| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/PhantomReference.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   [**NEXT CLASS**](http://docs.google.com/java/lang/ref/Reference.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/ref/PhantomReference.html)    [**NO FRAMES**](http://docs.google.com/PhantomReference.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#17dp8vu) |

## **java.lang.ref**

Class PhantomReference<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.PhantomReference<T>**

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

Phantom reference objects, which are enqueued after the collector determines that their referents may otherwise be reclaimed. Phantom references are most often used for scheduling pre-mortem cleanup actions in a more flexible way than is possible with the Java finalization mechanism.

If the garbage collector determines at a certain point in time that the referent of a phantom reference is [phantom reachable](http://docs.google.com/package-summary.html#reachability), then at that time or at some later time it will enqueue the reference.

In order to ensure that a reclaimable object remains so, the referent of a phantom reference may not be retrieved: The get method of a phantom reference always returns null.

Unlike soft and weak references, phantom references are not automatically cleared by the garbage collector as they are enqueued. An object that is reachable via phantom references will remain so until all such references are cleared or themselves become unreachable.

**Since:** 1.2

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

| **Method Summary** | |
| --- | --- |
| [T](http://docs.google.com/java/lang/ref/PhantomReference.html) | [**get**](http://docs.google.com/java/lang/ref/PhantomReference.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** |
| --- |

### PhantomReference

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

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

It is possible to create a phantom reference with a null queue, but such a reference is completely useless: Its get method will always return null and, since it does not have a queue, it will never be enqueued.

**Parameters:**referent - the object the new phantom 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/PhantomReference.html) **get**()

Returns this reference object's referent. Because the referent of a phantom reference is always inaccessible, this method always 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/PhantomReference.html)> **Returns:**null

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