**Exercise 7: Observer Pattern – Stock Market Monitoring App**

**Step 1: Subject Interface – Stock.java**

public interface Stock {

void registerObserver(Observer observer);

void removeObserver(Observer observer);

void notifyObservers();

}

**Step 2: Observer Interface – Observer.java**

public interface Observer {

void update(String stockName, double newPrice);

}

**Step 3: Concrete Subject – StockMarket.java**

import java.util.ArrayList;

import java.util.List;

public class StockMarket implements Stock {

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

private String stockName;

private double stockPrice;

public StockMarket(String stockName, double initialPrice) {

this.stockName = stockName;

this.stockPrice = initialPrice;

}

public void registerObserver(Observer observer) {

observers.add(observer);

}

public void removeObserver(Observer observer) {

observers.remove(observer);

}

public void notifyObservers() {

for (Observer o : observers) {

o.update(stockName, stockPrice);

}

}

// Method to simulate price update

public void setPrice(double newPrice) {

System.out.println("\nPrice updated for " + stockName + ": ₹" + newPrice);

this.stockPrice = newPrice;

notifyObservers();

}

}

**Step 4: Concrete Observers:**

**// File: MobileApp.java**

public class MobileApp implements Observer {

private String user;

public MobileApp(String user) {

this.user = user;

}

public void update(String stockName, double newPrice) {

System.out.println("MobileApp - " + user + ": " + stockName + " is now ₹" + newPrice);

}

}

**// File: WebApp.java**

public class WebApp implements Observer {

private String username;

public WebApp(String username) {

this.username = username;

}

public void update(String stockName, double newPrice) {

System.out.println("WebApp - " + username + ": " + stockName + " is now ₹" + newPrice);

}

}

**Step 5: Test Class – TestObserverPattern.java**

public class TestObserverPattern {

public static void main(String[] args) {

// Create stock market

StockMarket reliance = new StockMarket("Reliance", 2800.00);

// Create observers

Observer mobileUser1 = new MobileApp("Anushka");

Observer webUser1 = new WebApp("Investor42");

// Register observers

reliance.registerObserver(mobileUser1);

reliance.registerObserver(webUser1);

// Simulate price changes

reliance.setPrice(2815.50);

reliance.setPrice(2790.00);

// Unregister one observer

reliance.removeObserver(mobileUser1);

// One more price update

reliance.setPrice(2825.75);

}

}

OUTPUT:

A computer screen with white text

AI-generated content may be incorrect.