

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



# 私塾在线 《研磨设计模式》 ——跟着CC学设计系列精品教程

10101010101010101010101010101

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 本节课程概览

### n 学习桥接模式

#### 一：初识桥接模式

包括：定义、结构、参考实现

#### 二：体会桥接模式

包括：场景问题、不用模式的解决方案、使用模式的解决方案

#### 三：理解桥接模式

包括：认识桥接模式、谁来桥接、典型例子-JDBC 、

广义桥接-Java中无处不桥接 、桥接模式的优缺点

#### 四：思考桥接模式

包括：桥接模式的本质、对设计原则的体现、何时选用

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

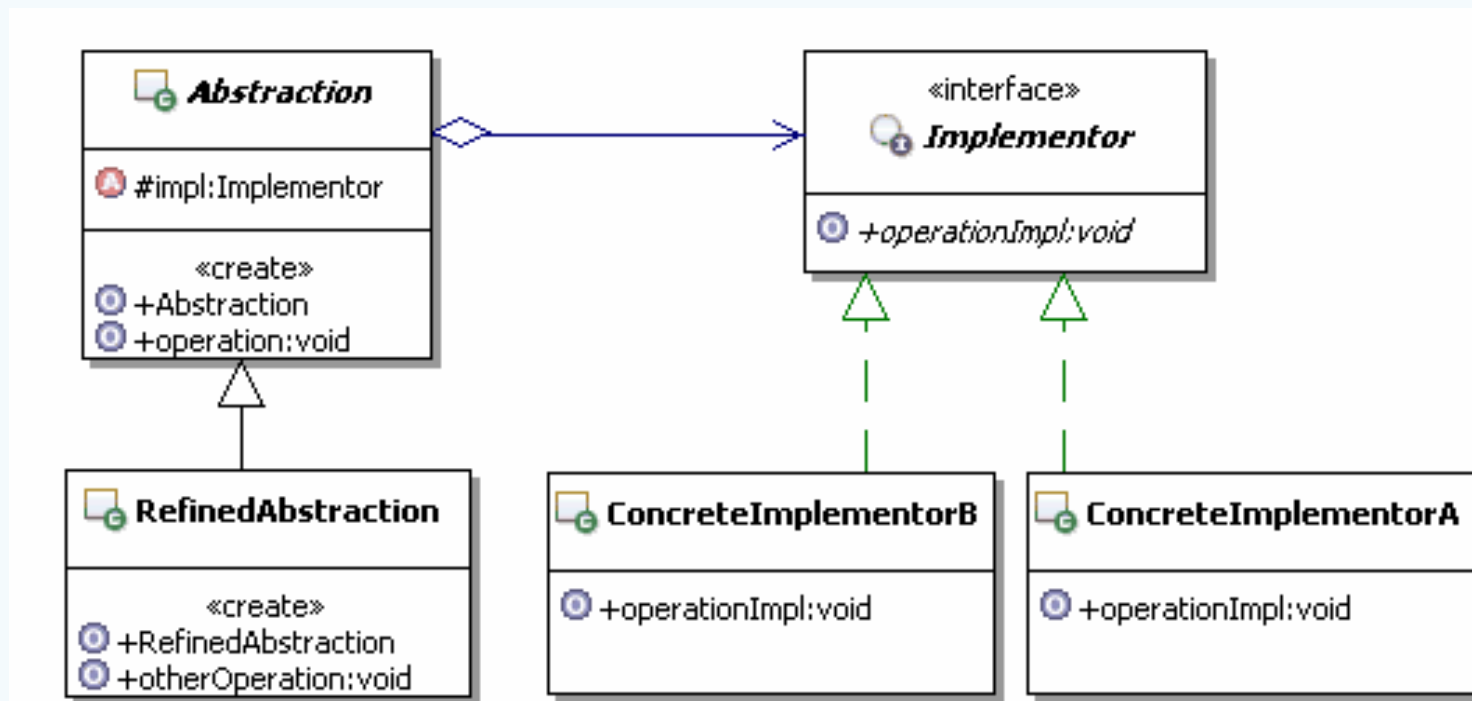
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 初识桥接模式

### n 定义

将抽象部分与它的实现部分分离，使它们都可以独立地变化。

### n 结构和说明



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 初识桥接模式

### Abstraction:

抽象部分的接口。通常在这个对象里面，要维护一个实现部分的对象引用，在抽象对象里面的方法，需要调用实现部分的对象来完成。这个对象里面的方法，通常都是跟具体的业务相关的方法。

### RefinedAbstraction:

扩展抽象部分的接口，通常在这些对象里面，定义跟实际业务相关的方法，这些方法的实现通常会使用Abstraction中定义的方法，也可能需要调用实现部分的对象来完成。

### Implementor:

定义实现部分的接口，这个接口不用和Abstraction里面的方法一致，通常是由Implementor接口提供基本的操作，而Abstraction里面定义的是基于这些基本操作的业务方法，也就是说Abstraction定义了基于这些基本操作的较高层次的操作。

ConcreteImplementor: 真正实现Implementor接口的对象。

做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 体会桥接模式

### n 发送提示信息

考虑这样一个实际的业务功能：发送提示信息。基本上所有带业务流程处理的系统都会有这样的功能，比如某人有了新的工作了，需要发送一条消息提示他。

从业务上看，消息又分成普通消息、加急消息和特急消息多种，不同的消息类型，业务功能处理是不一样的，比如加急消息是在消息上添加加急，而特急消息除了添加特急外，还会做一条催促的记录，多久不完成会继续催促。从发送消息的手段上看，又有系统内短消息、手机短消息、邮件等等。

现在要实现这样的发送提示信息的功能，该如何实现呢？

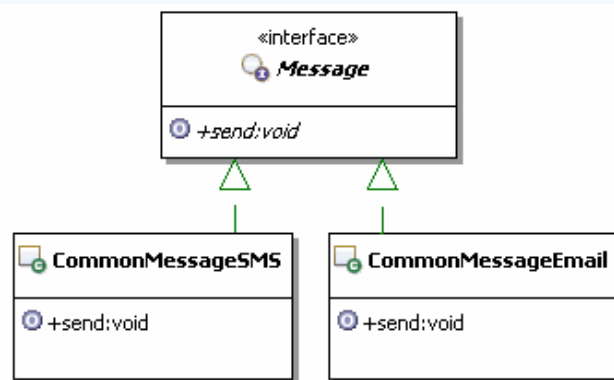
## 体会桥接模式

### n 不用模式的解决方案

#### 1: 实现简化版本

先考虑实现一个简单点的版本，比如：消息先只是实现发送普通消息，发送的方式呢，先实现系统内短消息和邮件。其它的功能，等这个版本完成过后，再继续添加，这样先把问题简单化，实现起来会容易一点。

(1) 由于发送普通消息会有两种不同的实现方式，为了让外部能统一操作，因此，把消息设计成接口，然后由两个不同的实现类，分别实现系统内短消息方式和邮件发送消息的方式。此时系统结构如图



