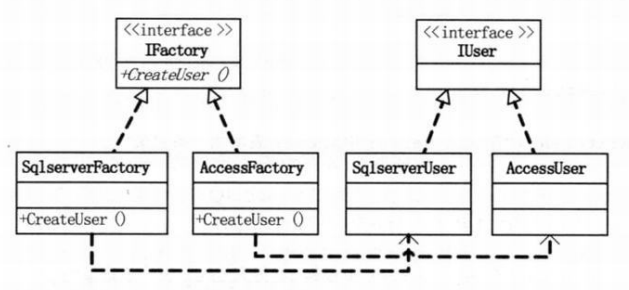
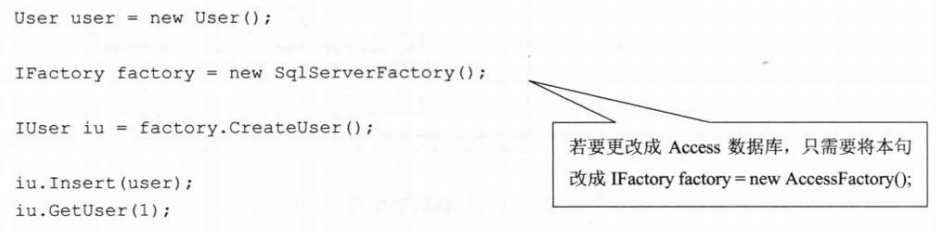
问题：数据库访问表，但是数据库可能发生变化，如sql、access、oracle。

分析：

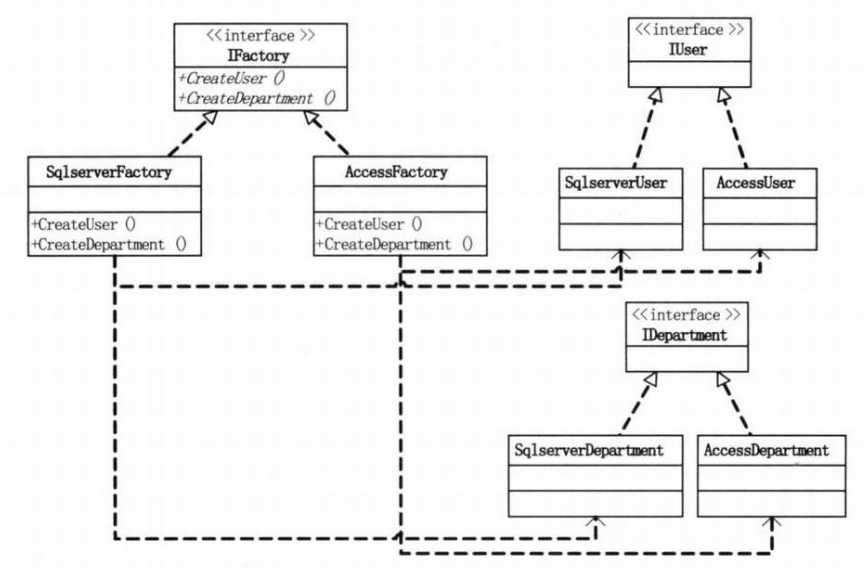
1. 包含User类，对User控制的sql类，在客户端使用时，初始化这个sql类来对User进行操作。在只包含一个User表时，可以利用工厂模式：



