

- [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 »](#)



Facade Design Pattern Tutorial

Posted by [Derek Banas](#) on Sep 29, 2012 in [Java Video Tutorial](#) | [15 comments](#)



Welcome to my Facade Design Pattern Tutorial! The Facade pattern is extremely easy to understand. Chances are you have used it already and just didn't know it.

The Facade pattern basically says that you should simplify your methods so that much of what is done is in the background. In technical terms you should decouple the client from the sub components needed to perform an operation.

All of the code that follows the video will fill you in on anything you don't catch in the video.

If you like videos like this, tell Google by pressing this button [googleplusone]

Share this if you'd like

Code from the Video

WELCOMETOBANK.JAVA

```
01 public class WelcomeToBank{
02
03     public WelcomeToBank() {
04
05         System.out.println("Welcome to ABC Bank");
06         System.out.println("We are happy to give you your money if we can
find it\n");
07
08
09     }
10
11 }
```

ACCOUNTNUMBERCHECK.JAVA

```
01 public class AccountNumberCheck{
02
03     private int accountNumber = 12345678;
04
05     public int getAccountNumber() { return accountNumber; }
06
07     public boolean accountActive(int acctNumToCheck){
08
09         if(acctNumToCheck == getAccountNumber()) {
10
11             return true;
```

```

12
13         } else {
14
15             return false;
16
17         }
18
19     }
20
21 }

```

SECURITYCODECHECK.JAVA

```

01 public class SecurityCodeCheck {
02
03     private int securityCode = 1234;
04
05     public int getSecurityCode() { return securityCode; }
06
07     public boolean isCodeCorrect(int secCodeToCheck){
08
09         if(secCodeToCheck == getSecurityCode()) {
10
11             return true;
12
13         } else {
14
15             return false;
16
17         }
18
19     }
20
21 }

```

FUNDSCHECK.JAVA

```

01 public class FundsCheck {
02
03     private double cashInAccount = 1000.00;
04
05     public double getCashInAccount() { return cashInAccount; }
06
07     public void decreaseCashInAccount(double cashWithdrawn) {
08         cashInAccount -= cashWithdrawn; }
09
10     public void increaseCashInAccount(double cashDeposited) {
11         cashInAccount += cashDeposited; }
12
13     public boolean haveEnoughMoney(double cashToWithdrawal) {
14
15         if(cashToWithdrawal > getCashInAccount()) {
16
17             System.out.println("Error: You don't have enough money");
18
19         }
20
21     }
22 }

```

```

16         System.out.println("Current Balance: " + getCashInAccount());
17
18         return false;
19
20     } else {
21
22         decreaseCashInAccount(cashToWithdrawal);
23
24         System.out.println("Withdrawal Complete: Current Balance is "
+ getCashInAccount());
25
26         return true;
27
28     }
29
30 }
31
32 public void makeDeposit(double cashToDeposit) {
33
34     increaseCashInAccount(cashToDeposit);
35
36     System.out.println("Deposit Complete: Current Balance is " +
getCashInAccount());
37
38 }
39
40 }

```

BANKACCOUNTFACADE.JAVA

```

01 // The Facade Design Pattern decouples or separates the client
02 // from all of the sub components
03
04 // The Facades aim is to simplify interfaces so you don't have
05 // to worry about what is going on under the hood
06
07 public class BankAccountFacade {
08
09     private int accountNumber;
10     private int securityCode;
11
12     AccountNumberCheck acctChecker;
13     SecurityCodeCheck codeChecker;
14     FundsCheck fundChecker;
15
16     WelcomeToBank bankWelcome;
17
18     public BankAccountFacade(int newAcctNum, int newSecCode){
19
20         accountNumber = newAcctNum;
21         securityCode = newSecCode;
22
23         bankWelcome = new WelcomeToBank();
24

