## 第十章 桥接模式

1. **（①B）设计模式将抽象部分与它的实现部分分离，使它们都可以独立变化。下图所示为改设计模式的类图，其中（②D）用于定义实现部分的接口。**

****

① A、Singleton B、Bridge C、Composite D、Façade

② A、Abstrarction B、ConcreteImplementorA

C、ConcreteImplementorB D、Implementor

1. **以下关于桥接模式的叙述错误的是（C）**
   1. 桥接模式的用意是将抽象化与实现化脱耦，使得两者可以独立变化
   2. 桥接模式将继承关系转换为关联关系，从而降低系统的耦合度
   3. 桥接模式可以动态的给一个对象增加功能，这些功能也可以动态的撤销
   4. 桥接模式可以从接口中分离实现功能，使得设计更具扩展性
2. **( C)不是桥接模式所使用的的场景。**
   1. 一个可以跨平台并支持多种格式的文件编辑器
   2. 一个支持多数据源的报表生成工具，可以用不同的图形方式显示报表信息
   3. 一个可动态选择排序算法的数据操作工具
   4. 一个支持多种编程语言的跨平台开发工具
3. 如果系统中存在两个以上的变化维度，是否可以使用桥接模式进行处理？如果可以，系统该如何设计？

桥接模式可以处理存在多个独立变化维度的系统，每一个独立维度对应一个继承结构，其中一个为“抽象类”层次结构，其他为“实现类”层次结构，“抽象类”层次结构中的抽象类与“实现类”层次结构中的接口之间存在抽象耦合关系。