链滴

GOF 设计模式小白教程之适配器模式

作者：valarchie

原文链接：https://ld246.com/article/1567003872778

来源网站：链滴

适配器模式（Adapter）

定义：

将一个类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类一起工作。

通俗解释：

有一个中国人和一个日本人，他们都只会母语并且不想再学习其他语言（意味着不修改两个不兼容的的代码）。但是他们之间又想进行交流，那怎么办呢？这时候就可以有一个翻译可以中国人日本人不学习新的语言，而他们两个人之间可以互相交流。这里的中国人和日本人是互不兼容的两个接口，而译就是他们之间的适配器。

代码：

中国人类型，只听得懂 中文使用者（目标接口）讲的话

```
public class Chinese {
    // 中国人只听得懂 中文使用者  所说的话
    public void listenChinese(ChineseSpeaker chineseSpeaker) {
        System.out.println(chineseSpeaker.sayChinese());
    }

}
```

中文使用者接口：中国人类型期望使用的接口

```
public interface ChineseSpeaker {
    String sayChinese();
}
```

日本人类型（被适配者），只会说日文

```
public class Japanese {

    public String sayJapanese() {
        return "konichiwa";
    }

}
```

翻译类型（适配器）实现了中文使用者接口，中国人类型就可以使用这个对象了。而这个翻译对象持日本人对象，并对日本人的日文进行了转换

```
public class Translator implements ChineseSpeaker{

    private Japanese japanese;

    public Translator(Japanese japanese) {
        this.japanese = japanese;
    }
```

原文链接：GOF 设计模式小白教程之适配器模式

```
    public String sayChinese() {

        if (japanese.sayJapanese().equals("konichiwa")) {
            return "你好！ ";
        } else {
            return "听不懂";
        }

    }

}
```

测试适配器

```
public class TestAdapter {

    public static void main(String[] args) {
        // 将日本人对象填入翻译对象
        Translator translator = new Translator(new Japanese());

        Chinese chinese = new Chinese();
        // 中国人不直接与日本人交流 而是与翻译
        chinese.listenChinese(translator);

    }

}
```

解析：

1. 客户端通过适配器可以透明地调用目标接口。
2. 复用了现存的类，程序员不需要修改原有代码而重用现有的适配者类。
3. 将目标类和适配者类解耦，解决了目标类和适配者类接口不一致的问题。