


设计模式 适配器模式 以手机充电器为例

原创 鸿洋_ 于 2014-05-14 21:59:35 发布 21248 收藏 17 版权

分类专栏: 【Java 设计模式】 设计模式融入生活 文章标签: 设计模式 适配器模式

 【Java 设计模式】 同时刊

67 订阅 10 篇文章 订阅专栏

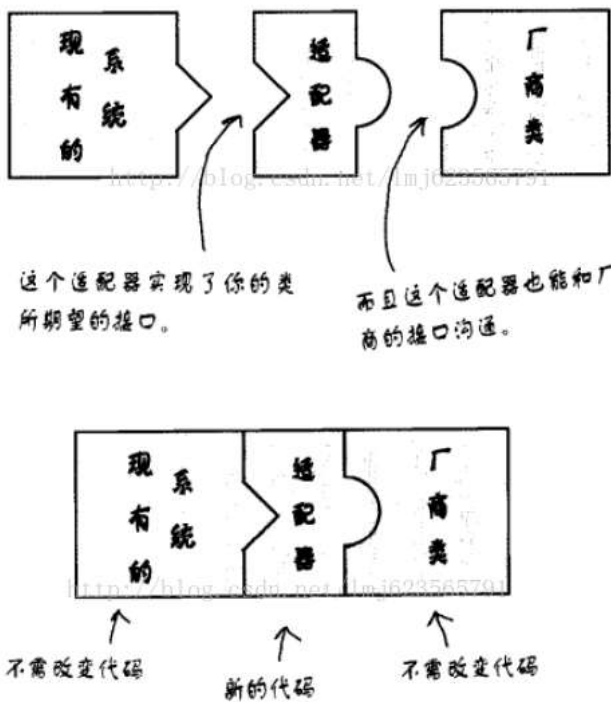
转载请标明出处: <http://blog.csdn.net/Imj623565791/article/details/25833393>

坚持，坚持~

继续 **设计模式**，尼玛停了段时间就不想写了，一定要坚持~今天带来适配器模式

老样子，定义：将一个类的接口转换成客户期望的另一个接口，适配器让原本接口不兼容的类可以相互合作。这个定义还好，说适配器的功能就是把一个接口转成另一个接口。

发现两张图片可以很好的解释 **适配器模式**：



这两张图很好的说明了适配器的作用哈，话说我当年买了个欧版的HTC G17，还配了个插头转化器，这个插头转化器就是起得适配器的作用。下来来点代码解释哈，如题目，手机充电器一般都是5V左右吧，咱天朝的家用交流电压220V，所以手机充电需要一个适配器（降压器），有什么物理名词使用错误的，见谅。

首先一部手机：Mobile.java

```
1 package com.zhy.pattern.adapter;
2
3 public class Mobile
4 {
5     /**
6      * 充电
7      * @param power
8      */
9     public void inputPower(V5Power power)
10    {
11        int provideV5Power = power.provideV5Power();
```

```
12 |         System.out.println("手机（客户端）：我需要5V电压充电，现在是-->");
    |
    | 14 | }
```

可以看出，手机依赖一个提供5V电压的接口：

```
1 | package com.zhy.pattern.adapter;
2 | /**
3 |  * 提供5V电压的一个接口
4 |  * @author zhy
5 |  *
6 |  */
7 | public interface V5Power
8 | {
9 |     public int provideV5Power();
10 | }
```

然后我们拥有的是220V家用交流电：

```
1 | package com.zhy.pattern.adapter;
2 |
3 | /**
4 |  * 家用220V交流电
5 |  * @author zhy
6 |  *
7 |  */
8 | public class V220Power
9 | {
10 |     /**
11 |      * 提供220V电压
12 |      * @return
13 |      */
14 |     public int provideV220Power()
15 |     {
16 |         System.out.println("我提供220V交流电压。");
17 |         return 220 ;
18 |     }
19 | }
```

下面我们需要一个适配器，完成220V转5V的作用：

```
1 | package com.zhy.pattern.adapter;
2 |
3 | /**
4 |  * 适配器，把220V电压变成5V
5 |  * @author zhy
6 |  *
7 |  */
8 | public class V5PowerAdapter implements V5Power
9 | {
10 |     /**
11 |      * 组合的方式
12 |      */
13 |     private V220Power v220Power ;
14 |
15 |     public V5PowerAdapter(V220Power v220Power)
16 |     {
17 |         this.v220Power = v220Power ;
18 |     }
19 |
20 |     @Override
```

```
21 | public int provideV5Power()22 | {
23 |     int power = v220Power.provideV220Power() ;
24 |     //power经过各种操作-->5
25 |     System.out.println("适配器：我悄悄的适配了电压。");
26 |     return 5 ;
27 | }
28 |
29 | }
```

最后测试，我们给手机冲个电：

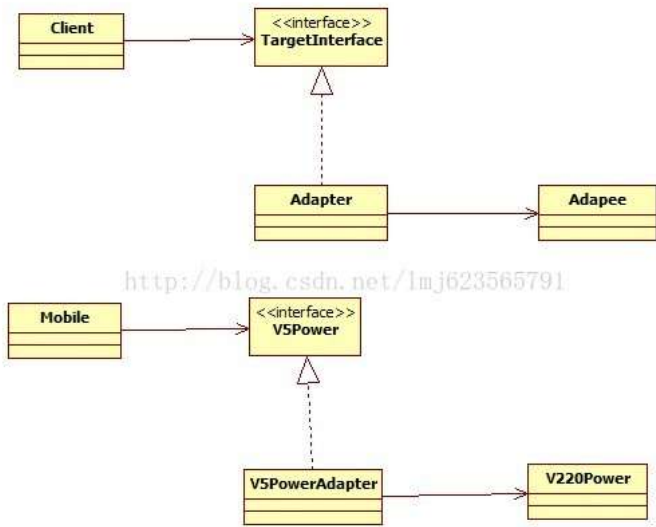
```
1 | package com.zhy.pattern.adapter;
2 |
3 | public class Test
4 | {
5 |     public static void main(String[] args)
6 |     {
7 |         Mobile mobile = new Mobile();
8 |
9 |         V5Power v5Power = new V5PowerAdapter(new V220Power()) ;
10 |        mobile.inputPower(v5Power);10 |    }
11 | }
```

输出：

- 1 | 现有类：我提供220V交流电压。
- 2 | 适配器：我悄悄的适配了电压。
- 3 | 手机（客户端）：我需要5V电压充电，现在是-->5V

可以看出，我们使用一个适配器完成了把220V转化了5V然后提供给手机使用，且我们使用了组合（OO设计原则），原有的手机，以及200V电压类都不需要变化，且手机（客户端）和220V（被适配者）完全解耦。

最后画个uml类图，便于大家理解：



上面是适配器的类图，下面是我们的例子的类图，咋样，还不错吧。没事画个图也不错，不然软件都白装了。。。。

最后，恭喜你，你又学会了一个设计模式，适配器模式。持续求赞，求留言~

显示推荐内容

觉得还不错? [一键收藏](#)

鸿洋_

关注

134

17

19

专栏目录