| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AbstractMap.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/AbstractList.html)   [**NEXT CLASS**](http://docs.google.com/java/util/AbstractMap.SimpleEntry.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/AbstractMap.html)    [**NO FRAMES**](http://docs.google.com/AbstractMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

## **java.util**

Class AbstractMap<K,V>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.util.AbstractMap<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> **Direct Known Subclasses:** [ConcurrentHashMap](http://docs.google.com/java/util/concurrent/ConcurrentHashMap.html), [ConcurrentSkipListMap](http://docs.google.com/java/util/concurrent/ConcurrentSkipListMap.html), [EnumMap](http://docs.google.com/java/util/EnumMap.html), [HashMap](http://docs.google.com/java/util/HashMap.html), [IdentityHashMap](http://docs.google.com/java/util/IdentityHashMap.html), [TreeMap](http://docs.google.com/java/util/TreeMap.html), [WeakHashMap](http://docs.google.com/java/util/WeakHashMap.html)

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

This class provides a skeletal implementation of the Map interface, to minimize the effort required to implement this interface.

To implement an unmodifiable map, the programmer needs only to extend this class and provide an implementation for the entrySet method, which returns a set-view of the map's mappings. Typically, the returned set will, in turn, be implemented atop AbstractSet. This set should not support the add or remove methods, and its iterator should not support the remove method.

To implement a modifiable map, the programmer must additionally override this class's put method (which otherwise throws an UnsupportedOperationException), and the iterator returned by entrySet().iterator() must additionally implement its remove method.

The programmer should generally provide a void (no argument) and map constructor, as per the recommendation in the Map interface specification.

The documentation for each non-abstract method in this class describes its implementation in detail. Each of these methods may be overridden if the map being implemented admits a more efficient implementation.

This class 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), [Collection](http://docs.google.com/java/util/Collection.html)

| **Nested Class Summary** | |
| --- | --- |
| static class | [**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)**>**            An Entry maintaining a key and a value. |
| static class | [**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)**>**            An Entry maintaining an immutable key and value. |

| **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)> |

| **Constructor Summary** | |
| --- | --- |
| protected | [**AbstractMap**](http://docs.google.com/java/util/AbstractMap.html#AbstractMap())()            Sole constructor. |

| **Method Summary** | |
| --- | --- |
| void | [**clear**](http://docs.google.com/java/util/AbstractMap.html#clear())()            Removes all of the mappings from this map (optional operation). |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/util/AbstractMap.html#clone())()            Returns a shallow copy of this AbstractMap instance: the keys and values themselves are not cloned. |
| boolean | [**containsKey**](http://docs.google.com/java/util/AbstractMap.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/AbstractMap.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. |
| abstract  [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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.html)>> | [**entrySet**](http://docs.google.com/java/util/AbstractMap.html#entrySet())()            Returns a [Set](http://docs.google.com/java/util/Set.html) view of the mappings contained in this map. |
| boolean | [**equals**](http://docs.google.com/java/util/AbstractMap.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) o)            Compares the specified object with this map for equality. |
| [V](http://docs.google.com/java/util/AbstractMap.html) | [**get**](http://docs.google.com/java/util/AbstractMap.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. |
| int | [**hashCode**](http://docs.google.com/java/util/AbstractMap.html#hashCode())()            Returns the hash code value for this map. |
| boolean | [**isEmpty**](http://docs.google.com/java/util/AbstractMap.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/AbstractMap.html)> | [**keySet**](http://docs.google.com/java/util/AbstractMap.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/AbstractMap.html) | [**put**](http://docs.google.com/java/util/AbstractMap.html#put(K,%20V))([K](http://docs.google.com/java/util/AbstractMap.html) key, [V](http://docs.google.com/java/util/AbstractMap.html) value)            Associates the specified value with the specified key in this map (optional operation). |
| void | [**putAll**](http://docs.google.com/java/util/AbstractMap.html#putAll(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<? extends [K](http://docs.google.com/java/util/AbstractMap.html),? extends [V](http://docs.google.com/java/util/AbstractMap.html)> m)            Copies all of the mappings from the specified map to this map (optional operation). |
| [V](http://docs.google.com/java/util/AbstractMap.html) | [**remove**](http://docs.google.com/java/util/AbstractMap.html#remove(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) key)            Removes the mapping for a key from this map if it is present (optional operation). |
| int | [**size**](http://docs.google.com/java/util/AbstractMap.html#size())()            Returns the number of key-value mappings in this map. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/util/AbstractMap.html#toString())()            Returns a string representation of this map. |
| [Collection](http://docs.google.com/java/util/Collection.html)<[V](http://docs.google.com/java/util/AbstractMap.html)> | [**values**](http://docs.google.com/java/util/AbstractMap.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.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)) |

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

### AbstractMap

protected **AbstractMap**()

Sole constructor. (For invocation by subclass constructors, typically implicit.)

