**Exercise 7: Implementing the Observer Pattern**

**Code:**

import java.util.ArrayList;

import java.util.List;

// Subject interface

interface Stock {

void registerObserver(Observer observer);

void deregisterObserver(Observer observer);

void notifyObservers();

}

// Concrete subject

class StockMarket implements Stock {

private List<Observer> observers;

private String stockName;

private double stockPrice;

public StockMarket(String stockName, double stockPrice) {

this.stockName = stockName;

this.stockPrice = stockPrice;

this.observers = new ArrayList<>();

}

@Override

public void registerObserver(Observer observer) {

observers.add(observer);

System.out.println(observer + " registered.");

}

@Override

public void deregisterObserver(Observer observer) {

observers.remove(observer);

System.out.println(observer + " deregistered.");

}

@Override

public void notifyObservers() {

for (Observer observer : observers) {

observer.update(stockName, stockPrice);

}

}

// Method to change stock price

public void setStockPrice(double stockPrice) {

System.out.println("\nStock price updated: " + stockName + " = " + stockPrice);

this.stockPrice = stockPrice;

notifyObservers();

}

}

// Observer interface

interface Observer {

void update(String stockName, double stockPrice);

}

// Concrete observer - Mobile App

class MobileApp implements Observer {

private String appName;

public MobileApp(String appName) {

this.appName = appName;

}

@Override

public void update(String stockName, double stockPrice) {

System.out.println(appName + " - New price of " + stockName + ": " + stockPrice);

}

@Override

public String toString() {

return "MobileApp{" + appName + '}';

}

}

// Concrete observer - Web App

class WebApp implements Observer {

private String siteName;

public WebApp(String siteName) {

this.siteName = siteName;

}

@Override

public void update(String stockName, double stockPrice) {

System.out.println(siteName + " - New price of " + stockName + ": " + stockPrice);

}

@Override

public String toString() {

return "WebApp{" + siteName + '}';

}

}

// Test class

public class ObserverPatternExample {

public static void main(String[] args) {

StockMarket stockMarket = new StockMarket("ABC Corp", 100.0);

Observer mobileApp = new MobileApp("StockTracker");

Observer webApp = new WebApp("StockWeb");

stockMarket.registerObserver(mobileApp);

stockMarket.registerObserver(webApp);

stockMarket.setStockPrice(105.5);

stockMarket.setStockPrice(110.0);

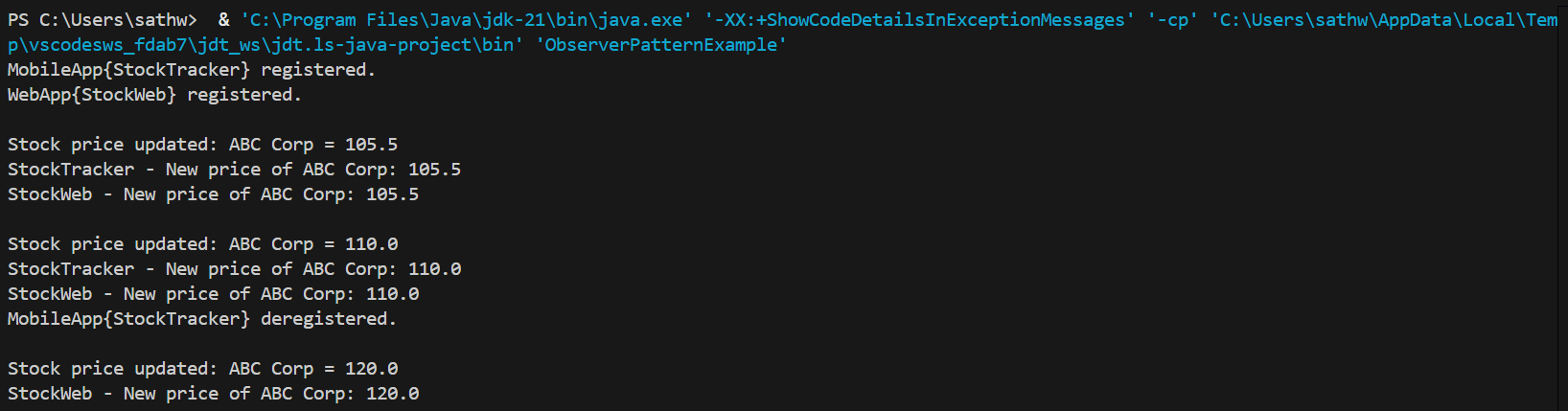
stockMarket.deregisterObserver(mobileApp);

stockMarket.setStockPrice(120.0);

}

}

**Output:**

****