| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Observable.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/NoSuchElementException.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Observer.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/Observable.html)    [**NO FRAMES**](http://docs.google.com/Observable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **java.util**

Class Observable

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.util.Observable**

public class **Observable**extends [Object](http://docs.google.com/java/lang/Object.html)

This class represents an observable object, or "data" in the model-view paradigm. It can be subclassed to represent an object that the application wants to have observed.

An observable object can have one or more observers. An observer may be any object that implements interface Observer. After an observable instance changes, an application calling the Observable's notifyObservers method causes all of its observers to be notified of the change by a call to their update method.

The order in which notifications will be delivered is unspecified. The default implementation provided in the Observable class will notify Observers in the order in which they registered interest, but subclasses may change this order, use no guaranteed order, deliver notifications on separate threads, or may guarantee that their subclass follows this order, as they choose.

Note that this notification mechanism is has nothing to do with threads and is completely separate from the wait and notify mechanism of class Object.

When an observable object is newly created, its set of observers is empty. Two observers are considered the same if and only if the equals method returns true for them.

**Since:** JDK1.0 **See Also:**[notifyObservers()](http://docs.google.com/java/util/Observable.html#notifyObservers()), [notifyObservers(java.lang.Object)](http://docs.google.com/java/util/Observable.html#notifyObservers(java.lang.Object)), [Observer](http://docs.google.com/java/util/Observer.html), [Observer.update(java.util.Observable, java.lang.Object)](http://docs.google.com/java/util/Observer.html#update(java.util.Observable,%20java.lang.Object))

| **Constructor Summary** | |
| --- | --- |
| [**Observable**](http://docs.google.com/java/util/Observable.html#Observable())()            Construct an Observable with zero Observers. |

| **Method Summary** | |
| --- | --- |
| void | [**addObserver**](http://docs.google.com/java/util/Observable.html#addObserver(java.util.Observer))([Observer](http://docs.google.com/java/util/Observer.html) o)            Adds an observer to the set of observers for this object, provided that it is not the same as some observer already in the set. |
| protected  void | [**clearChanged**](http://docs.google.com/java/util/Observable.html#clearChanged())()            Indicates that this object has no longer changed, or that it has already notified all of its observers of its most recent change, so that the hasChanged method will now return false. |
| int | [**countObservers**](http://docs.google.com/java/util/Observable.html#countObservers())()            Returns the number of observers of this Observable object. |
| void | [**deleteObserver**](http://docs.google.com/java/util/Observable.html#deleteObserver(java.util.Observer))([Observer](http://docs.google.com/java/util/Observer.html) o)            Deletes an observer from the set of observers of this object. |
| void | [**deleteObservers**](http://docs.google.com/java/util/Observable.html#deleteObservers())()            Clears the observer list so that this object no longer has any observers. |
| boolean | [**hasChanged**](http://docs.google.com/java/util/Observable.html#hasChanged())()            Tests if this object has changed. |
| void | [**notifyObservers**](http://docs.google.com/java/util/Observable.html#notifyObservers())()            If this object has changed, as indicated by the hasChanged method, then notify all of its observers and then call the clearChanged method to indicate that this object has no longer changed. |
| void | [**notifyObservers**](http://docs.google.com/java/util/Observable.html#notifyObservers(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) arg)            If this object has changed, as indicated by the hasChanged method, then notify all of its observers and then call the clearChanged method to indicate that this object has no longer changed. |
| protected  void | [**setChanged**](http://docs.google.com/java/util/Observable.html#setChanged())()            Marks this Observable object as having been changed; the hasChanged method will now return true. |

| **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** |
| --- |

### Observable

public **Observable**()

Construct an Observable with zero Observers.

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

### addObserver

public void **addObserver**([Observer](http://docs.google.com/java/util/Observer.html) o)

Adds an observer to the set of observers for this object, provided that it is not the same as some observer already in the set. The order in which notifications will be delivered to multiple observers is not specified. See the class comment.

**Parameters:**o - an observer to be added. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the parameter o is null.

### deleteObserver

public void **deleteObserver**([Observer](http://docs.google.com/java/util/Observer.html) o)

Deletes an observer from the set of observers of this object. Passing null to this method will have no effect.

**Parameters:**o - the observer to be deleted.

### notifyObservers

public void **notifyObservers**()

If this object has changed, as indicated by the hasChanged method, then notify all of its observers and then call the clearChanged method to indicate that this object has no longer changed.

Each observer has its update method called with two arguments: this observable object and null. In other words, this method is equivalent to:

notifyObservers(null)

**See Also:**[clearChanged()](http://docs.google.com/java/util/Observable.html#clearChanged()), [hasChanged()](http://docs.google.com/java/util/Observable.html#hasChanged()), [Observer.update(java.util.Observable, java.lang.Object)](http://docs.google.com/java/util/Observer.html#update(java.util.Observable,%20java.lang.Object))

### notifyObservers

public void **notifyObservers**([Object](http://docs.google.com/java/lang/Object.html) arg)

If this object has changed, as indicated by the hasChanged method, then notify all of its observers and then call the clearChanged method to indicate that this object has no longer changed.

Each observer has its update method called with two arguments: this observable object and the arg argument.

**Parameters:**arg - any object.**See Also:**[clearChanged()](http://docs.google.com/java/util/Observable.html#clearChanged()), [hasChanged()](http://docs.google.com/java/util/Observable.html#hasChanged()), [Observer.update(java.util.Observable, java.lang.Object)](http://docs.google.com/java/util/Observer.html#update(java.util.Observable,%20java.lang.Object))

### deleteObservers

public void **deleteObservers**()

Clears the observer list so that this object no longer has any observers.

### setChanged

protected void **setChanged**()

Marks this Observable object as having been changed; the hasChanged method will now return true.

### clearChanged

protected void **clearChanged**()

Indicates that this object has no longer changed, or that it has already notified all of its observers of its most recent change, so that the hasChanged method will now return false. This method is called automatically by the notifyObservers methods.

**See Also:**[notifyObservers()](http://docs.google.com/java/util/Observable.html#notifyObservers()), [notifyObservers(java.lang.Object)](http://docs.google.com/java/util/Observable.html#notifyObservers(java.lang.Object))

### hasChanged

public boolean **hasChanged**()

Tests if this object has changed.

**Returns:**true if and only if the setChanged method has been called more recently than the clearChanged method on this object; false otherwise.**See Also:**[clearChanged()](http://docs.google.com/java/util/Observable.html#clearChanged()), [setChanged()](http://docs.google.com/java/util/Observable.html#setChanged())

### countObservers

public int **countObservers**()

Returns the number of observers of this Observable object.

**Returns:**the number of observers of this object.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Observable.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/NoSuchElementException.html)   [**NEXT CLASS**](http://docs.google.com/java/util/Observer.html) | [**FRAMES**](http://docs.google.com/index.html?java/util/Observable.html)    [**NO FRAMES**](http://docs.google.com/Observable.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).