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

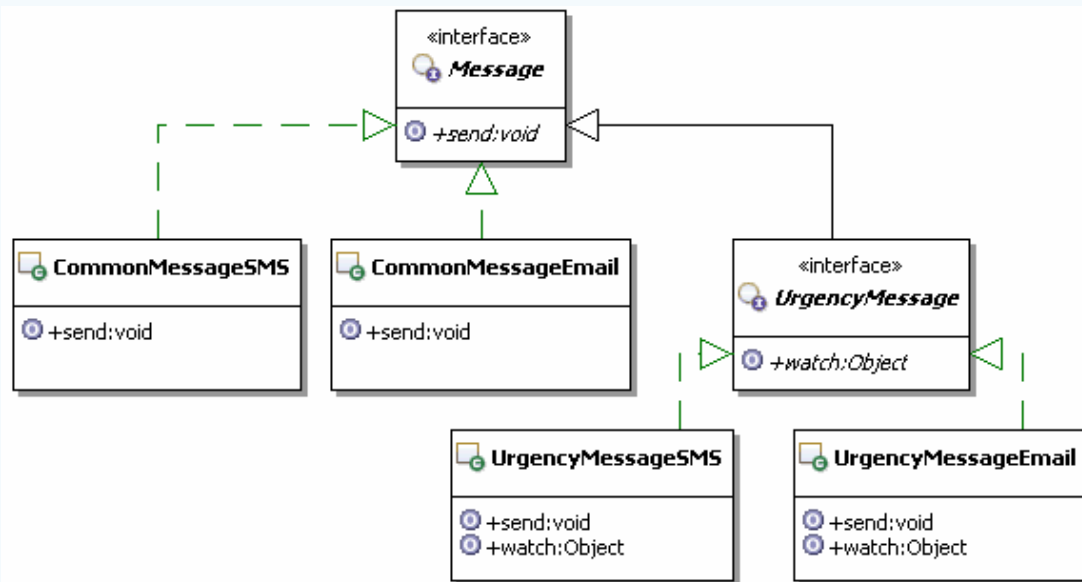


## 体会桥接模式

### 2: 实现发送加急消息

添加发送加急消息的功能，同样是站内短消息和Email的方式。

加急消息的实现跟普通消息不同，加急消息会自动在消息上添加加急，然后再发送消息；另外加急消息会提供监控的方法，让客户端可以随时通过这个方法了解对于加急消息处理的进度等。因此加急消息需要扩展出一个新的接口，除了基本的发送消息的功能，还需要添加监控的功能，这个时候，系统的结构如图：



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 体会桥接模式

### n 有何问题

上面这样实现，好像也能满足基本的功能要求，可是这么实现好不好呢？有没有什么问题呢？

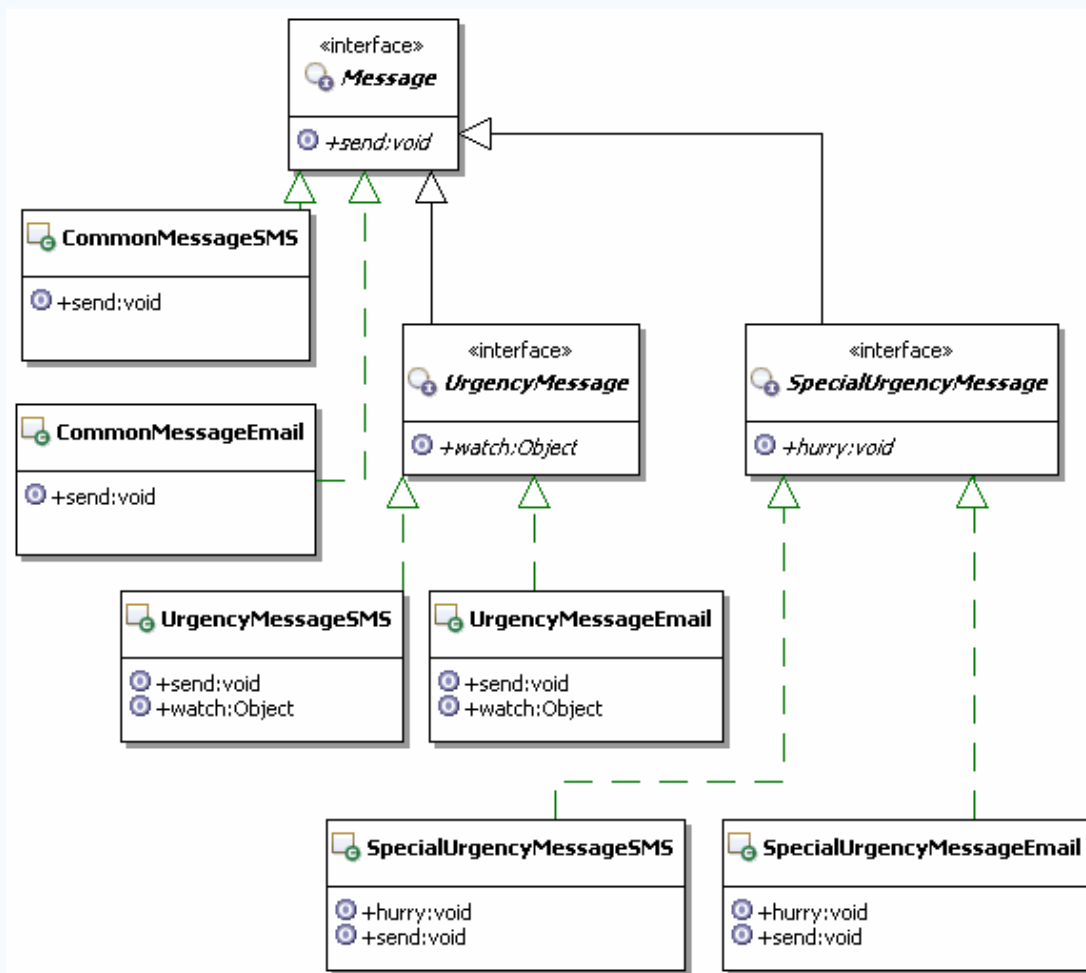
咱们继续向下来添加功能实现，为了简洁，就不再去进行代码示意了，通过实现的结构示意图就可以看出实现上的问题。

### 1: 继续添加特急消息的处理

特急消息不需要查看处理进程，只要没有完成，就直接催促，也就是说，对于特急消息，在普通消息的处理基础上，需要添加催促的功能。而特急消息、还有催促的发送方式，相应的实现方式还是发送站内短消息和Email两种，此时系统的结构如图：



## 体会桥接模式



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 体会桥接模式

仔细观察上面的系统结构示意图，会发现一个很明显的问题，那就是：通过这种继承的方式来扩展消息处理，会非常不方便。

你看，实现加急消息处理的时候，必须实现站内短消息和Email两种处理方式，因为业务处理可能不同；在实现特急消息处理的时候，又必须实现站内短消息和Email这两种处理方式。

