

JAVA.UTIL.OBSERVABLE.COUNTOBSERVERS METHOD

http://www.tutorialspoint.com/java/util/observable_countobservers.htm

Copyright © tutorialspoint.com

Description

The **java.util.Observable.countObservers** returns the number of observers of this **Observable** object.

Declaration

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

```
public int countObservers()
```

Parameters

- NA

Return Value

This method returns the number of observers of this object.

Exception

- NA

Example

The following example shows the usage of java.util.Observable.countObservers 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 String name;
    public ObservableDemo(String name) {
        this.name = name;
    }

    public static void main(String[] args) {
        // create watched and watcher objects
        ObservedObject watched = new ObservedObject("Original Value");
```

```

// watcher object listens to object change
ObservableDemo watcher1 = new ObservableDemo("Watcher1");
ObservableDemo watcher2 = new ObservableDemo("Watcher2");

// add observer to the watched object
watched.addObserver(watcher1);
watched.addObserver(watcher2);
// count observers
int count = watched.countObservers();
System.out.println(count+" observers listening");
// trigger value change
watched.setValue("New Value");
// remove first observer
watched.deleteObserver(watcher1);
count = watched.countObservers();
System.out.println(count+" observers listening");
// trigger value change
watched.setValue("Latest Value");
}

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

```

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

```

2 observers listening
Value changed to new value: New Value
Watcher2 called with Arguments: New Value
Watcher1 called with Arguments: New Value
1 observers listening
Value changed to new value: Latest Value
Watcher2 called with Arguments: Latest Value

```

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