

- [Home](#)
 - [About](#)
 - [Business Plan »](#)
 - [Communication »](#)
 - [Dieting](#)
 - [Sales](#)
 - [Sitemap](#)
 - [Videos »](#)
 - [Web Design »](#)
-
- [Communication »](#)
 - [Diet Nutritional](#)
 - [Flash Tutorial](#)
 - [How To »](#)
 - [Investing](#)
 - [iPad »](#)
 - [Marketing »](#)
 - [Most Popular](#)
 - [Royalty Free Photos](#)
 - [Sales](#)
 - [Web Design »](#)



Mediator Design Pattern Tutorial

Posted by [Derek Banas](#) on Oct 29, 2012 in [Java Video Tutorial](#) | [6 comments](#)



Welcome to my Mediator Design Pattern Tutorial! The Mediator design pattern is used to handle communication between related objects (Colleagues). All communication is handled by a Mediator Object and the Colleagues don't need to know anything about each other to work together.

The Gang of Four description: Allows loose coupling by encapsulating the way disparate sets of objects interact and communicate with each other. Allows for the actions of each object set to vary independently of one another.

If you like videos like this, it helps to tell Google [googleplusone]

Share if you know anyone that would like this

Code from the Video

STOCKOFFER.JAVA

```
01 public class StockOffer{
02
03     private int stockShares = 0;
04     private String stockSymbol = "";
05     private int colleagueCode = 0;
06
07     public StockOffer(int numOfShares, String stock, int collCode){
08
09         stockShares = numOfShares;
10         stockSymbol = stock;
11         colleagueCode = collCode;
12
13     }
14
15     public int getstockShares() { return stockShares; }
16     public String getStockSymbol() { return stockSymbol; }
17     public int getCollCode() { return colleagueCode; }
18
19 }
```

COLLEAGUE.JAVA

```
01 public abstract class Colleague{
02
03     private Mediator mediator;
04     private int colleagueCode;
```

```

05
06
07     public Colleague(Mediator newMediator){
08         mediator = newMediator;
09
10         mediator.addColleague(this);
11
12     }
13
14     public void saleOffer(String stock, int shares){
15
16         mediator.saleOffer(stock, shares, this.colleagueCode);
17
18     }
19
20     public void buyOffer(String stock, int shares){
21
22         mediator.buyOffer(stock, shares, this.colleagueCode);
23
24     }
25
26     public void setCollCode(int collCode){ colleagueCode = collCode; }
27
28
29 }

```

GORMANSLACKS.JAVA

```

01 public class GormanSlacks extends Colleague{
02
03     public GormanSlacks(Mediator newMediator) {
04         super(newMediator);
05
06         System.out.println("Gorman Slacks signed up with the
stockexchange\n");
07     }
08 }
09
10 }

```

JTPOORMAN.JAVA

```

01 public class JTPoorman extends Colleague{
02
03     public JTPoorman(Mediator newMediator) {
04         super(newMediator);
05
06         System.out.println("JT Poorman signed up with the
stockexchange\n");
07     }
08 }
09
10 }

```

MEDIATOR.JAVA

```
1 public interface Mediator {
2
3     public void saleOffer(String stock, int shares, int collCode);
4
5     public void buyOffer(String stock, int shares, int collCode);
6
7     public void addColleague(Colleague colleague);
8
9 }
```