这意味着，以后每次扩展一下消息处理，都必须要实现这两种处理方式，是不是很痛苦，这还不算完，如果要添加新的实现方式呢？继续向下看吧。

### 2: 继续添加发送手机消息的处理方式

如果看到上面的实现，你还感觉问题不是很大的话，继续完成功能，添加发送手机消息的处理方式。

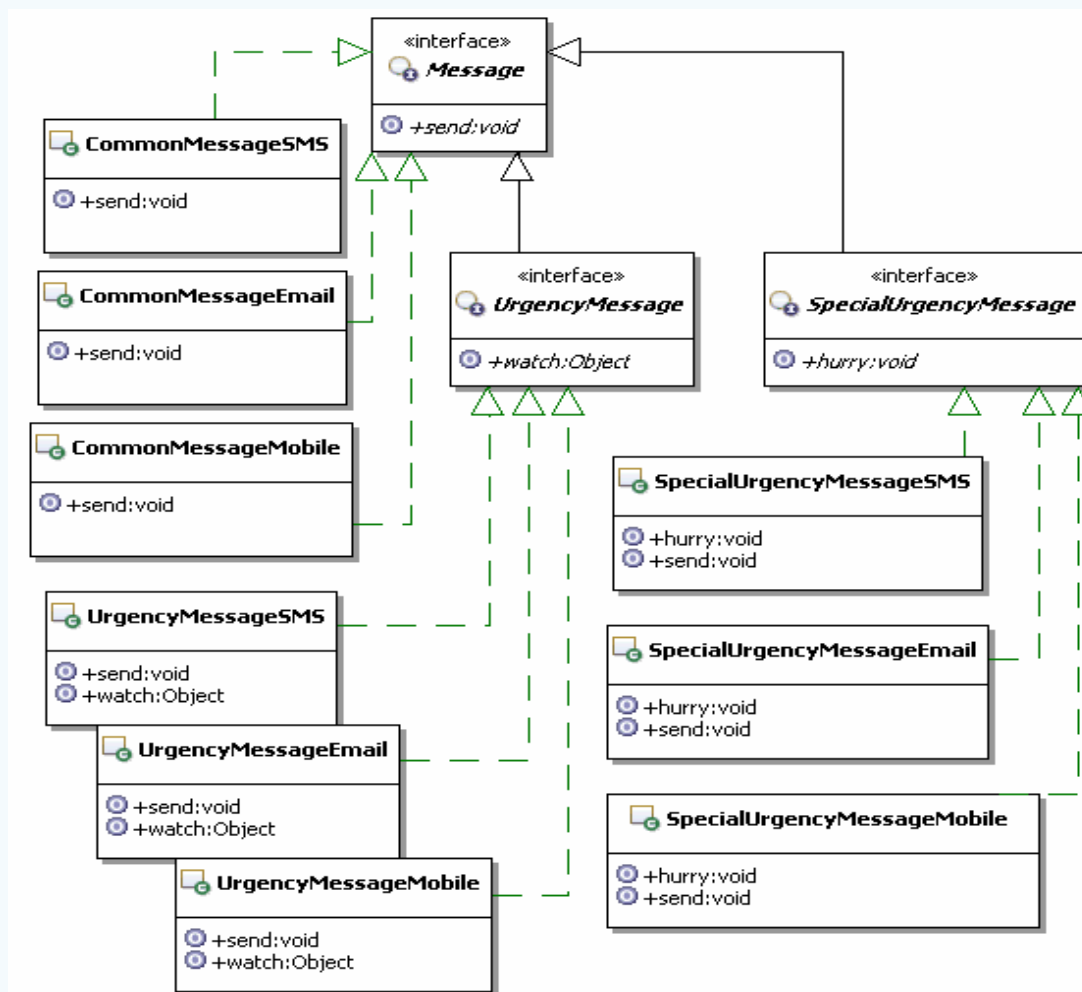
仔细观察现在的实现，如果要添加一种新的发送消息的方式，是需要每一种抽象的具体实现里面，都要添加发送手机消息的处理的。也就是说：发送普通消息、加急消息和特急消息的处理，都可以通过手机来发送。这就意味着，需要添加三个实现。此时系统结构如图：

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

## 体会桥接模式



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 体会桥接模式

### 3: 小结一下出现的问题

采用通过继承来扩展的实现方式，有个明显的缺点：扩展消息的种类不太容易，不同种类的消息具有不同的业务，也就是有不同的实现，在这种情况下，每个种类的消息，需要实现所有不同的消息发送方式。

更可怕的是，如果要新加入一种消息的发送方式，那么会要求所有的消息种类，都要加入这种新的发送方式的实现。

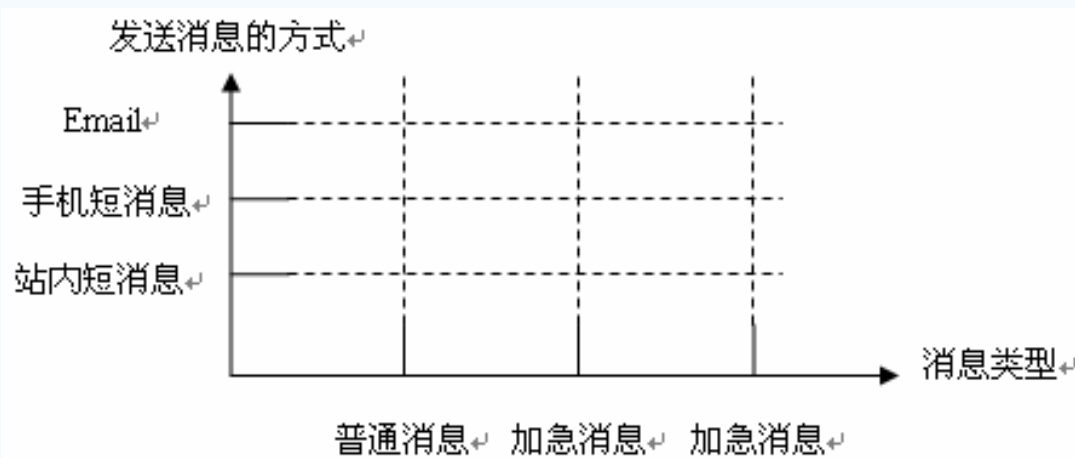
要是考虑业务功能上再扩展一下呢？比如：要求实现群发消息，也就是一次可以发送多条消息，这就意味着很多地方都得修改，太恐怖了。

那么究竟该如何实现才能既实现功能，又能灵活的扩展呢？

## 体会桥接模式

### n 使用模式来解决的思路

仔细分析上面的示例，根据示例的功能要求，示例的变化具有两个纬度，一个纬度是抽象的消息这边，包括普通消息、加急消息和特急消息，这几个抽象的消息本身就具有一定的关系，加急消息和特急消息会扩展普通消息；另一个纬度在具体的消息发送方式上，包括站内短消息、Email和手机短信息，这几个方式是平等的，可被切换的方式。这两个纬度一共可以组合出9种不同的可能性来，它们的关系如下图



做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 体会桥接模式

现在出现问题的根本原因，就在于消息的抽象和实现是混杂在一起的，这就导致了，一个纬度的变化，会引起另一个纬度进行相应的变化，从而使得程序扩展起来非常困难。

要想解决这个问题，就必须把这两个纬度分开，也就是将抽象部分和实现部分分开，让它们相互独立，这样就可以实现独立的变化，使扩展变得简单。

桥接模式通过引入实现的接口，把实现部分从系统中分离出去；那么，抽象这边如何使用具体的实现呢？肯定是面向实现的接口来编程了，为了让抽象这边能够很方便的与实现结合起来，把顶层的抽象接口改成抽象类，在里面持有一个具体的实现部分的实例。

