

Java设计模式之Facade模式计算机等级考试 PDF转换可能丢失  
图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/646/2021\\_2022\\_Java\\_E8\\_AE\\_BE\\_E8\\_AE\\_A1\\_c97\\_646099.htm](https://www.100test.com/kao_ti2020/646/2021_2022_Java_E8_AE_BE_E8_AE_A1_c97_646099.htm) GOF 《设计模式》一书

对Facade模式是这样描述的: 为子系统中的一组接口提供一个统一接口。Facade模式定义了一个更高层的接口，使子系统更加容易使用。大致意思是说：使用一种比原有方式更简单的办法与系统交互。

例如，我们把一个很重要的文件，放在了第二抽屉里，而第二个抽屉的钥匙放在了第一个抽屉里，我们要想取出这个文件，第一步肯定要拿到第一个抽屉的钥匙，然后打开它再拿出第二个抽屉的钥匙，最后打开第二个抽屉取出文件。我就上面说的那个情形写一下实现代码，首先我们要实现二个子系统,呵呵，把抽屉比喻成系统，有点夸张了（DrawerOne、DrawerTwo）：

```
class DrawerOne { public void open(){ System.out.println("第一个抽屉被打开了").  
getKey(). } public void getKey(){ System.out.println("得到第二个  
抽屉的钥匙"). } } class DrawerTwo{ public void open(){  
System.out.println("第二个抽屉被打开了"). getFile(). } public  
void getFile(){ System.out.println("得到这个重要文件"). } } public  
class Client{ public static void main(String []args){ DrawerOne  
darwerOne=new DrawerOne(). DrawerTwo darwerTwo=new  
DrawerTwo(). darwerOne.open(). darwerTwo.open(). } } 由于没有使用Fa?ade模式，可以看到要想得到这个文件要首先打开第一个抽屉，然后再打开第二个抽屉，在我们实际所开发的系统中，有时候客户要实现某一操作，并不需要知道实现这一操作的详细步骤，而是简单地点击某一个按钮就可以得到自
```

己想要的结果。下面对上面的代码使用Facade模式进行改进，建立一个FacadeDrawer类：100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)