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

### size

public int **size**()

Returns the number of key-value mappings in this map. If the map contains more than Integer.MAX\_VALUE elements, returns Integer.MAX\_VALUE.

This implementation returns entrySet().size().

**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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.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 implementation returns size() == 0.

**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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.html)> **Returns:**true if this map contains no key-value mappings

### 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. More formally, returns true if and only if this map contains at least one mapping to a value v such that (value==null ? v==null : value.equals(v)). This operation will probably require time linear in the map size for most implementations of the Map interface.

This implementation iterates over entrySet() searching for an entry with the specified value. If such an entry is found, true is returned. If the iteration terminates without finding such an entry, false is returned. Note that this implementation requires linear time in the size of the map.

**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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.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 **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the value is of an inappropriate type for this map (optional) [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified value is null and this map does not permit null values (optional)

### 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. More formally, returns true if and only if this map contains a mapping for a key k such that (key==null ? k==null : key.equals(k)). (There can be at most one such mapping.)

This implementation iterates over entrySet() searching for an entry with the specified key. If such an entry is found, true is returned. If the iteration terminates without finding such an entry, false is returned. Note that this implementation requires linear time in the size of the map; many implementations will override this method.

**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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.html)> **Parameters:**key - key whose presence in this map is to be tested **Returns:**true if this map contains a mapping for the specified key **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the key is of an inappropriate type for this map (optional) [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key is null and this map does not permit null keys (optional)

### get

public [V](http://docs.google.com/java/util/AbstractMap.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.)

If this map permits null values, then 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/Map.html#containsKey(java.lang.Object)) operation may be used to distinguish these two cases.

This implementation iterates over entrySet() searching for an entry with the specified key. If such an entry is found, the entry's value is returned. If the iteration terminates without finding such an entry, null is returned. Note that this implementation requires linear time in the size of the map; many implementations will override this method.

**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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.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 **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the key is of an inappropriate type for this map (optional) [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key is null and this map does not permit null keys (optional)

### put

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

Associates the specified value with the specified key in this map (optional operation). If the map previously contained a mapping for the key, the old value is replaced by the specified value. (A map m is said to contain a mapping for a key k if and only if [m.containsKey(k)](http://docs.google.com/java/util/Map.html#containsKey(java.lang.Object)) would return true.)

This implementation always throws an UnsupportedOperationException.

**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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.html)> **Parameters:**key - key with which the specified value is to be associatedvalue - 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, if the implementation supports null values.) **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the put operation is not supported by this map [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the class of the specified key or value prevents it from being stored in this map [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key or value is null and this map does not permit null keys or values [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if some property of the specified key or value prevents it from being stored in this map

### remove

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

Removes the mapping for a key from this map if it is present (optional operation). 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.

If this map permits null values, then 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.

This implementation iterates over entrySet() searching for an entry with the specified key. If such an entry is found, its value is obtained with its getValue operation, the entry is removed from the collection (and the backing map) with the iterator's remove operation, and the saved value is returned. If the iteration terminates without finding such an entry, null is returned. Note that this implementation requires linear time in the size of the map; many implementations will override this method.

Note that this implementation throws an UnsupportedOperationException if the entrySet iterator does not support the remove method and this map contains a mapping for the specified key.

**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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.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. **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the remove operation is not supported by this map [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the key is of an inappropriate type for this map (optional) [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified key is null and this map does not permit null keys (optional)

### putAll

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

Copies all of the mappings from the specified map to this map (optional operation). The effect of this call is equivalent to that of calling [put(k, v)](http://docs.google.com/java/util/Map.html#put(K,%20V)) on this map once for each mapping from key k to value v in the specified map. The behavior of this operation is undefined if the specified map is modified while the operation is in progress.

This implementation iterates over the specified map's entrySet() collection, and calls this map's put operation once for each entry returned by the iteration.

Note that this implementation throws an UnsupportedOperationException if this map does not support the put operation and the specified map is nonempty.

**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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.html)> **Parameters:**m - mappings to be stored in this map **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the putAll operation is not supported by this map [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the class of a key or value in the specified map prevents it from being stored in this map [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the specified map is null, or if this map does not permit null keys or values, and the specified map contains null keys or values [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if some property of a key or value in the specified map prevents it from being stored in this map

### clear

public void **clear**()

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

This implementation calls entrySet().clear().

