



```

1  public interface DisplayElement {
2      public void display();
3  }
4

```

```

1  public interface Observer {
2      public void update(float temp, float humidity, float pressure);
3  }
4

```

bservador src / main / java WeatherData

Project

- Observer C:\Users\gerve\OneDrive\Escritorio\8vo sem
- src
 - main
 - java
 - org.example
 - DisplayElement
 - Observer
 - Subject
 - WeatherData
 - resources
 - test
 - java
 - .gitignore
 - pom.xml
 - External Libraries
 - Scratches and Consoles

1 import ...

3

no usages

4 public class WeatherData implements Subject{

2 usages

5 private float temperature, humidity, pressure;

5 usages

6 private List<Observer> observers;

7

no usages

8 public WeatherData() { observers = new ArrayList(); }

11

no usages

12 @Override

13 public void registerObserver(Observer o) { observers.add(o); }

16

no usages

17 @Override

18 public void removeObserver(Observer o) {

19 int i = observers.indexOf(o);

20 if(i >= 0){

21 observers.remove(o);

22 }

23 }

24

1 usage

25 @Override

26 public void notifyObserver() {

27 for(Observer o: observers){

28 o.update(temperature, humidity, pressure);

```
Observador / src / main / java / WeatherData

Project
  Observador C:\Users\gerve\OneDrive\Escritorio\8vo sem
    .idea
    src
      main
        java
          org.example
            DisplayElement
            Observer
            Subject
            WeatherData
          resources
          test
            java
            .gitignore
            pom.xml
        External Libraries
        Scratches and Consoles

17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

@Override
public void removeObserver(Observer o) {
    int i = observers.indexOf(o);
    if(i >= 0){
        observers.remove(o);
    }
}

1 usage
@Override
public void notifyObserver() {
    for(Observer o: observers){
        o.update(temperature, humidity, pressure);
    }
}

1 usage
public void measurementsChanged(){
    //Más acciones
    notifyObserver();
}

no usages
public void setMeasurements(float temperature, float humidity, float pressure) {
    this.temperature = temperature;
    this.humidity = humidity;
    this.pressure = pressure;

    measurementsChanged();
}
}
```