**ObserverPattern Example**

**Main.java**

package com.observer;

public class Main {

    public static void main(String[] args) {

        StockMarket stockMarket = new StockMarket();

        Observer mobileClient = new MobileApp("Alice");

        Observer webClient = new WebApp("Bob");

        stockMarket.registerObserver(mobileClient);

        stockMarket.registerObserver(webClient);

        stockMarket.setStockPrice(150.75);

        System.out.println();

        stockMarket.setStockPrice(157.20);

        System.out.println();

        stockMarket.removeObserver(webClient);

        stockMarket.setStockPrice(160.00);

    }

}

**MobileApp.java**

package com.observer;

public class MobileApp implements Observer {

    private String user;

    public MobileApp(String user) {

        this.user = user;

    }

    @Override

    public void update(double price) {

        System.out.println("MobileApp [" + user + "] - Stock price updated to: $" + price);

    }

}

**Observer.java**

package com.observer;

public interface Observer {

    void update(double price);

}

**Stock.java**

package com.observer;

public interface Stock {

    void registerObserver(Observer o);

    void removeObserver(Observer o);

    void notifyObservers();

}

**StockMarket.java**

package com.observer;

import java.util.ArrayList;

import java.util.List;

public class StockMarket implements Stock {

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

    private double stockPrice;

    public void setStockPrice(double price) {

        this.stockPrice = price;

        System.out.println("Stock price updated to: $" + price);

        notifyObservers();

    }

    @Override

    public void registerObserver(Observer o) {

        observers.add(o);

    }

    @Override

    public void removeObserver(Observer o) {

        observers.remove(o);

    }

    @Override

    public void notifyObservers() {

        for (Observer observer : observers) {

            observer.update(stockPrice);

        }

    }

}

**WebApp.java**

package com.observer;

public class WebApp implements Observer {

    private String user;

    public WebApp(String user) {

        this.user = user;

    }

    @Override

    public void update(double price) {

        System.out.println("WebApp [" + user + "] - Stock price updated to: $" + price);

    }

}

