| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/WeakHashMap.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/Vector.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?java/util/WeakHashMap.html)    [**NO FRAMES**](http://docs.google.com/WeakHashMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#1ksv4uv) |

## **java.util**

Class WeakHashMap<K,V>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.util.AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<K,V>  
 **java.util.WeakHashMap<K,V>**

**Type Parameters:**K - the type of keys maintained by this mapV - the type of mapped values **All Implemented Interfaces:** [Map](http://docs.google.com/java/util/Map.html)<K,V>

public class **WeakHashMap<K,V>**extends [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<K,V>implements [Map](http://docs.google.com/java/util/Map.html)<K,V>

A hashtable-based Map implementation with *weak keys*. An entry in a WeakHashMap will automatically be removed when its key is no longer in ordinary use. More precisely, the presence of a mapping for a given key will not prevent the key from being discarded by the garbage collector, that is, made finalizable, finalized, and then reclaimed. When a key has been discarded its entry is effectively removed from the map, so this class behaves somewhat differently from other Map implementations.

Both null values and the null key are supported. This class has performance characteristics similar to those of the HashMap class, and has the same efficiency parameters of *initial capacity* and *load factor*.

Like most collection classes, this class is not synchronized. A synchronized WeakHashMap may be constructed using the [Collections.synchronizedMap](http://docs.google.com/java/util/Collections.html#synchronizedMap(java.util.Map)) method.

This class is intended primarily for use with key objects whose equals methods test for object identity using the == operator. Once such a key is discarded it can never be recreated, so it is impossible to do a lookup of that key in a WeakHashMap at some later time and be surprised that its entry has been removed. This class will work perfectly well with key objects whose equals methods are not based upon object identity, such as String instances. With such recreatable key objects, however, the automatic removal of WeakHashMap entries whose keys have been discarded may prove to be confusing.

The behavior of the WeakHashMap class depends in part upon the actions of the garbage collector, so several familiar (though not required) Map invariants do not hold for this class. Because the garbage collector may discard keys at any time, a WeakHashMap may behave as though an unknown thread is silently removing entries. In particular, even if you synchronize on a WeakHashMap instance and invoke none of its mutator methods, it is possible for the size method to return smaller values over time, for the isEmpty method to return false and then true, for the containsKey method to return true and later false for a given key, for the get method to return a value for a given key but later return null, for the put method to return null and the remove method to return false for a key that previously appeared to be in the map, and for successive examinations of the key set, the value collection, and the entry set to yield successively smaller numbers of elements.

Each key object in a WeakHashMap is stored indirectly as the referent of a weak reference. Therefore a key will automatically be removed only after the weak references to it, both inside and outside of the map, have been cleared by the garbage collector.

**Implementation note:** The value objects in a WeakHashMap are held by ordinary strong references. Thus care should be taken to ensure that value objects do not strongly refer to their own keys, either directly or indirectly, since that will prevent the keys from being discarded. Note that a value object may refer indirectly to its key via the WeakHashMap itself; that is, a value object may strongly refer to some other key object whose associated value object, in turn, strongly refers to the key of the first value object. One way to deal with this is to wrap values themselves within WeakReferences before inserting, as in: m.put(key, new WeakReference(value)), and then unwrapping upon each get.

The iterators returned by the iterator method of the collections returned by all of this class's "collection view methods" are *fail-fast*: if the map is structurally modified at any time after the iterator is created, in any way except through the iterator's own remove method, the iterator will throw a [ConcurrentModificationException](http://docs.google.com/java/util/ConcurrentModificationException.html). Thus, in the face of concurrent modification, the iterator fails quickly and cleanly, rather than risking arbitrary, non-deterministic behavior at an undetermined time in the future.

Note that the fail-fast behavior of an iterator cannot be guaranteed as it is, generally speaking, impossible to make any hard guarantees in the presence of unsynchronized concurrent modification. Fail-fast iterators throw ConcurrentModificationException on a best-effort basis. Therefore, it would be wrong to write a program that depended on this exception for its correctness: *the fail-fast behavior of iterators should be used only to detect bugs.*

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

**Since:** 1.2 **See Also:**[HashMap](http://docs.google.com/java/util/HashMap.html), [WeakReference](http://docs.google.com/java/lang/ref/WeakReference.html)

