**九、设计模式**

**88. 说一下你熟悉的设计模式？**

参考：[常用的设计模式汇总，超详细！](http://mp.weixin.qq.com/s?__biz=MzIwMTY0NDU3Nw==&mid=2651938221&idx=1&sn=9cb29d1eb0fdbdb5f976306b08d5bdcc&chksm=8d0f32e3ba78bbf547c6039038682706a2eaf83002158c58060d5eb57bdd83eb966a1e223ef6&scene=21#wechat_redirect)

**89. 简单工厂和抽象工厂有什么区别？**

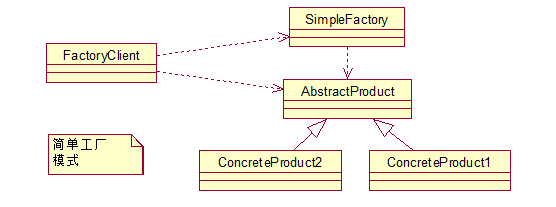
**简单工厂模式**：

这个模式本身很简单而且使用在业务较简单的情况下。一般用于小项目或者具体产品很少扩展的情况（这样工厂类才不用经常更改）。

它由三种角色组成：

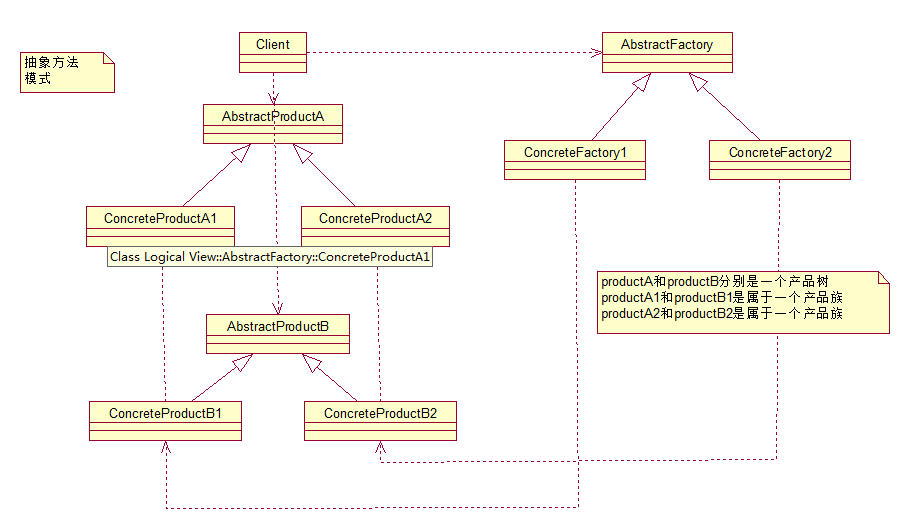
* 工厂类角色：这是本模式的核心，含有一定的商业逻辑和判断逻辑，根据逻辑不同，产生具体的工厂产品。如例子中的Driver类。
* 抽象产品角色：它一般是具体产品继承的父类或者实现的接口。由接口或者抽象类来实现。如例中的Car接口。
* 具体产品角色：工厂类所创建的对象就是此角色的实例。在java中由一个具体类实现，如例子中的Benz、Bmw类。

来用类图来清晰的表示下的它们之间的关系：



**抽象工厂模式：**

先来认识下什么是产品族： 位于不同产品等级结构中，功能相关联的产品组成的家族。



图中的BmwCar和BenzCar就是两个产品树（产品层次结构）；而如图所示的BenzSportsCar和BmwSportsCar就是一个产品族。他们都可以放到跑车家族中，因此功能有所关联。同理BmwBussinessCar和BenzBusinessCar也是一个产品族。

**可以这么说，它和工厂方法模式的区别就在于需要创建对象的复杂程度上。而且抽象工厂模式是三个里面最为抽象、最具一般性的。抽象工厂模式的用意为：给客户端提供一个接口，可以创建多个产品族中的产品对象。**

而且使用抽象工厂模式还要满足一下条件：

1. 系统中有多个产品族，而系统一次只可能消费其中一族产品
2. 同属于同一个产品族的产品以其使用。

来看看抽象工厂模式的各个角色（和工厂方法的如出一辙）：

* 抽象工厂角色： 这是工厂方法模式的核心，它与应用程序无关。是具体工厂角色必须实现的接口或者必须继承的父类。在java中它由抽象类或者接口来实现。
* 具体工厂角色：它含有和具体业务逻辑有关的代码。由应用程序调用以创建对应的具体产品的对象。在java中它由具体的类来实现。
* 抽象产品角色：它是具体产品继承的父类或者是实现的接口。在java中一般有抽象类或者接口来实现。
* 具体产品角色：具体工厂角色所创建的对象就是此角色的实例。在java中由具体的类来实现。