**Exercise 7: Implementing the Observer Pattern**

**Code:**

**Stock.java**

public interface Stock {

void register(Observer o);

void deregister(Observer o);

void notifyObservers();

}

**Observer.java**

public interface Observer {

void update(String stockName, double price);

}

**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 void setStock(String stockName, double stockPrice) {

this.stockName = stockName;

this.stockPrice = stockPrice;

notifyObservers();

}

@Override

public void register(Observer o) {

observers.add(o);

}

@Override

public void deregister(Observer o) {

observers.remove(o);

}

@Override

public void notifyObservers() {

for (Observer o : observers) {

o.update(stockName, stockPrice);

}

}

}

**MobileApp.java**

public class MobileApp implements Observer {

private String name;

public MobileApp(String name) {

this.name = name;

}

@Override

public void update(String stockName, double price) {

System.out.println(name + " (Mobile): " + stockName + " updated to $" + price);

}

}

**WebApp.java**

public class WebApp implements Observer {

private String name;

public WebApp(String name) {

this.name = name;

}

@Override

public void update(String stockName, double price) {

System.out.println(name + " (Web): " + stockName + " updated to $" + price);

}

}

**ObserverPatternExample.java**

public class ObserverPatternExample {

public static void main(String[] args) {

StockMarket market = new StockMarket();

Observer mobileUser = new MobileApp("Charlie");

Observer webUser = new WebApp("John");

market.register(mobileUser);

market.register(webUser);

market.setStock("AAPL", 285.50);

market.setStock("GOOG", 1732.80);

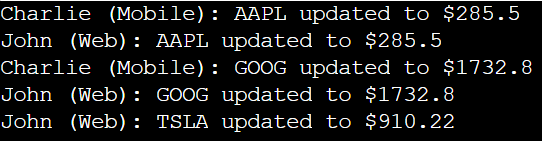
market.deregister(mobileUser);

market.setStock("TSLA", 910.22);

}

}

**Output:**

****