**Exercise 7: Implementing the Observer Pattern**

**Scenario:**

You are developing a stock market monitoring application where multiple clients need to be notified whenever stock prices change. Use the Observer Pattern to achieve this.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **ObserverPatternExample**.
2. **Define Subject Interface:**
   * Create an interface **Stock** with methods to **register**, **deregister**, and **notify** observers.
3. **Implement Concrete Subject:**
   * Create a class **StockMarket** that implements **Stock** and maintains a list of observers.
4. **Define Observer Interface:**
   * Create an interface Observer with a method **update().**
5. **Implement Concrete Observers:**
   * Create classes **MobileApp**, **WebApp** that implement Observer.
6. **Test the Observer Implementation:**
   * Create a test class to demonstrate the registration and notification of observers.

## Answer:-

Main.java

public class Main {

    public static void main(String[] args) {

        StockMarket sm = new StockMarket();

        Observer m = new MobileApp("Ram");

        Observer w = new WebApp("Sam");

        sm.reg(m);

        sm.reg(w);

        System.out.println("Setting stock price at 150.0");

        sm.setSp(150.0);

        sm.dereg(w);

        System.out.println("Updating stock price at 175.0");

        sm.setSp(175.0);

    }

}

MobileApp.java

public class MobileApp implements Observer {

    String usr;

    public MobileApp(String usr) {

        this.usr = usr;

    }

    @Override

    public void update(double price) {

        System.out.println("MobileApp- Stock Price of user "+usr+" updated: $" + price);

    }

    @Override

    public String toString() {

        return usr;

    }

}

Observer.java

public interface Observer {

    void update(double price);

}

Stock.java

public interface Stock {

    void reg(Observer observer);

    void dereg(Observer observer);

    void notifys();

}

StockMarket.java

import java.util.ArrayList;

import java.util.List;

public class StockMarket implements Stock {

    List<Observer> obs = new ArrayList<>();

    double sp;

    public void setSp(double sp) {

        this.sp = sp;

        notifys();

    }

    @Override

    public void reg(Observer ob) {

        System.out.println("User " +ob+" is registered.");

        obs.add(ob);

    }

    @Override

    public void dereg(Observer ob) {

        System.out.println("User " +ob+" is deregistered.");

        obs.remove(ob);

    }

    @Override

    public void notifys() {

        for (Observer ob : obs) {

            ob.update(sp);

        }

    }

}

WebApp.java

public class WebApp implements Observer {

    String usr;

    public WebApp(String usr) {

        this.usr = usr;

    }

    @Override

    public void update(double price) {

        System.out.println("WebApp- Stock Price of user "+usr+" updated: " + price);

    }

    @Override

    public String toString() {

        return usr;

    }

}

