

JAVA.UTIL.OBSERVABLE.ADDOBSERVER METHOD

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

Copyright © tutorialspoint.com

Description

The **java.util.addObserver** *Observer*o method adds the specified observer **o** to the set of **observers** for this object if, the observer is not the same as some observer already in the set.

Declaration

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

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

Parameters

- **o** -- The observer to be added.

Return Value

- **NA**

Exception

- **NullPointerException** -- if the specified observer is **null**.

Example

The following example shows the usage of **java.util.Observable.addObserver** *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("New Value");
    }
}
```

```
// add observer to the watched object
watched.addObserver(watcher);

// trigger value change
watched.setValue("Latest Value");
}

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: New Value
Value changed to new value: Latest Value
Update called with Arguments: Latest Value
```

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