| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SoftReference.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/lang/ref/ReferenceQueue.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/ref/WeakReference.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/ref/SoftReference.html)    [**NO FRAMES**](http://docs.google.com/SoftReference.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#26in1rg) |

## **java.lang.ref**

Class SoftReference<T>

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.lang.ref.Reference](http://docs.google.com/java/lang/ref/Reference.html)<T>  
 **java.lang.ref.SoftReference<T>**

public class **SoftReference<T>**extends [Reference](http://docs.google.com/java/lang/ref/Reference.html)<T>

Soft reference objects, which are cleared at the discretion of the garbage collector in response to memory demand. Soft references are most often used to implement memory-sensitive caches.

Suppose that the garbage collector determines at a certain point in time that an object is [softly reachable](http://docs.google.com/package-summary.html#reachability). At that time it may choose to clear atomically all soft references to that object and all soft references to any other softly-reachable objects from which that object is reachable through a chain of strong references. At the same time or at some later time it will enqueue those newly-cleared soft references that are registered with reference queues.

All soft references to softly-reachable objects are guaranteed to have been cleared before the virtual machine throws an OutOfMemoryError. Otherwise no constraints are placed upon the time at which a soft reference will be cleared or the order in which a set of such references to different objects will be cleared. Virtual machine implementations are, however, encouraged to bias against clearing recently-created or recently-used soft references.

Direct instances of this class may be used to implement simple caches; this class or derived subclasses may also be used in larger data structures to implement more sophisticated caches. As long as the referent of a soft reference is strongly reachable, that is, is actually in use, the soft reference will not be cleared. Thus a sophisticated cache can, for example, prevent its most recently used entries from being discarded by keeping strong referents to those entries, leaving the remaining entries to be discarded at the discretion of the garbage collector.

**Since:** 1.2

| **Constructor Summary** | |
| --- | --- |
| [**SoftReference**](http://docs.google.com/java/lang/ref/SoftReference.html#SoftReference(T))([T](http://docs.google.com/java/lang/ref/SoftReference.html) referent)            Creates a new soft reference that refers to the given object. |
| [**SoftReference**](http://docs.google.com/java/lang/ref/SoftReference.html#SoftReference(T,%20java.lang.ref.ReferenceQueue))([T](http://docs.google.com/java/lang/ref/SoftReference.html) referent, [ReferenceQueue](http://docs.google.com/java/lang/ref/ReferenceQueue.html)<? super [T](http://docs.google.com/java/lang/ref/SoftReference.html)> q)            Creates a new soft reference that refers to the given object and is registered with the given queue. |

| **Method Summary** | |
| --- | --- |
| [T](http://docs.google.com/java/lang/ref/SoftReference.html) | [**get**](http://docs.google.com/java/lang/ref/SoftReference.html#get())()            Returns this reference object's referent. |

| **Methods inherited from class java.lang.ref.**[**Reference**](http://docs.google.com/java/lang/ref/Reference.html) |
| --- |
| [clear](http://docs.google.com/java/lang/ref/Reference.html#clear()), [enqueue](http://docs.google.com/java/lang/ref/Reference.html#enqueue()), [isEnqueued](http://docs.google.com/java/lang/ref/Reference.html#isEnqueued()) |

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

### SoftReference

public **SoftReference**([T](http://docs.google.com/java/lang/ref/SoftReference.html) referent)

Creates a new soft reference that refers to the given object. The new reference is not registered with any queue.

**Parameters:**referent - object the new soft reference will refer to

### SoftReference

public **SoftReference**([T](http://docs.google.com/java/lang/ref/SoftReference.html) referent,  
 [ReferenceQueue](http://docs.google.com/java/lang/ref/ReferenceQueue.html)<? super [T](http://docs.google.com/java/lang/ref/SoftReference.html)> q)

Creates a new soft reference that refers to the given object and is registered with the given queue.

**Parameters:**referent - object the new soft reference will refer toq - the queue with which the reference is to be registered, or null if registration is not required

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

### get

public [T](http://docs.google.com/java/lang/ref/SoftReference.html) **get**()

Returns this reference object's referent. If this reference object has been cleared, either by the program or by the garbage collector, then this method returns null.

**Overrides:**[get](http://docs.google.com/java/lang/ref/Reference.html#get()) in class [Reference](http://docs.google.com/java/lang/ref/Reference.html)<[T](http://docs.google.com/java/lang/ref/SoftReference.html)> **Returns:**The object to which this reference refers, or null if this reference object has been cleared

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SoftReference.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/lang/ref/ReferenceQueue.html)   [**NEXT CLASS**](http://docs.google.com/java/lang/ref/WeakReference.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/ref/SoftReference.html)    [**NO FRAMES**](http://docs.google.com/SoftReference.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#26in1rg) |

[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).