这样一来，对于需要发送消息的客户端而言，就只需要创建相应的消息对象，然后调用这个消息对象的方法就可以了，这个消息对象会调用持有的真正的消息发送方式来把消息发送出去。也就是说客户端只是想要发送消息而已，并不想关心具体如何发送。



## 体会桥接模式

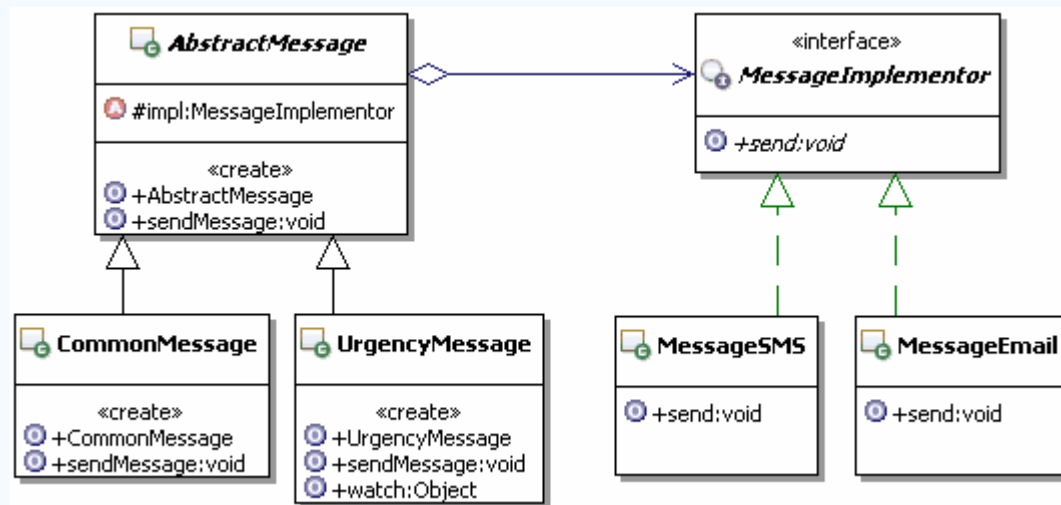
### n 使用模式来解决

首要任务是把抽象部分和实现部分分离出来，分析要实现的功能，抽象部分就是各个消息的类型所对应的功能，而实现部分就是各种发送消息的方式。

其次要给抽象部分和实现部分分别定义接口，然后分别实现它们。

#### 1: 从简单功能开始

从相对简单的功能开始，先实现普通消息和加急消息的功能，发送方式先实现站内短消息和Email这两种。使用桥接模式来实现这些功能的程序结构如图



做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 体会桥接模式

### 2: 添加功能

看了上面的实现，发现使用桥接模式来实现也不是很困难啊，关键得看是否能解决前面提出的问题，那就来添加还未实现的功能看看，添加对特急消息的处理，同时添加一个使用手机发送消息的方式。该怎么实现呢？

很简单，只需要在抽象部分再添加一个特急消息的类，扩展抽象消息就可以把特急消息的处理功能加入到系统中了；对于添加手机发送消息的方式也很简单，在实现部分新增加一个实现类，实现用手机发送消息的方式，也就可以了。

这么简单？好像看起来完全没有了前面所提到的问题。的确如此，采用桥接模式来实现过后，抽象部分和实现部分分离开了，可以相互独立的变化，而不会相互影响。因此在抽象部分添加新的消息处理，对发送消息的实现部分是没有影响的；反过来增加发送消息的方式，对消息处理部分也是没有影响的。

## 理解桥接模式

### n 认识桥接模式

#### 1: 什么是桥接

在桥接模式里面，不太好理解的就是桥接的概念，什么是桥接？为何需要桥接？如何桥接？把这些问题搞清楚了，也就基本明白桥接的含义了。

一个一个来，先看什么是桥接？所谓桥接，通俗点说就是在不同的东西之间搭一个桥，让他们能够连接起来，可以相互通讯和使用。那么在桥接模式中到底是给什么东西来搭桥呢？就是为被分离了的抽象部分和实现部分来搭桥，比如前面示例中抽象的消息和具体消息发送之间搭个桥。

但是这里要注意一个问题：在桥接模式中的桥接是单向的，也就是只能是抽象部分的对象去使用具体实现部分的对象，而不能反过来，也就是个单向桥。

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解桥接模式

### 2: 为何需要桥接

为了达到让抽象部分和实现部分都可以独立变化的目的，在桥接模式中，是把抽象部分和实现部分分离开来的，虽然从程序结构上是分开了，但是在抽象部分实现的时候，还是需要使用具体的实现的，这可怎么办呢？抽象部分如何才能调用到具体实现部分的功能呢？很简单，搭个桥不就可以了，搭个桥，让抽象部分通过这个桥就可以调用到实现部分的功能了，因此需要桥接。

### 3: 如何桥接

这个理解上也很简单，只要让抽象部分拥有实现部分的接口对象，这就桥接上了，在抽象部分就可以通过这个接口来调用具体实现部分的功能。也就是说，桥接在程序上就体现成了在抽象部分拥有实现部分的接口对象，维护桥接就是维护这个关系。

## 理解桥接模式

### 4：独立变化

桥接模式的意图：使得抽象和实现可以独立变化，都可以分别扩充。也就是说抽象部分和实现部分是一种非常松散的关系，从某个角度来讲，抽象部分和实现部分是可以完全分开的，独立的，抽象部分不过是一个使用实现部分对外接口的程序罢了。

如果这么看桥接模式的话，就类似于策略模式了，抽象部分需要根据某个策略，来选择真实的实现，也就是说桥接模式的抽象部分相当于策略模式的上下文。更原始的就直接类似于面向接口编程，通过接口分离的两个部分而已。但是别忘了，桥接模式的抽象部分，是可以继续扩展和变化的，而策略模式只有上下文，是不存在所谓抽象部分的。

那抽象和实现为何还要组合在一起呢？原因是在抽象部分和实现部分还是存在内部联系的，抽象部分的实现通常是需要调用实现部分的功能来实现的



## 理解桥接模式

### 5: 动态变换功能

由于桥接模式中的抽象部分和实现部分是完全分离的，因此可以在运行时动态组合具体的真实实现，从而达到动态变换功能的目的。

从另外一个角度看，抽象部分和实现部分没有固定的绑定关系了，因此同一个真实实现可以被不同的抽象对象使用，反过来，同一个抽象也可以有多个不同的实现。就像前面示例的那样，比如：站内短消息的实现功能，可以被普通消息、加急消息或是特急消息等不同的消息对象使用；反过来，某个消息具体的发送方式，可以是站内短消息，或者是Email，也可以是手机短消息等具体的发送方式。

### 6: 退化的桥接模式

如果Implementor仅有一个实现，那么就没有必要创建Implementor接口了，这是一种桥接模式退化的情况。这个时候Abstraction和Implementor是一一对应的关系，虽然如此，也还是要保持它们的分离状态，这样的话，它们才不会相互影响，才可以分别扩展。

