**Exercise 7: Implementing the Observer Pattern**

import java.util.ArrayList;

import java.util.List;

public class Main {

    public static void main(String[] args) {

        StockMarket stockMarket = new StockMarket();

        Observer mobileApp = new MobileApp("MobileApp");

        Observer webApp = new WebApp("WebApp");

        stockMarket.registerObserver(mobileApp);

        stockMarket.registerObserver(webApp);

        System.out.println("\n--- Updating stock price to ₹1550.75 ---");

        stockMarket.setPrice(1550.75);

        stockMarket.removeObserver(webApp);

        System.out.println("\n--- Updating stock price to ₹1602.10 ---");

        stockMarket.setPrice(1602.10);

    }

}

interface Stock {

    void registerObserver(Observer o);

    void removeObserver(Observer o);

    void notifyObservers();

}

class StockMarket implements Stock {

    private List<Observer> observers = new ArrayList<>();

    private double price;

    @Override

    public void registerObserver(Observer o) {

        observers.add(o);

        System.out.println(o.getName() + " has subscribed.");

    }

    @Override

    public void removeObserver(Observer o) {

        observers.remove(o);

        System.out.println(o.getName() + " has unsubscribed.");

    }

    @Override

    public void notifyObservers() {

        for (Observer o : observers) {

            o.update(price);

        }

    }

    public void setPrice(double newPrice) {

        this.price = newPrice;

        notifyObservers();

    }

}

interface Observer {

    void update(double price);

    String getName();

}

class MobileApp implements Observer {

    private String name;

    public MobileApp(String name) {

        this.name = name;

    }

    @Override

    public void update(double price) {

        System.out.println(name + " received price update: ₹" + price);

    }

    @Override

    public String getName() {

        return name;

    }

}

class WebApp implements Observer {

    private String name;

    public WebApp(String name) {

        this.name = name;

    }

    @Override

    public void update(double price) {

        System.out.println(name + " received price update: ₹" + price);

    }

    @Override

    public String getName() {

        return name;

    }

}

Output:

