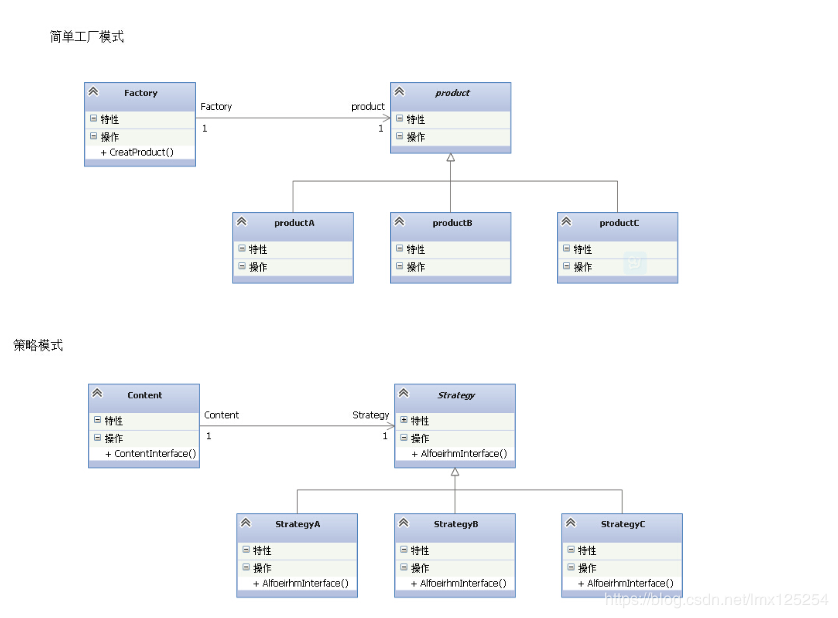
### 简单工厂模式和策略模式的区别

这两种模式大多数时候都是可以混合使用的。

可以看下两种模式的UML图，也是十分像的



但有时还是有一定区别。

工厂模式注重对象的创建，创建好后对象实例交付给调用者使用。

策略模式注重的是计算结果的输出，调用者往往只需要一个获取结果的接口。

比如如下案例：

创建的是一个数据库相关的对象（针对不同的数据库，工厂创建不同类型的实例），然后需要使用这个实例操作表一、表二......表N，对应N个方法。

如果这个案例使用策略模式来实现，那么在策略类中需要把所有这些调用者需要使用到的方法都得在包装一遍。

**所以**，使用工厂模式还是使用策略模式，关键在于调用者需要使用的是对象实例，还是只需要一个计算结果。

（由此可见《大话设计模式》中简单工厂模式的例子不是很恰当，更适合使用策略模式来解决 \_\_\_\_ 个人见解  ( ^\_^ )）

[工厂模式](https://www.baidu.com/s?wd=%E5%B7%A5%E5%8E%82%E6%A8%A1%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d9PhmYrj9bmH9bmHDsPyfk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHmsPWTznHcznHb1rjbvPWn3Ps" \t "https://www.cnblogs.com/yangjian-java/p/_blank)是创建型模式。策略模式是行为性模式。一个关注对象创建。一个关注行为的封装。

策略模式就是定义一系列的算法，这些算法可以在需要的时候替换和扩展．

工厂模式是生成型的模式，在你需要的时候构建具体的实例．

一般情况下，策略模式是为了解决的是策略的切换与扩展，更简洁的说是定义策略族，分别封装起来，让他们之间可以相互替换，策略模式让策略的变化独立于使用策略的客户。可以应用的场景有优惠系统、工资计算系统等。

而工厂模式主要解决的是资源的统一分发，将对象的创建完全独立出来，让对象的创建和具体的使用客户无关。主要应用在多[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "https://www.cnblogs.com/yangjian-java/p/_blank)选择，类库文件加载等。