也就是说，就算不要Implementor接口了，也要保持Abstraction和Implementor是分离的，模式的分离机制仍然是非常有用的。

**做最好的在线学习社区**

网 址: <http://sishuok.com>

咨询QQ: 2371651507



## 理解桥接模式

### 7: 桥接模式和继承

继承是扩展对象功能的一种常见手段，通常情况下，继承扩展的功能变化纬度都是一纬的，也就是变化的因素只有一类。

对于出现变化因素有两类的，也就是有两个变化纬度的情况，继承实现就会比较痛苦。比如上面的示例，就有两个变化纬度，一个是消息的类别，不同的消息类别处理不同；另外一个消息的发送方式。

从理论上来说，如果用继承的方式来实现这种有两个变化纬度的情况，最后实际的实现类应该是两个纬度上可变数量的乘积那么多个。比如上面的示例，在消息类别的纬度上，目前的可变数量是3个，普通消息、加急消息和特急消息；在消息发送方式的纬度上，目前的可变数量也是3个，站内短消息、Email和手机短消息。这种情况下，如果要实现全的话，那么需要的实现类应该是： $3 \times 3 = 9$ 个。

## 理解桥接模式

如果要在任何一个纬度上进行扩展，都需要实现另外一个纬度上的可变数量那么多个实现类，这也是为何会感到扩展起来很困难。而且随着程序规模的加大，会越来越难以扩展和维护。

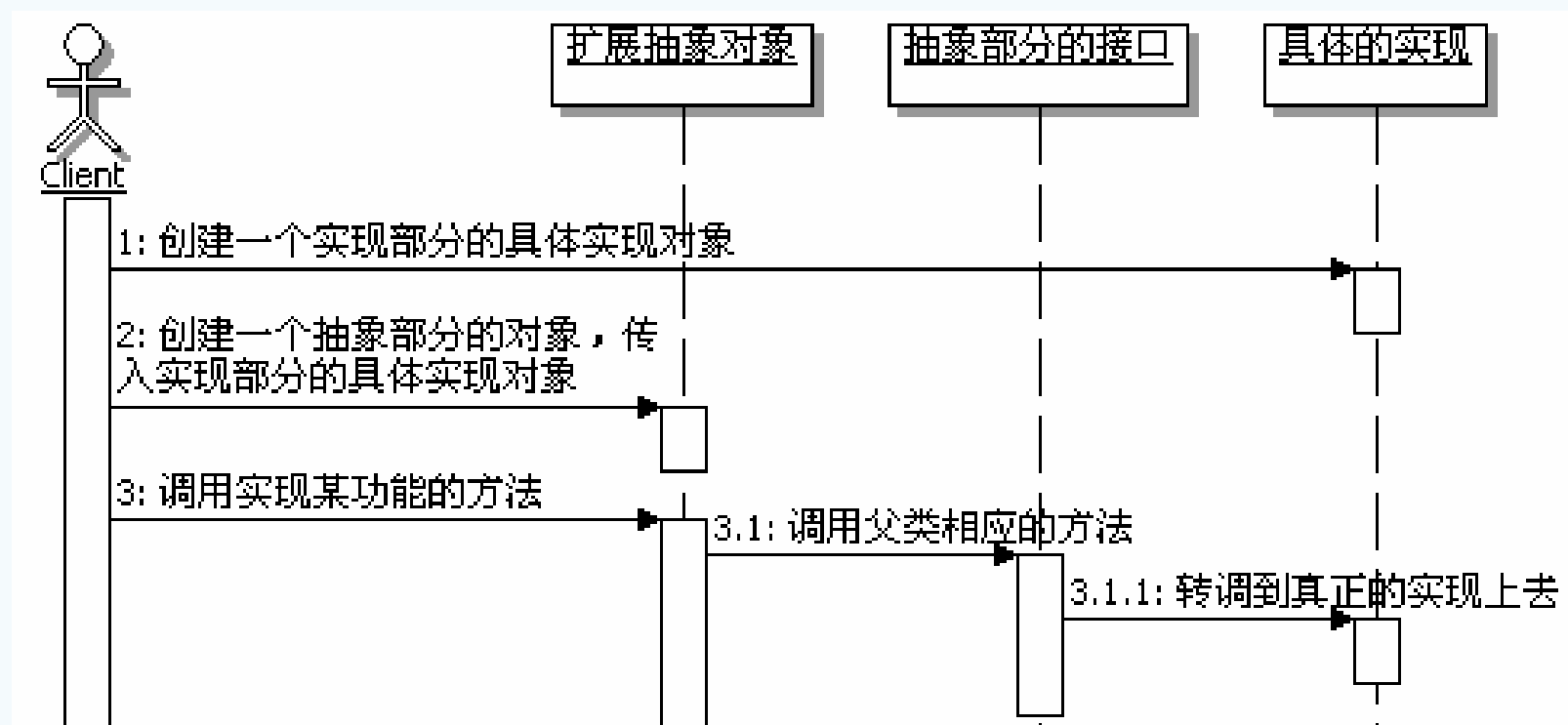
而桥接模式就是用来解决这种有两个变化纬度的情况下，如何灵活的扩展功能的一个很好的方案。其实，桥接模式主要是把继承改成了使用对象组合，从而把两个纬度分开，让每一个纬度单独去变化，最后通过对象组合的方式，把两个纬度组合起来，每一种组合的方式就相当于原来继承中的一种实现，这样就有效的减少了实际实现的类的个数。

从理论上来说，如果用桥接模式的方式来实现这种有两个变化纬度的情况，最后实际的实现类应该是两个纬度上可变数量的和那么多个。同样是上面那个示例，使用桥接模式来实现，实现全的话，最后需要的实现类的数目应该是：  
 $3 + 3 = 6$ 个。

这也从侧面体现了，使用对象组合的方式比继承要来得更灵活。

## 理解桥接模式

### 8: 桥接模式的调用顺序示意图



## 理解桥接模式

### n 谁来桥接

所谓谁来桥接，就是谁来负责创建抽象部分和实现部分的关系，说得更直白点，就是谁来负责创建Implementor的对象，并把它设置到抽象部分的对象里面去，这点对于使用桥接模式来说，是十分重要的一点。

大致有如下几种实现方式：

- 1: 由客户端负责创建Implementor的对象，并在创建抽象部分的对象的时候，把它设置到抽象部分的对象里面去，前面的示例采用的就是这个方式
- 2: 可以在抽象部分的对象构建的时候，由抽象部分的对象自己来创建相应的Implementor的对象，当然可以给它传递一些参数，它可以根据参数来选择并创建具体的Implementor的对象
- 3: 可以在Abstraction中选择并创建一个缺省的Implementor的对象，然后子类可以根据需要改变这个实现
- 4: 也可以使用抽象工厂或者简单工厂来选择并创建具体的Implementor的对象，抽象部分的类可以通过调用工厂的方法来获取Implementor的对象
- 5: 如果使用IoC/DI容器的话，还可以通过IoC/DI容器来创建具体的Implementor的对象，并注入回到Abstraction中