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

| **Nested classes/interfaces inherited from class java.util.**[**AbstractMap**](http://docs.google.com/java/util/AbstractMap.html) |
| --- |
| [AbstractMap.SimpleEntry](http://docs.google.com/java/util/AbstractMap.SimpleEntry.html)<[K](http://docs.google.com/java/util/AbstractMap.SimpleEntry.html),[V](http://docs.google.com/java/util/AbstractMap.SimpleEntry.html)>, [AbstractMap.SimpleImmutableEntry](http://docs.google.com/java/util/AbstractMap.SimpleImmutableEntry.html)<[K](http://docs.google.com/java/util/AbstractMap.SimpleImmutableEntry.html),[V](http://docs.google.com/java/util/AbstractMap.SimpleImmutableEntry.html)> |

| **Constructor Summary** | |
| --- | --- |
| [**WeakHashMap**](http://docs.google.com/java/util/WeakHashMap.html#WeakHashMap())()            Constructs a new, empty WeakHashMap with the default initial capacity (16) and load factor (0.75). |
| [**WeakHashMap**](http://docs.google.com/java/util/WeakHashMap.html#WeakHashMap(int))(int initialCapacity)            Constructs a new, empty WeakHashMap with the given initial capacity and the default load factor (0.75). |
| [**WeakHashMap**](http://docs.google.com/java/util/WeakHashMap.html#WeakHashMap(int,%20float))(int initialCapacity, float loadFactor)            Constructs a new, empty WeakHashMap with the given initial capacity and the given load factor. |
| [**WeakHashMap**](http://docs.google.com/java/util/WeakHashMap.html#WeakHashMap(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<? extends [K](http://docs.google.com/java/util/WeakHashMap.html),? extends [V](http://docs.google.com/java/util/WeakHashMap.html)> m)            Constructs a new WeakHashMap with the same mappings as the specified map. |

| **Method Summary** | |
| --- | --- |
| void | [**clear**](http://docs.google.com/java/util/WeakHashMap.html#clear())()            Removes all of the mappings from this map. |
| boolean | [**containsKey**](http://docs.google.com/java/util/WeakHashMap.html#containsKey(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Returns true if this map contains a mapping for the specified key. |
| boolean | [**containsValue**](http://docs.google.com/java/util/WeakHashMap.html#containsValue(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) value)            Returns true if this map maps one or more keys to the specified value. |
| [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/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>> | [**entrySet**](http://docs.google.com/java/util/WeakHashMap.html#entrySet())()            Returns a [Set](http://docs.google.com/java/util/Set.html) view of the mappings contained in this map. |
| [V](http://docs.google.com/java/util/WeakHashMap.html) | [**get**](http://docs.google.com/java/util/WeakHashMap.html#get(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key. |
| boolean | [**isEmpty**](http://docs.google.com/java/util/WeakHashMap.html#isEmpty())()            Returns true if this map contains no key-value mappings. |
| [Set](http://docs.google.com/java/util/Set.html)<[K](http://docs.google.com/java/util/WeakHashMap.html)> | [**keySet**](http://docs.google.com/java/util/WeakHashMap.html#keySet())()            Returns a [Set](http://docs.google.com/java/util/Set.html) view of the keys contained in this map. |
| [V](http://docs.google.com/java/util/WeakHashMap.html) | [**put**](http://docs.google.com/java/util/WeakHashMap.html#put(K,%20V))([K](http://docs.google.com/java/util/WeakHashMap.html) key, [V](http://docs.google.com/java/util/WeakHashMap.html) value)            Associates the specified value with the specified key in this map. |
| void | [**putAll**](http://docs.google.com/java/util/WeakHashMap.html#putAll(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<? extends [K](http://docs.google.com/java/util/WeakHashMap.html),? extends [V](http://docs.google.com/java/util/WeakHashMap.html)> m)            Copies all of the mappings from the specified map to this map. |
| [V](http://docs.google.com/java/util/WeakHashMap.html) | [**remove**](http://docs.google.com/java/util/WeakHashMap.html#remove(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Removes the mapping for a key from this weak hash map if it is present. |
| int | [**size**](http://docs.google.com/java/util/WeakHashMap.html#size())()            Returns the number of key-value mappings in this map. |
| [Collection](http://docs.google.com/java/util/Collection.html)<[V](http://docs.google.com/java/util/WeakHashMap.html)> | [**values**](http://docs.google.com/java/util/WeakHashMap.html#values())()            Returns a [Collection](http://docs.google.com/java/util/Collection.html) view of the values contained in this map. |