STOCKMEDIATOR.JAVA

```
001 import java.util.ArrayList;
002
003 public class StockMediator implements Mediator{
004
005     private ArrayList<Colleague> colleagues;
006     private ArrayList<StockOffer> stockBuyOffers;
007     private ArrayList<StockOffer> stockSaleOffers;
008
009     private int colleagueCodes = 0;
010
011     public StockMediator(){
012
013         colleagues = new ArrayList<Colleague>();
014         stockBuyOffers = new ArrayList<StockOffer>();
015         stockSaleOffers = new ArrayList<StockOffer>();
016     }
017
018     public void addColleague(Colleague newColleague){
019
020         colleagues.add(newColleague);
021
022         colleagueCodes++;
023
024         newColleague.setCollCode(colleagueCodes);
025
026     }
027
028     public void saleOffer(String stock, int shares, int collCode) {
029
030         boolean stockSold = false;
031
032         for(StockOffer offer: stockBuyOffers){
033
034             if((offer.getStockSymbol() == stock) &&
035 (offer.getstockShares() == shares)){
036
037                 System.out.println(shares + " shares of " + stock +
038 " sold to colleague code " + offer.getCollCode());
039
040                 stockBuyOffers.remove(offer);
041             }
042         }
043     }
044 }
```

```

040
041         stockSold = true;
042
043     }
044
045     if(stockSold){ break; }
046
047 }
048
049 if(!stockSold) {
050
051     System.out.println(shares + " shares of " + stock +
052         " added to inventory");
053
054     StockOffer newOffering = new StockOffer(shares, stock,
collCode);
055
056     stockSaleOffers.add(newOffering);
057
058 }
059
060 }
061
062 public void buyOffer(String stock, int shares, int collCode) {
063
064     boolean stockBought = false;
065
066     for(StockOffer offer: stockSaleOffers){
067
068         if((offer.getStockSymbol() == stock) &&
(offer.getstockShares() == shares)){
069
070             System.out.println(shares + " shares of " + stock +
071                 " bought by colleague code " +
offer.getCollCode());
072
073             stockSaleOffers.remove(offer);
074
075             stockBought = true;
076
077         }
078
079         if(stockBought){ break; }
080
081     }
082
083     if(!stockBought) {
084
085         System.out.println(shares + " shares of " + stock +
086             " added to inventory");
087
088         StockOffer newOffering = new StockOffer(shares, stock,
collCode);
089

```

```

090         stockBuyOffers.add(newOffering);
091     }
092 }
093
094 }
095
096 public void getstockOfferings(){
097     System.out.println("\nStocks for Sale");
098
099     for(StockOffer offer: stockSaleOffers){
100
101         System.out.println(offer.getstockShares() + " of " +
102 offer.getStockSymbol());
103     }
104
105     System.out.println("\nStock Buy Offers");
106
107     for(StockOffer offer: stockBuyOffers){
108
109         System.out.println(offer.getstockShares() + " of " +
110 offer.getStockSymbol());
111     }
112 }
113
114 }
115
116 }

```

TESTSTOCKMEDIATOR.JAVA

```

01 public class TestStockMediator{
02
03     public static void main(String[] args){
04
05         StockMediator nyse = new StockMediator();
06
07         GormanSlacks broker = new GormanSlacks(nyse);
08
09         JTPoorman broker2 = new JTPoorman(nyse);
10
11         broker.saleOffer("MSFT", 100);
12         broker.saleOffer("GOOG", 50);
13
14         broker2.buyOffer("MSFT", 100);
15         broker2.saleOffer("NRG", 10);
16
17         broker.buyOffer("NRG", 10);
18
19         nyse.getstockOfferings();
20
21     }
22

```

6 Responses to “Mediator Design Pattern Tutorial”



1. *atul* says:

[February 17, 2013 at 3:05 am](#)

small correction in buyOffer() function.

```
System.out.println(shares + " shares of " + stock +
" bought by colleague code " + collCode);
```

collCode instead of offer.getCollCode()

[Reply](#)



o *WhistlerHusky* says:

[March 2, 2014 at 6:24 am](#)

I think you can use both of them 😊

[Reply](#)



o *Priyadarshan* says:

[July 19, 2014 at 3:01 am](#)

Yes Atul u r right .. because a buyer cannot buy the shares he is selling .. buy share offer operation is initiated by Colleague 2 i.e. JP Morgan!!

@Derek – Good practicc al example great tutorial!!

[Reply](#)



■ *Derek Banas* says:

[July 21, 2014 at 6:02 pm](#)

Thank you 😊

[Reply](#)



2. *Mike* says:

[June 27, 2013 at 4:34 am](#)

Hi Derek!

I want to thank you very much for the effort you make preparing these tutorials. There are very few people

on the web that are both willing to teach and along with that know what they're doing. These videos are great supplement to the GOF's book.

BTW, this is the first video that was really hard to grasp. The reason was the financial theme you've chosen to deal with. It took me 5 minutes to understand the pattern, but 20 minutes to get into broker-shares-sale-buy relationship 😊 I'm definitely dumb in finance..