下面分别给出后面几种实现方式的示例：

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

## 理解桥接模式

1: 由抽象部分的对象自己来创建相应的Implementor的对象

分成两种情况，一种是需要外部传入参数，一种是不需要外部传入参数。

2: 在Abstraction中创建缺省的Implementor对象

对于这种方式，实现比较简单，直接在Abstraction的构造方法中，创建一个缺省的Implementor对象，然后子类根据需要，看是直接使用还是覆盖掉。

3: 使用抽象工厂或者是简单工厂

对于这种方式，根据具体需要来选择，如果是想要创建一系列实现对象，那就使用抽象工厂，如果是创建单个的实现对象，那就使用简单工厂。

这种方法的优点是Abstraction类不用和任何一个Implementor类直接耦合。

4: 使用IoC/DI的方式

对于这种方式，Abstraction的实现就更简单了，只需要实现注入Implementor对象的方法就可以了，其它的Abstraction就不管了。

IoC/DI容器会负责创建Implementor对象，并设置回到Abstraction对象中，使用IoC/DI的方式，并不会改变Abstraction和Implementor的关系，Abstraction同样需要持有相应的Implementor对象，同样会把功能委托给Implementor对象去实现。

做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

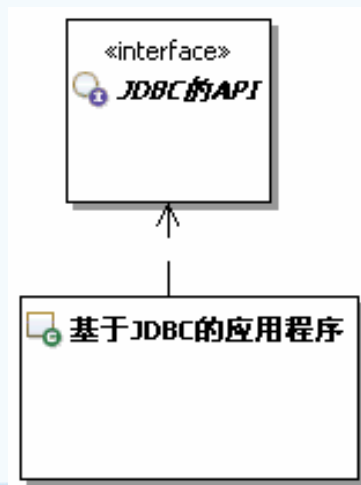


## 理解桥接模式

### n 典型例子-JDBC

在Java应用中，对于桥接模式有一个非常典型的例子，就是：应用程序使用JDBC驱动程序进行开发的方式。所谓驱动程序，指的是按照预先约定好的接口来操作计算机系统或者是外围设备的程序。

我们写的应用程序，是面向JDBC的API在开发，这些接口就相当于桥接模式中的抽象部分的接口。那么怎样得到这些API的呢？是通过DriverManager来得到的。此时的系统结构如图



做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

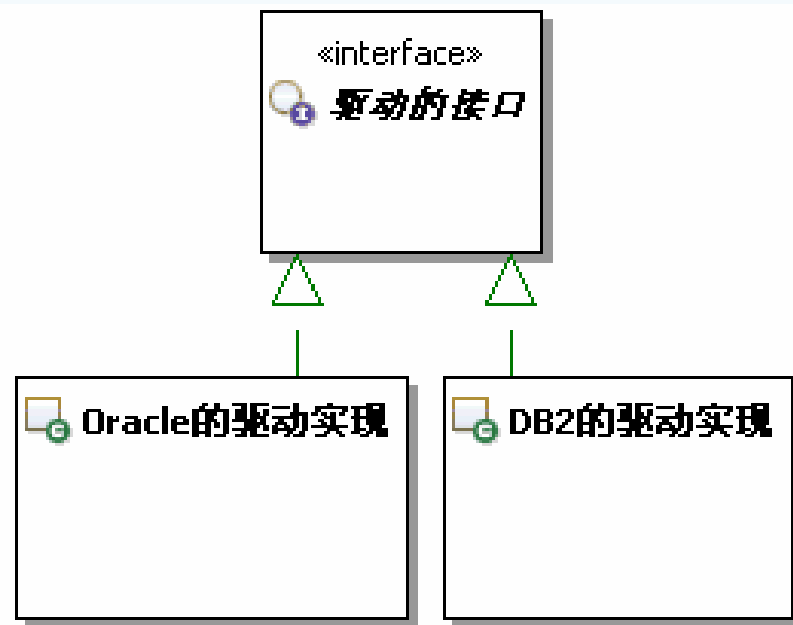
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 理解桥接模式

那么这些JDBC的API，谁去实现呢？光有接口，没有实现也不行啊。

该驱动程序登场了，JDBC的驱动程序实现了JDBC的API，驱动程序就相当于桥接模式中的具体实现部分。而且不同的数据库，由于数据库实现不一样，可执行的Sql也不完全一样，因此对于JDBC驱动的实现也是不一样的，也就是不同的数据库会有不同的驱动实现。此时驱动程序这边的程序结构如图



做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

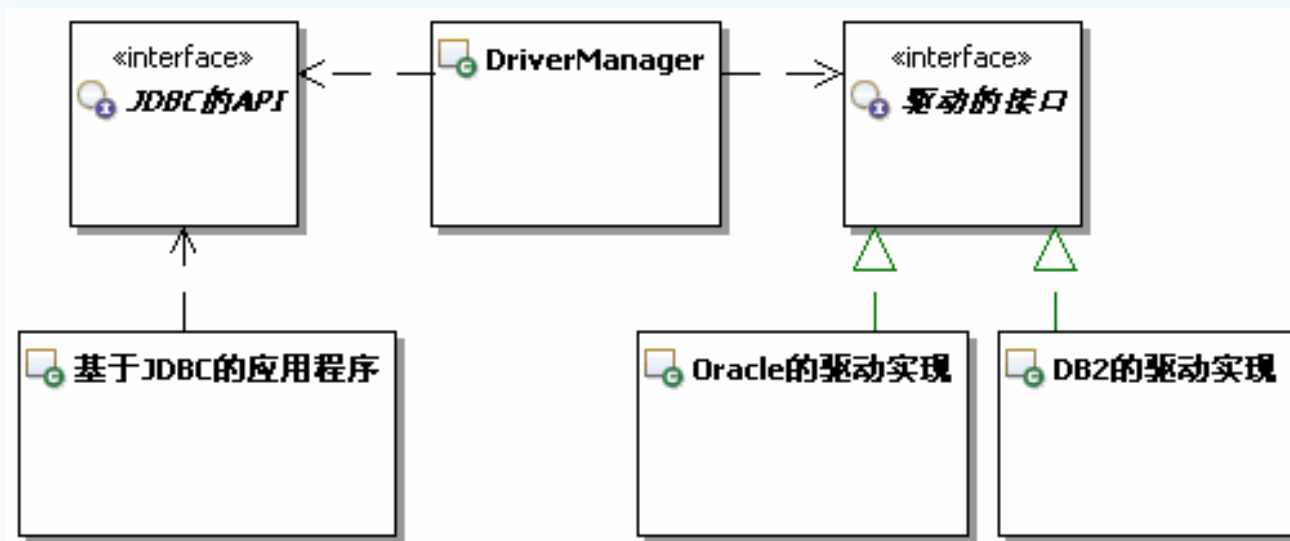
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解桥接模式

有了抽象部分——JDBC的API，有了具体实现部分——驱动程序，那么它们如何连接起来呢？就是如何桥接呢？

就是前面提到的DriverManager来把它们桥接起来，从某个侧面来看，DriverManager在这里起到了类似于简单工厂的功能，基于JDBC的应用程序需要使用JDBC的API，如何得到呢？就通过DriverManager来获取相应的对象。

那么此时系统的整体结构如图



做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解桥接模式

通过上图可以看出，基于JDBC的应用程序，使用JDBC的API，相当于是对数据库操作的抽象的扩展，算作桥接模式的抽象部分；而具体的接口实现是由驱动来完成的，驱动这边自然就相当于桥接模式的实现部分了。而桥接的方式，不再是让抽象部分持有实现部分，而是采用了类似于工厂的做法，通过DriverManager来把抽象部分和实现部分对接起来，从而实现抽象部分和实现部分解耦。