| **Methods inherited from class java.util.**[**AbstractMap**](http://docs.google.com/java/util/AbstractMap.html) |
| --- |
| [clone](http://docs.google.com/java/util/AbstractMap.html#clone()), [equals](http://docs.google.com/java/util/AbstractMap.html#equals(java.lang.Object)), [hashCode](http://docs.google.com/java/util/AbstractMap.html#hashCode()), [toString](http://docs.google.com/java/util/AbstractMap.html#toString()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Methods inherited from interface java.util.**[**Map**](http://docs.google.com/java/util/Map.html) |
| --- |
| [equals](http://docs.google.com/java/util/Map.html#equals(java.lang.Object)), [hashCode](http://docs.google.com/java/util/Map.html#hashCode()) |

| **Constructor Detail** |
| --- |

### WeakHashMap

public **WeakHashMap**(int initialCapacity,  
 float loadFactor)

Constructs a new, empty WeakHashMap with the given initial capacity and the given load factor.

**Parameters:**initialCapacity - The initial capacity of the WeakHashMaploadFactor - The load factor of the WeakHashMap **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the initial capacity is negative, or if the load factor is nonpositive.

### WeakHashMap

public **WeakHashMap**(int initialCapacity)

Constructs a new, empty WeakHashMap with the given initial capacity and the default load factor (0.75).

**Parameters:**initialCapacity - The initial capacity of the WeakHashMap **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the initial capacity is negative

### WeakHashMap

public **WeakHashMap**()

Constructs a new, empty WeakHashMap with the default initial capacity (16) and load factor (0.75).

### WeakHashMap

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

Constructs a new WeakHashMap with the same mappings as the specified map. The WeakHashMap is created with the default load factor (0.75) and an initial capacity sufficient to hold the mappings in the specified map.

**Parameters:**m - the map whose mappings are to be placed in this map **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified map is null**Since:** 1.3

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

### size

public int **size**()

Returns the number of key-value mappings in this map. This result is a snapshot, and may not reflect unprocessed entries that will be removed before next attempted access because they are no longer referenced.

**Specified by:**[size](http://docs.google.com/java/util/Map.html#size()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Overrides:**[size](http://docs.google.com/java/util/AbstractMap.html#size()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)> **Returns:**the number of key-value mappings in this map

### isEmpty

public boolean **isEmpty**()

Returns true if this map contains no key-value mappings. This result is a snapshot, and may not reflect unprocessed entries that will be removed before next attempted access because they are no longer referenced.

**Specified by:**[isEmpty](http://docs.google.com/java/util/Map.html#isEmpty()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Overrides:**[isEmpty](http://docs.google.com/java/util/AbstractMap.html#isEmpty()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)> **Returns:**true if this map contains no key-value mappings

### get

public [V](http://docs.google.com/java/util/WeakHashMap.html) **get**([Object](http://docs.google.com/java/lang/Object.html) key)

Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key.

More formally, if this map contains a mapping from a key k to a value v such that (key==null ? k==null : key.equals(k)), then this method returns v; otherwise it returns null. (There can be at most one such mapping.)

A return value of null does not *necessarily* indicate that the map contains no mapping for the key; it's also possible that the map explicitly maps the key to null. The [containsKey](http://docs.google.com/java/util/WeakHashMap.html#containsKey(java.lang.Object)) operation may be used to distinguish these two cases.

**Specified by:**[get](http://docs.google.com/java/util/Map.html#get(java.lang.Object)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Overrides:**[get](http://docs.google.com/java/util/AbstractMap.html#get(java.lang.Object)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)> **Parameters:**key - the key whose associated value is to be returned **Returns:**the value to which the specified key is mapped, or null if this map contains no mapping for the key**See Also:**[put(Object, Object)](http://docs.google.com/java/util/WeakHashMap.html#put(K,%20V))

### containsKey

public boolean **containsKey**([Object](http://docs.google.com/java/lang/Object.html) key)

Returns true if this map contains a mapping for the specified key.

**Specified by:**[containsKey](http://docs.google.com/java/util/Map.html#containsKey(java.lang.Object)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Overrides:**[containsKey](http://docs.google.com/java/util/AbstractMap.html#containsKey(java.lang.Object)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)> **Parameters:**key - The key whose presence in this map is to be tested **Returns:**true if there is a mapping for key; false otherwise

### put

public [V](http://docs.google.com/java/util/WeakHashMap.html) **put**([K](http://docs.google.com/java/util/WeakHashMap.html) key,  
 [V](http://docs.google.com/java/util/WeakHashMap.html) value)

Associates the specified value with the specified key in this map. If the map previously contained a mapping for this key, the old value is replaced.