Regards,
Mike

[Reply](#)



o [Derek Banas](#) says:
[June 28, 2013 at 12:47 pm](#)

Hi Mike,

Sorry about the weird example. I have been doing my best to avoid weird tutorials like that in the future. I'm happy you were able to understand it though in the end 😊

[Reply](#)

Leave a Reply

Your email address will not be published.

Comment

Name

Email

Website

Search

Help Me Make Free Education

Social Networks

Facebook

YouTube

Twitter

LinkedIn

Buy me a Cup of Coffee

"Donations help me to keep the site running. One dollar is greatly appreciated." - (Pay Pal Secured)

My Facebook Page

Archives

- [March 2022](#)
- [February 2022](#)
- [January 2022](#)
- [June 2021](#)
- [May 2021](#)
- [April 2021](#)
- [March 2021](#)
- [February 2021](#)
- [January 2021](#)
- [December 2020](#)
- [November 2020](#)
- [October 2020](#)
- [September 2020](#)
- [August 2020](#)
- [July 2020](#)
- [June 2020](#)
- [May 2020](#)
- [April 2020](#)
- [March 2020](#)
- [February 2020](#)
- [January 2020](#)
- [December 2019](#)
- [November 2019](#)
- [October 2019](#)
- [August 2019](#)
- [July 2019](#)
- [June 2019](#)
- [May 2019](#)
- [April 2019](#)
- [March 2019](#)
- [February 2019](#)
- [January 2019](#)
- [December 2018](#)
- [October 2018](#)
- [September 2018](#)
- [August 2018](#)
- [July 2018](#)
- [June 2018](#)
- [May 2018](#)
- [April 2018](#)
- [March 2018](#)
- [February 2018](#)
- [January 2018](#)
- [December 2017](#)
- [November 2017](#)

- [October 2017](#)
- [September 2017](#)
- [August 2017](#)
- [July 2017](#)
- [June 2017](#)
- [May 2017](#)
- [April 2017](#)
- [March 2017](#)
- [February 2017](#)
- [January 2017](#)
- [December 2016](#)
- [November 2016](#)
- [October 2016](#)
- [September 2016](#)
- [August 2016](#)
- [July 2016](#)
- [June 2016](#)
- [May 2016](#)
- [April 2016](#)
- [March 2016](#)
- [February 2016](#)
- [January 2016](#)
- [December 2015](#)
- [November 2015](#)
- [October 2015](#)
- [September 2015](#)
- [August 2015](#)
- [July 2015](#)
- [June 2015](#)
- [May 2015](#)
- [April 2015](#)
- [March 2015](#)
- [February 2015](#)
- [January 2015](#)
- [December 2014](#)
- [November 2014](#)
- [October 2014](#)
- [September 2014](#)
- [August 2014](#)
- [July 2014](#)
- [June 2014](#)
- [May 2014](#)
- [April 2014](#)
- [March 2014](#)
- [February 2014](#)
- [January 2014](#)
- [December 2013](#)
- [November 2013](#)
- [October 2013](#)
- [September 2013](#)
- [August 2013](#)
- [July 2013](#)
- [June 2013](#)
- [May 2013](#)

- [April 2013](#)
- [March 2013](#)
- [February 2013](#)
- [January 2013](#)
- [December 2012](#)
- [November 2012](#)
- [October 2012](#)
- [September 2012](#)
- [August 2012](#)
- [July 2012](#)
- [June 2012](#)
- [May 2012](#)
- [April 2012](#)
- [March 2012](#)
- [February 2012](#)
- [January 2012](#)
- [December 2011](#)
- [November 2011](#)
- [October 2011](#)
- [September 2011](#)
- [August 2011](#)
- [July 2011](#)
- [June 2011](#)
- [May 2011](#)
- [April 2011](#)
- [March 2011](#)
- [February 2011](#)
- [January 2011](#)
- [December 2010](#)
- [November 2010](#)
- [October 2010](#)
- [September 2010](#)
- [August 2010](#)
- [July 2010](#)
- [June 2010](#)
- [May 2010](#)
- [April 2010](#)
- [March 2010](#)
- [February 2010](#)
- [January 2010](#)
- [December 2009](#)