator is a "weakly consistent" iterator that will never throw [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html), and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

This method is equivalent to method navigableKeySet.

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

### descendingKeySet

[NavigableSet](http://docs.google.com/java/util/NavigableSet.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **descendingKeySet**()

Returns a reverse order [NavigableSet](http://docs.google.com/java/util/NavigableSet.html) view of the keys contained in this map. The set's iterator returns the keys in descending order. The set is backed by the map, so changes to the map are reflected in the set, and vice-versa. 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.

The view's iterator is a "weakly consistent" iterator that will never throw [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html), and guarantees to traverse elements as they existed upon construction of the iterator, and may (but is not guaranteed to) reflect any modifications subsequent to construction.

**Specified by:**[descendingKeySet](http://docs.google.com/java/util/NavigableMap.html#descendingKeySet()) in interface [NavigableMap](http://docs.google.com/java/util/NavigableMap.html)<[K](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html),[V](http://docs.google.com/java/util/concurrent/ConcurrentNavigableMap.html)> **Returns:**a reverse order navigable set view of the keys in this map

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