Note that this implementation throws an UnsupportedOperationException if the entrySet does not support the clear operation.

**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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.html)> **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if the clear operation is not supported by this map

### keySet

public [Set](http://docs.google.com/java/util/Set.html)<[K](http://docs.google.com/java/util/AbstractMap.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.

This implementation returns a set that subclasses [AbstractSet](http://docs.google.com/java/util/AbstractSet.html). The subclass's iterator method returns a "wrapper object" over this map's entrySet() iterator. The size method delegates to this map's size method and the contains method delegates to this map's containsKey method.

The set is created the first time this method is called, and returned in response to all subsequent calls. No synchronization is performed, so there is a slight chance that multiple calls to this method will not all return the same set.

**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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.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/AbstractMap.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.

This implementation returns a collection that subclasses [AbstractCollection](http://docs.google.com/java/util/AbstractCollection.html). The subclass's iterator method returns a "wrapper object" over this map's entrySet() iterator. The size method delegates to this map's size method and the contains method delegates to this map's containsValue method.

The collection is created the first time this method is called, and returned in response to all subsequent calls. No synchronization is performed, so there is a slight chance that multiple calls to this method will not all return the same collection.

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

### entrySet

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

**Description copied from interface:** [**Map**](http://docs.google.com/java/util/Map.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/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.html)> **Returns:**a set view of the mappings contained in this map

### equals

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

Compares the specified object with this map for equality. Returns true if the given object is also a map and the two maps represent the same mappings. More formally, two maps m1 and m2 represent the same mappings if m1.entrySet().equals(m2.entrySet()). This ensures that the equals method works properly across different implementations of the Map interface.

This implementation first checks if the specified object is this map; if so it returns true. Then, it checks if the specified object is a map whose size is identical to the size of this map; if not, it returns false. If so, it iterates over this map's entrySet collection, and checks that the specified map contains each mapping that this map contains. If the specified map fails to contain such a mapping, false is returned. If the iteration completes, true is returned.

**Specified by:**[equals](http://docs.google.com/java/util/Map.html#equals(java.lang.Object)) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.html)>**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**o - object to be compared for equality with this map **Returns:**true if the specified object is equal to this map**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Returns the hash code value for this map. The hash code of a map is defined to be the sum of the hash codes of each entry in the map's entrySet() view. This ensures that m1.equals(m2) implies that m1.hashCode()==m2.hashCode() for any two maps m1 and m2, as required by the general contract of [Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()).

This implementation iterates over entrySet(), calling [hashCode()](http://docs.google.com/java/util/Map.Entry.html#hashCode()) on each element (entry) in the set, and adding up the results.

**Specified by:**[hashCode](http://docs.google.com/java/util/Map.html#hashCode()) in interface [Map](http://docs.google.com/java/util/Map.html)<[K](http://docs.google.com/java/util/AbstractMap.html),[V](http://docs.google.com/java/util/AbstractMap.html)>**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**the hash code value for this map**See Also:**[Map.Entry.hashCode()](http://docs.google.com/java/util/Map.Entry.html#hashCode()), [Object.equals(Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Set.equals(Object)](http://docs.google.com/java/util/Set.html#equals(java.lang.Object))

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**()

Returns a string representation of this map. The string representation consists of a list of key-value mappings in the order returned by the map's entrySet view's iterator, enclosed in braces ("{}"). Adjacent mappings are separated by the characters ", " (comma and space). Each key-value mapping is rendered as the key followed by an equals sign ("=") followed by the associated value. Keys and values are converted to strings as by [String.valueOf(Object)](http://docs.google.com/java/lang/String.html#valueOf(java.lang.Object)).

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of this map

### clone

protected [Object](http://docs.google.com/java/lang/Object.html) **clone**()  
 throws [CloneNotSupportedException](http://docs.google.com/java/lang/CloneNotSupportedException.html)

Returns a shallow copy of this AbstractMap instance: the keys and values themselves are not cloned.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a shallow copy of this map **Throws:** [CloneNotSupportedException](http://docs.google.com/java/lang/CloneNotSupportedException.html) - if the object's class does not support the Cloneable interface. Subclasses that override the clone method can also throw this exception to indicate that an instance cannot be cloned.**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/AbstractMap.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/AbstractList.html)   [**NEXT CLASS**](http://docs.google.com/java/util/AbstractMap.SimpleEntry.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/AbstractMap.html)    [**NO FRAMES**](http://docs.google.com/AbstractMap.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#4d34og8) | [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).