JAVA.UTIL.OBSERVABLE.DELETEOBSERVER METHOD

 Copyright © tutorialspoint.com

Description

The **java.util.Observable.deleteObserver***Observero* method deletes the specified observer **o** from the set of observers of this object.

Declaration

Following is the declaration for **java.util.Observable.deleteObserver** method

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

Parameters

- **o** -- The observer to be deleted from the set of observers.

Return Value

- **NA**

Exception

- **NA**

Example

The following example shows the usage of java.util.Observable.deleteObserver*Observer* method.

```
package com.tutorialspoint;

import java.util.Observable;
import java.util.Observer;

class ObservedObject extends Observable {
   private String watchedValue;

   public ObservedObject(String value) {
   watchedValue = value;
   }

   public void setValue(String value) {
   // if value has changed notify observers
   if(!watchedValue.equals(value)) {
   System.out.println("Value changed to new value: "+value);
   watchedValue = value;

   // mark as value changed
   setChanged();
   // trigger notification
   notifyObservers(value);
   }
   }
}
public class ObservableDemo implements Observer {
   public static void main(String[] args) {
   // create watched and watcher objects
   ObservedObject watched = new ObservedObject("Original Value");
   // watcher object listens to object change
   ObservableDemo watcher = new ObservableDemo();

   // trigger value change
   watched.setValue("Value before addObserver");
```

```
   // add observer to the watched object
   watched.addObserver(watcher);
   // trigger value change
   watched.setValue("Value after addObserver");
   // delete observer
   watched.deleteObserver(watcher);
   // trigger value change
   watched.setValue("Value after deleteObserver");
   }

   public void update(Observable obj, Object arg) {
   System.out.println("Update called with Arguments: "+arg);
   }
}
```

Let us compile and run the above program, this will produce the following result:

```
Value changed to new value: Value before addObserver
Value changed to new value: Value after addObserver
Update called with Arguments: Value after addObserver
Value changed to new value: Value after deleteObserver
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js