Class java.util.Observable

```
java.lang.Object
   |
   +----java.util.Observable
```

public class **Observable**
extends Object

This class should be subclassed by observable objetc, or "data" in the Model−View paradigm. An Observable object may have any number of Observers. Whenever the Observable instance changes, it notifies all of its observers. Notification is done by calling the update() method on all observers.

Version:
1.12, 08/18/95
Author:
Chris Warth

Constructor Index

o **Observable**()

Method Index

o **addObserver**(Observer)
Adds an observer to the observer list.
o **clearChanged**()
Clears an observable change.
o **countObservers**()
Counts the number of observers.
o **deleteObserver**(Observer)
Deletes an observer from the observer list.
o **deleteObservers**()
Deletes observers from the observer list.
o **hasChanged**()
Returns a true boolean if an observable change has occurred.

o **notifyObservers**()
Notifies all observers if an observable change occurs.
o **notifyObservers**(Object)
Notifies all observers of the specified observable change which occurred.
o **setChanged**()
Sets a flag to note an observable change.

Constructors

o Observable

```
public Observable()
```

Methods

o addObserver

```
public synchronized void addObserver(Observer o)
```

Adds an observer to the observer list.
Parameters:
o – the observer to be added

o deleteObserver

```
public synchronized void deleteObserver(Observer o)
```

Deletes an observer from the observer list.
Parameters:
o – the observer to be deleted

o notifyObservers

```
public void notifyObservers()
```

Notifies all observers if an observable change occurs.

o notifyObservers

```
public synchronized void notifyObservers(Object arg)
```

Notifies all observers of the specified observable change which occurred.
Parameters:
arg – what is being notified

o deleteObservers

```
public synchronized void deleteObservers()
```

Deletes observers from the observer list.

o **setChanged**

```
protected synchronized void setChanged()
```

Sets a flag to note an observable change.

o **clearChanged**

```
protected synchronized void clearChanged()
```

Clears an observable change.

o **hasChanged**

```
public synchronized boolean hasChanged()
```

Returns a true boolean if an observable change has occurred.

o **countObservers**

```
public synchronized int countObservers()
```

Counts the number of observers.

All Packages   This Package   Previous   Next