**Exercise 7: Implementing the Observer Pattern**

**Code:**

**Stock Interface: -**

public interface Stock {  
 void registerObserver(Observer o);  
 void deregisterObserver(Observer o);  
 void notifyObserver();  
}

**StockMarket Class: -**

import java.util.ArrayList;  
import java.util.List;  
public class StockMarket implements Stock{  
 private List<Observer> observerList;  
 private double stockPrice;  
 public StockMarket(){  
 observerList=new ArrayList<>();  
 }  
 public void setStockPrice(double price){  
 this.stockPrice=price;  
 notifyObserver();  
 }  
 @Override  
 public void registerObserver(Observer o) {  
 observerList.add(o);  
 }  
 @Override  
 public void deregisterObserver(Observer o) {  
 observerList.remove(o);  
 }  
 @Override  
 public void notifyObserver() {  
 for (Observer o : observerList) {  
 o.update(stockPrice);  
 }  
 }  
}

**Observer Interface: -**

public interface Observer {  
 void update(double price);  
}

**MobileApp Class: -**

public class MobileApp implements Observer{  
 private String appname;  
  
 public MobileApp(String appname) {  
 this.appname = appname;  
 }  
 @Override  
 public void update(double stockPrice) {  
 System.*out*.println(appname +" received updated stock price: $" + stockPrice);  
 }  
}

**WebApp Class: -**

public class WebApp implements Observer {  
 private String websiteName;  
  
 public WebApp(String websiteName) {  
 this.websiteName = websiteName;  
 }  
 @Override  
 public void update(double stockPrice) {  
 System.*out*.println(websiteName +" received updated stock price: $" + stockPrice);  
 }  
}

**TestClass (Main Class): -**

public class TestClass {  
 public static void main(String[] args){  
 StockMarket stockMarket = new StockMarket();  
 Observer mobile1 = new MobileApp("Groww");  
 Observer mobile2 = new MobileApp("5paisa");  
 Observer web1 = new WebApp("Finnhub");  
  
 stockMarket.registerObserver(mobile1);  
 stockMarket.registerObserver(mobile2);  
 stockMarket.registerObserver(web1);  
  
 stockMarket.setStockPrice(110.50);  
 stockMarket.deregisterObserver(mobile2);  
 stockMarket.setStockPrice(125.75);  
 }  
}

**Output:**

A screenshot of a computer program

AI-generated content may be incorrect.