

# 17 适配器模式

适配器模式目的是解决两个软件实体间的接口不兼容问题。让其可以一起工作

别名是包装器Wrapper，这是一个相对简单的模式。也有很多程序实际场景，例如发现新的模块或者接口不符合需求，我们可以尝试改接口实现，但是如果模块很复杂或者这是别人的库就不行了。另一种就是创建一个适配器来将原接口转换位符合需求的，用户和适配器打交道就行了

## 现实中的适配器

- 港式插头转换器
- 电源适配器（电脑，笔记本）
- USB转接口

## 应用

- 一种亡羊补牢的模式，使用适配器将旧接口包装成一个新的接口，让其继续保持生命力，例如之前接口返回XML格式数据，我们可以创建一个XML-JSON适配器来解决
- 此外例如我们希望实现一个通用地图SDK，加入百度和谷歌地图的展示方法一个叫show，一个叫display的话怎么办呢？这里我们可以新增GoogleMapAdapter并在里面定义show方法，具体调用googleMap.display来完成适配
- 也有一些数据我们需要从多个数据源聚合，但是发现有一些是对象数组，有一些是对象KV，这个时候可以新增一个格式转换适配器来避免改动原先代码

## 小结

这种模式相对简单，而且之前的装饰者，代理和外观模式其实都比较类似。都是一种“包装模式”，区别主要在于模式的意图：

- 适配器模式目的是解决接口不匹配问题，不考虑接口具体实现以及未来如何演化。
- 装饰者和代理模式主要是给原来对象增加额外功能，代理模式则是控制对对象的访问
- 外观模式作用倒是和适配器比较类似，可以把外观模式看成一组对象的适配器。