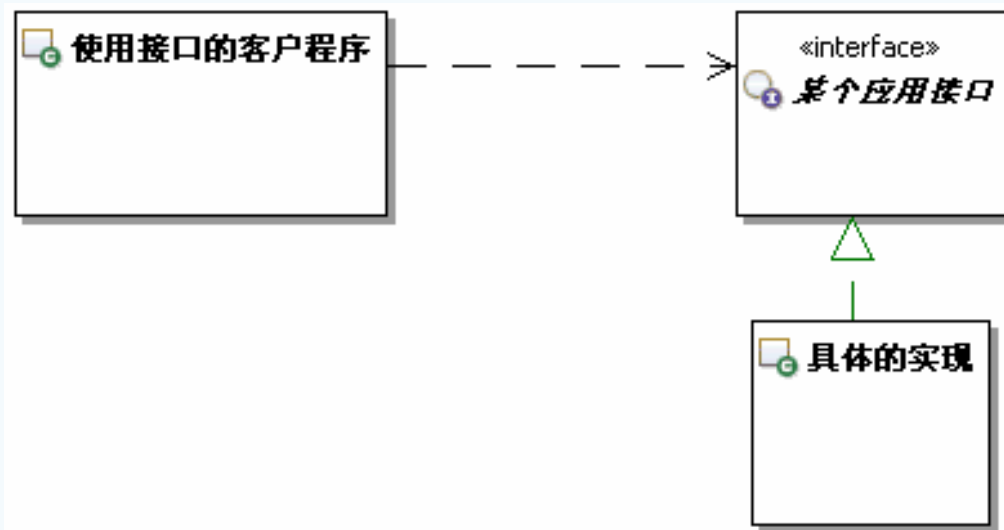
JDBC的这种架构，把抽象和具体分离开来，从而使得抽象和具体部分都可以独立扩展。对于应用程序而言，只要选用不同的驱动，就可以让程序操作不同的数据库，而无需更改应用程序，从而实现在不同的数据库上移植；对于驱动程序而言，为数据库实现不同的驱动程序，并不会影响应用程序。而且，JDBC的这种架构，还合理的划分了应用程序开发人员和驱动程序开发人员的边界。

## 理解桥接模式

### n 广义桥接-Java中无处不桥接

使用Java编写程序，一个很重要的原则就是“面向接口编程”，说得准确点应该是“面向抽象编程”，由于在Java开发中，更多的使用接口而非抽象类，因此通常就说成“面向接口编程”了。

接口把具体的实现和使用接口的客户程序分离开来，从而使得具体的实现和使用接口的客户程序可以分别扩展，而不会相互影响。结构如图



做最好的在线学习社区

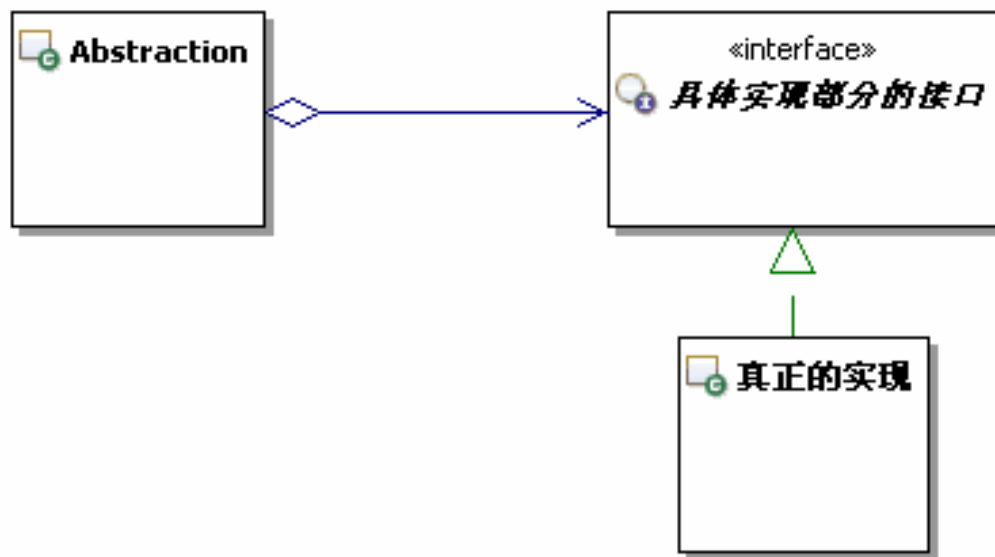
网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解桥接模式

可能有些朋友会觉得，听起来怎么像是桥接模式的功能呢？没错，如果把桥接模式的抽象部分先稍稍简化一下，暂时不要RefinedAbstraction部分，那么就跟上面的结构图差不多了。去掉RefinedAbstraction后的简化的桥接模式结构示意图如图



## 理解桥接模式

是不是差不多呢？有朋友可能会觉得还是有很大差异，差异主要在：前面接口的客户程序是直接使用接口对象，不像桥接模式的抽象部分那样，是持有具体实现部分的接口，这就导致画出来的结构图，一个是依赖，一个是聚合关联。

请思考它们的本质功能，桥接模式中的抽象部分持有具体实现部分的接口，最终目的是什么，还不是需要通过调用具体实现部分的接口中的方法，来完成一定的功能，这跟直接使用接口没有什么不同，只是表现形式有点不一样。再说，前面那个使用接口的客户程序也可以持有相应的接口对象，这样从形式上就一样了。

也就是说，从某个角度来讲，桥接模式不过就是对“面向抽象编程”这个设计原则的扩展。正是通过具体实现的接口，把抽象部分和具体的实现分离开来，抽象部分相当于是使用实现部分接口的客户程序，这样抽象部分和实现部分就松散耦合了，从而可以实现相互独立的变化。

这样一来，几乎可以把所有面向抽象编写的程序，都视作是桥接模式的体现，至少算是简化的桥接模式，就算是广义的桥接吧。而Java编程很强调“面向抽象编程”，因此，广义的桥接，在Java中可以说是无处不在。

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

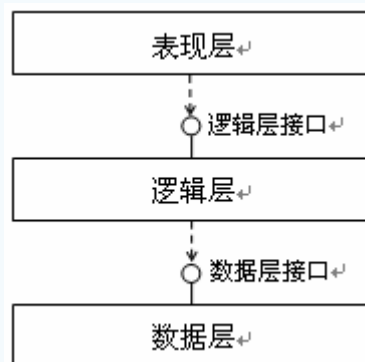
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送



## 理解桥接模式

再举个大家最熟悉的例子来示例一下。在Java应用开发中，分层实现算是最基本的设计方式了吧，就拿大家最熟的三层架构来说，表现层、逻辑层和数据层，或许有些朋友对它们称呼的名称不同，但都是同一回事情。

三层的基本关系是表现层调用逻辑层，逻辑层调用数据层，通过什么来进行调用呢？当然是接口了，它们的基本结构如图



通过接口来进行调用，使得表现层和逻辑层分离开来，也就是说表现层的变化，不会影响到逻辑层，同理逻辑层的变化不会影响到表现层。这也是同一套逻辑层和数据层，就能够同时支持不同的表现层实现的原因，比如支持Swing或Web方式的表现层。