**Specified by:**[put](http://docs.google.com/java/util/Map.html#put(K,%20V)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Overrides:**[put](http://docs.google.com/java/util/AbstractMap.html#put(K,%20V)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)> **Parameters:**key - key with which the specified value is to be associated.value - value to be associated with the specified key. **Returns:**the previous value associated with key, or null if there was no mapping for key. (A null return can also indicate that the map previously associated null with key.)

### putAll

public void **putAll**([Map](http://docs.google.com/java/util/Map.html)<? extends [K](http://docs.google.com/java/util/WeakHashMap.html),? extends [V](http://docs.google.com/java/util/WeakHashMap.html)> m)

Copies all of the mappings from the specified map to this map. These mappings will replace any mappings that this map had for any of the keys currently in the specified map.

**Specified by:**[putAll](http://docs.google.com/java/util/Map.html#putAll(java.util.Map)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Overrides:**[putAll](http://docs.google.com/java/util/AbstractMap.html#putAll(java.util.Map)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)> **Parameters:**m - mappings to be stored in this map. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified map is null.

### remove

public [V](http://docs.google.com/java/util/WeakHashMap.html) **remove**([Object](http://docs.google.com/java/lang/Object.html) key)

Removes the mapping for a key from this weak hash map if it is present. More formally, if this map contains a mapping from key k to value v such that (key==null ? k==null : key.equals(k)), that mapping is removed. (The map can contain at most one such mapping.)

Returns the value to which this map previously associated the key, or null if the map contained no mapping for the key. A return value of null does not *necessarily* indicate that the map contained no mapping for the key; it's also possible that the map explicitly mapped the key to null.

The map will not contain a mapping for the specified key once the call returns.

**Specified by:**[remove](http://docs.google.com/java/util/Map.html#remove(java.lang.Object)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Overrides:**[remove](http://docs.google.com/java/util/AbstractMap.html#remove(java.lang.Object)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)> **Parameters:**key - key whose mapping is to be removed from the map **Returns:**the previous value associated with key, or null if there was no mapping for key

### clear

public void **clear**()

Removes all of the mappings from this map. The map will be empty after this call returns.

**Specified by:**[clear](http://docs.google.com/java/util/Map.html#clear()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Overrides:**[clear](http://docs.google.com/java/util/AbstractMap.html#clear()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>

### containsValue

public boolean **containsValue**([Object](http://docs.google.com/java/lang/Object.html) value)

Returns true if this map maps one or more keys to the specified value.

**Specified by:**[containsValue](http://docs.google.com/java/util/Map.html#containsValue(java.lang.Object)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Overrides:**[containsValue](http://docs.google.com/java/util/AbstractMap.html#containsValue(java.lang.Object)) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)> **Parameters:**value - value whose presence in this map is to be tested **Returns:**true if this map maps one or more keys to the specified value

### keySet

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

Returns a [Set](http://docs.google.com/java/util/Set.html) view of the keys contained in this map. 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/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Overrides:**[keySet](http://docs.google.com/java/util/AbstractMap.html#keySet()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)> **Returns:**a set view of the keys contained in this map

### values

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

Returns a [Collection](http://docs.google.com/java/util/Collection.html) view of the values contained in this map. 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/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Overrides:**[values](http://docs.google.com/java/util/AbstractMap.html#values()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)> **Returns:**a collection view of the values contained in this map

### entrySet

public [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/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>> **entrySet**()

Returns a [Set](http://docs.google.com/java/util/Set.html) view of the mappings contained in this map. 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/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)>**Specified by:**[entrySet](http://docs.google.com/java/util/AbstractMap.html#entrySet()) in class [AbstractMap](http://docs.google.com/java/util/AbstractMap.html)<[K](http://docs.google.com/java/util/WeakHashMap.html),[V](http://docs.google.com/java/util/WeakHashMap.html)> **Returns:**a set view of the mappings contained 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/WeakHashMap.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/Vector.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?java/util/WeakHashMap.html)    [**NO FRAMES**](http://docs.google.com/WeakHashMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#1ksv4uv) |

[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).