```

```

25         acctChecker = new AccountNumberCheck();
26         codeChecker = new SecurityCodeCheck();
27         fundChecker = new FundsCheck();
28
29     }
30
31     public int getAccountNumber() { return accountNumber; }
32
33     public int getSecurityCode() { return securityCode; }
34
35
36     public void withdrawCash(double cashToGet){
37
38         if(acctChecker.accountActive(getAccountNumber()) &&
39             codeChecker.isCodeCorrect(getSecurityCode()) &&
40             fundChecker.haveEnoughMoney(cashToGet)) {
41
42             System.out.println("Transaction Complete\n");
43
44         } else {
45
46             System.out.println("Transaction Failed\n");
47
48         }
49     }
50
51
52
53     public void depositCash(double cashToDeposit){
54
55         if(acctChecker.accountActive(getAccountNumber()) &&
56             codeChecker.isCodeCorrect(getSecurityCode())) {
57
58             fundChecker.makeDeposit(cashToDeposit);
59
60             System.out.println("Transaction Complete\n");
61
62         } else {
63
64             System.out.println("Transaction Failed\n");
65
66         }
67     }
68
69
70 }

```

TESTBANKACCOUNT.JAVA

```

01 public class TestBankAccount {
02
03     public static void main(String[] args){
04
05         BankAccountFacade accessingBank = new BankAccountFacade(12345678,

```

```

06 1234);
07     accessingBank.withdrawCash(50.00);
08
09     accessingBank.withdrawCash(990.00);
10
11 }
12
13 }

```

15 Responses to “Facade Design Pattern Tutorial”



1. [Raj](#) says:

[June 11, 2013 at 5:50 am](#)

Good one. Thanks Derek Banas

[Reply](#)



o [Derek Banas](#) says:

[June 11, 2013 at 6:22 pm](#)

You're very welcome 😊

[Reply](#)



2. [Roandmy](#) says:

[June 12, 2013 at 9:35 am](#)

Thanks.! 😊

[Reply](#)



o [Derek Banas](#) says:

[June 14, 2013 at 7:09 am](#)

You're very welcome 😊

[Reply](#)



3. [Anthony Trần](#) says:

[September 11, 2013 at 12:55 am](#)

I really admire and appreciate your work but i have a question: You said facade uses a simplified interface to decouples or separates the client from all the sub components. But i don't see any interface in the code. Why is that?

[Reply](#)



- o [Derek Banas](#) says:
[September 11, 2013 at 5:29 pm](#)

Thank you 😊 I'm sorry about the confusion. I wasn't referring to a java interface, but instead a general interface like a GUI. I should have used a different word. Sorry about that.

[Reply](#)



4. [Abhishek](#) says:
[October 2, 2013 at 11:28 am](#)

Very nicely explained. Thanks Derek 😊

[Reply](#)



- o [Derek Banas](#) says:
[October 3, 2013 at 10:25 am](#)

You're very welcome 😊 Thank you

[Reply](#)



5. [jj](#) says:
[December 9, 2013 at 5:13 pm](#)

You made it very clear with simple examples.
now I linked Facade – Withdraw money , Command – remote etc. easy to understand 😊 thanks a lot
Derek

[Reply](#)



- o [Derek Banas](#) says:
[December 10, 2013 at 6:11 pm](#)

Thank you 😊 I'm happy that I could help.

[Reply](#)



6. [narayana](#) says:
[December 15, 2013 at 4:06 pm](#)

As you said, we use this pattern many places in our code, but never realized it is Façade design pattern.
Thanks for making it clear.

[Reply](#)



7. *Shahad* says:
[February 25, 2014 at 3:43 pm](#)

THANK YOU
U R MUCH MORE BETTER THAN MY OWN TEACHER =O

[Reply](#)



- o *Derek Banas* says:
[February 26, 2014 at 6:25 pm](#)

Thank you very much 😊

[Reply](#)



8. *Malik* says:
[November 13, 2014 at 3:37 am](#)

Thanks a lot , very good example and nicely explained.

[Reply](#)



- o *Derek Banas* says:
[November 13, 2014 at 12:11 pm](#)

You're very welcome 😊 Thank you

[Reply](#)

Leave a Reply

Your email address will not be published.

Comment

Name

Email

Website

[Submit Comment](#)

Search

[Search](#)

Help Me Make Free Education

[Donate Crypto](#)

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)

[Donate](#)



[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](#)

Powered by [WordPress](#) | Designed by [Elegant Themes](#)
[About the Author](#) [Google+](#)