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

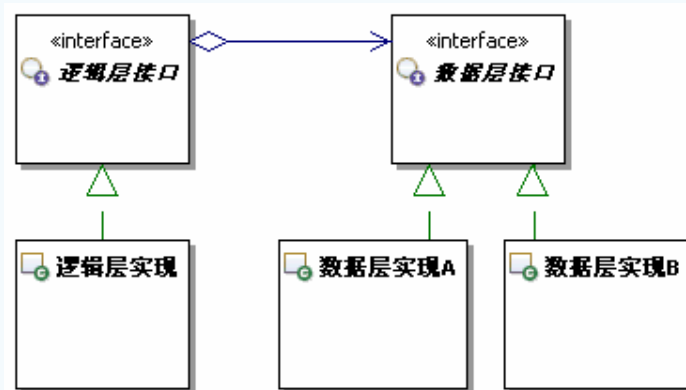
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解桥接模式

在逻辑层和数据层之间也是通过接口来调用，同样使得逻辑层和数据层分离开，使得它们可以独立的扩展。尤其是数据层，可能会有很多的实现方式，比如：数据库实现、文件实现等，就算是数据库实现，又有针对不同数据库的实现，如Oracle、DB2等等。

总之，通过面向抽象编程，三层架构的各层都能够独立的扩展和变化，而不会对其它层次产生影响。这正好是桥接模式的功能，实现抽象和实现的分离，从而使得它们可以独立的变化。当然三层架构不只是一个地方使用桥接模式，而是至少在两个地方来使用了桥接模式，一个在表现层和逻辑层之间，一个在逻辑层和数据层之间。

下面先分别看看这两个使用桥接模式的地方的程序结构，然后再综合起来看看整体的程序结构。先看看逻辑层和数据层之间的程序结构，如图



做最好的在线学习社区

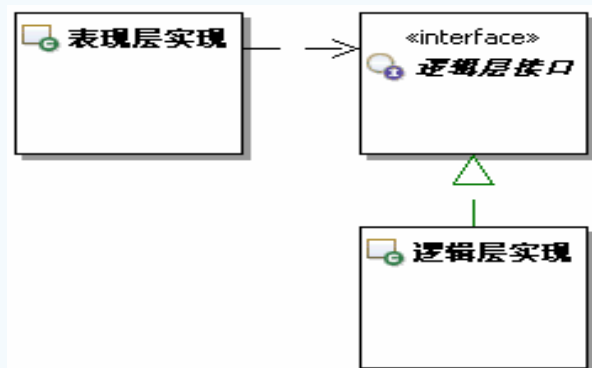
网 址: <http://sishuok.com>

咨询QQ: 2371651507

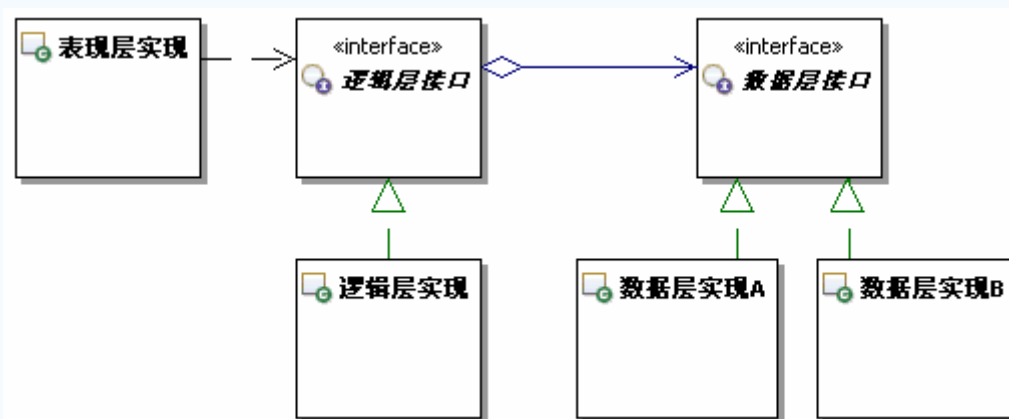
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解桥接模式

再看看表现层和逻辑层之间的结构示意图，如图



然后再把它们结合起来，看看结合后的程序结构



做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解桥接模式

从广义桥接模式的角度来看，平日熟悉的三层架构其实就是在组合使用桥接模式。从这个图还可以看出，桥接模式是可以连续组合使用的，一个桥接模式的实现部分，可以作为下一个桥接模式的抽象部分。如此类推，可以从三层架构扩展到四层、五层、直到N层架构，都可以使用桥接模式来组合。

如果从更本质的角度来看，基本上只要是面向抽象编写的Java程序，都可以视为是桥接模式的应用，都是让抽象和实现相分离，从而使它们能独立的变化。不过只要分离的目的达到了，叫不叫桥接模式就无所谓了。

做最好的在线学习社区

网 址：<http://sishuok.com>

咨询QQ：2371651507

私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费在线学习视频独家大放送

## 理解桥接模式

- n 桥接模式的优缺点
  - 1: 分离抽象和实现部分
  - 2: 更好的扩展性
  - 3: 可动态切换实现
  - 4: 可减少子类的个数



## 思考桥接模式

### n 桥接模式的本质

桥接模式的本质是：分离抽象和实现

### n 桥接模式对设计原则的体现

桥接模式很好的实现了开闭原则：通常应用桥接模式的地方，抽象部分和实现部分都是可变化的，也就是应用会有两个变化纬度，桥接模式就是找到这两个变化，并分别封装起来，从而合理的实现OCP。

桥接模式还很好的体现了：多用对象组合，少用对象继承：

在前面的示例中，如果使用对象继承来扩展功能，不但让对象之间有很强的耦合性，而且会需要很多的子类才能完成相应的功能，前面已经讲述过了，需要两个纬度上的可变化数量的乘积个子类。

采用对象的组合，松散了对象之间的耦合性，不但使每个对象变得简单和可维护，还大大减少了子类的个数，根据前面的讲述，大约需要两个纬度上的可变化数量的和这么多个子类。

**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

## 思考桥接模式

### n 何时选用桥接模式

- 1: 如果你不希望在抽象和实现部分采用固定的绑定关系，可以采用桥接模式，来把抽象和实现部分分开，然后在程序运行期间来动态的设置抽象部分需要用到的具体的实现，还可以动态切换具体的实现
- 2: 如果出现抽象部分和实现部分都应该可以扩展的情况，可以采用桥接模式，让抽象部分和实现部分可以独立的变化，从而可以灵活的进行单独扩展，而不是搅在一起，扩展一边会影响到另一边。
- 3: 如果希望实现部分的修改，不会对客户产生影响，可以采用桥接模式，客户是面向抽象的接口在运行，实现部分的修改，可以独立于抽象部分，也就不会对客户产生影响了，也可以说对客户是透明的
- 4: 如果采用继承的实现方案，会导致产生很多子类，对于这种情况，可以考虑采用桥接模式，分析功能变化的原因，看看是否能分离成不同的纬度，然后通过桥接模式来分离它们，从而减少子类的数目