

简述
模式中角色
示例说明
类适配器模式代码实现
对象适配器模式代码实现
总结
优点
缺点

简述

此模式的作用就是兼容原本接口不匹配的两个类，起到桥梁的作用

模式中角色

- Target：目标接口，也就是期望得到的接口
- Adaptee：需要适配的接口
- Adapter：适配器，负责把Adaptee转换成Target的类

示例说明

手机充电头，其实就是一个电源适配器，生活用电的电压是220v，而手机充电的电压是5v，所以这时候就需要使用到电源

类适配器模式代码实现

```
1 //Target接口
2 public interface Volt5 {
3     public int getVolt5();
4 }
5
6 //Adaptee类
7 public class Volt220 {
8     public int getVlot220(){
9         return 220;
10    }
11 }
12
13 //Adapter类
14 public class VoltAdapter extends Volt220 implements Volt5 {
15     @Override
16     public int getVolt5() {
17         return 5;
18     }
19 }
20
```

```

21 public class Client{
22     public static void main(String[] args){
23         VoltAdapter adapter = new VoltAdapter();
24         System.out.println("输出电压: " + adapter.getVolt5());
25     }
26 }

```

对象适配器模式代码实现

与类适配器模式不同的是，对象适配器模式不适用继承关系链接Adaptee类，而是使用代理关系链接到Adaptee类

```

1 //Target接口
2 public interface Volt5 {
3     public int getVolt5();
4 }
5
6 //Adaptee类
7 public class Volt220 {
8     public int getVlot220(){
9         return 220;
10    }
11 }
12
13 //Adapter类
14 public class VoltAdapter implements Volt5 {
15
16     Volt220 volt220;
17
18     public VoltAdapter(Volt220 volt220){
19         this.volt220 = volt220;
20     }
21
22     public int getVolt220(){
23         return volt220.getVlot220();
24     }
25
26     @Override
27     public int getVolt5() {
28         return 5;
29     }
30 }
31
32 public class Client{
33     public static void main(String[] args){
34         VoltAdapter adapter = new VoltAdapter(new Volt220());
35         System.out.println("输出电压: " + adapter.getVolt5());
36     }
37 }

```

总结

优点

- 复用性好，当现有的类不符合需求时，通过适配器就可以使这些功能得到更好地复用。
- 扩展性好，可以更好地扩展系统的功能

缺点

过多的使用，会让代码更加凌乱