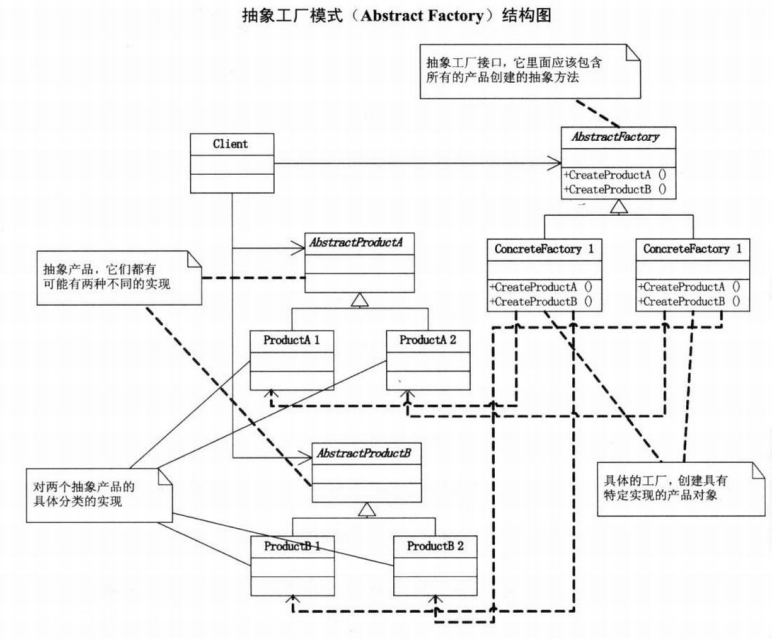
那么在数据库切换时：



但是，如果在加个Department表：



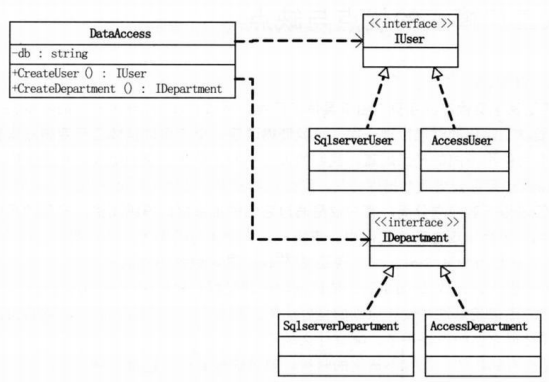
1. 抽象工厂模式：



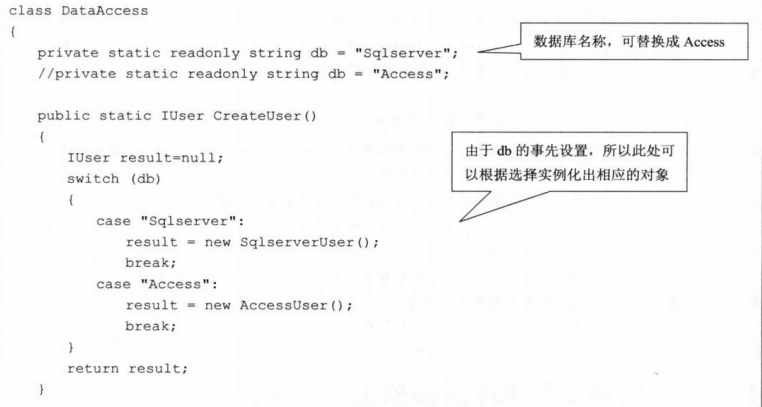
ConcreteFactory相当于SqlseverFactory，而ProductA相当于SqlserverUser。

通常在运行时刻创建一个ConcreteFactory类的实例，这个具体的工厂再创建特定实现的产品对象。客户端只需指定不同的具体工厂就可以了。好处：1）便于交换产品系列 2）具体的创建过程与客户端分离。

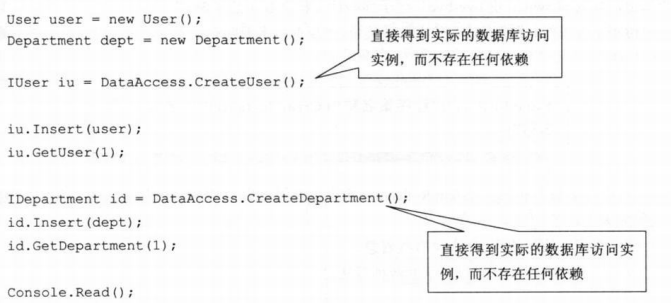
1. 如果再增加Oracle，那么需要大量地修改Product和Factory系列，最好的方法是：



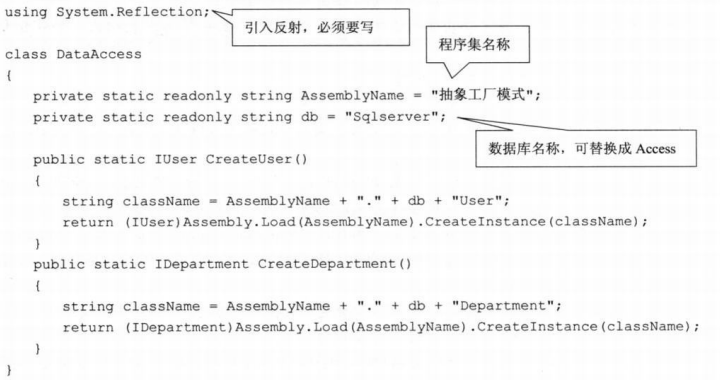
其中的db表示使用哪个数据库：



这样在这里规定了哪个数据库后，客户端：

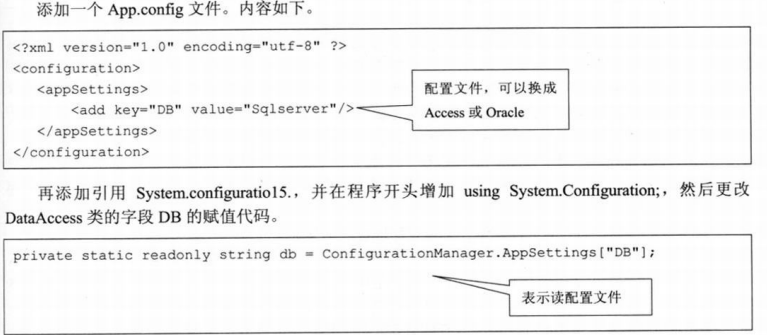


但是switch、if仍然存在，利用反射



可以看到，只需修改db即可

添加配置文件：



但是要注意的是，反射会增加复杂度，如果使用不当会导致成本很高。

Java反射机制

Java反射机制是在运行状态中，对于任意一个类，都能够知道这个类的所有属性和方法；对于任意一个对象，都能够调用它的任意一个方法；这种动态获取的信息以及动态调用对象的方法的功能称为java语言的反射机制。

Java反射机制主要提供了以下功能： 在运行时判断任意一个对象所属的类；在运行时构造任意一个类的对象；在运行时判断任意一个类所具有的成员变量和方法；在运行时调用任意一个对象的方法；生成动态代理。