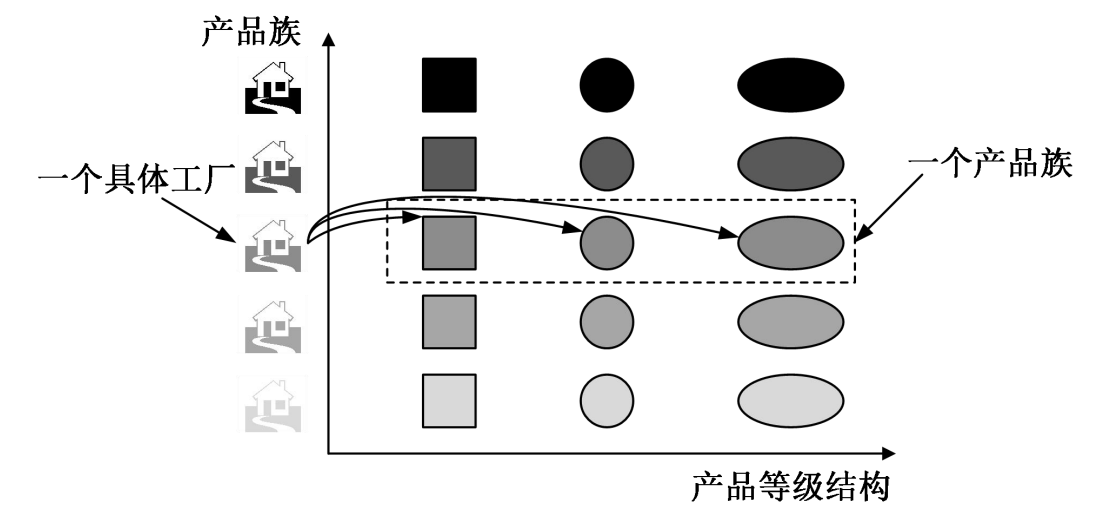


# 抽象工厂模式

在工厂方法模式中具体工厂负责生产具体的产品，每一个具体工厂对应一种具体产品，工厂方法也具有唯一性，一般情况下，一个具体工厂中只有一个工厂方法或者一组重载的工厂方法。但是有时候我们需要一个工厂可以提供多个产品对象，而不是单一的产品对象，也就是说每个产品又分为不同的等级结构，而多个产品处于同一等级结构的产品集合又构成一个产品族，每个具体工厂可以生产一个产品族中的所有产品，此时就要用到**抽象工厂模式**：提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。

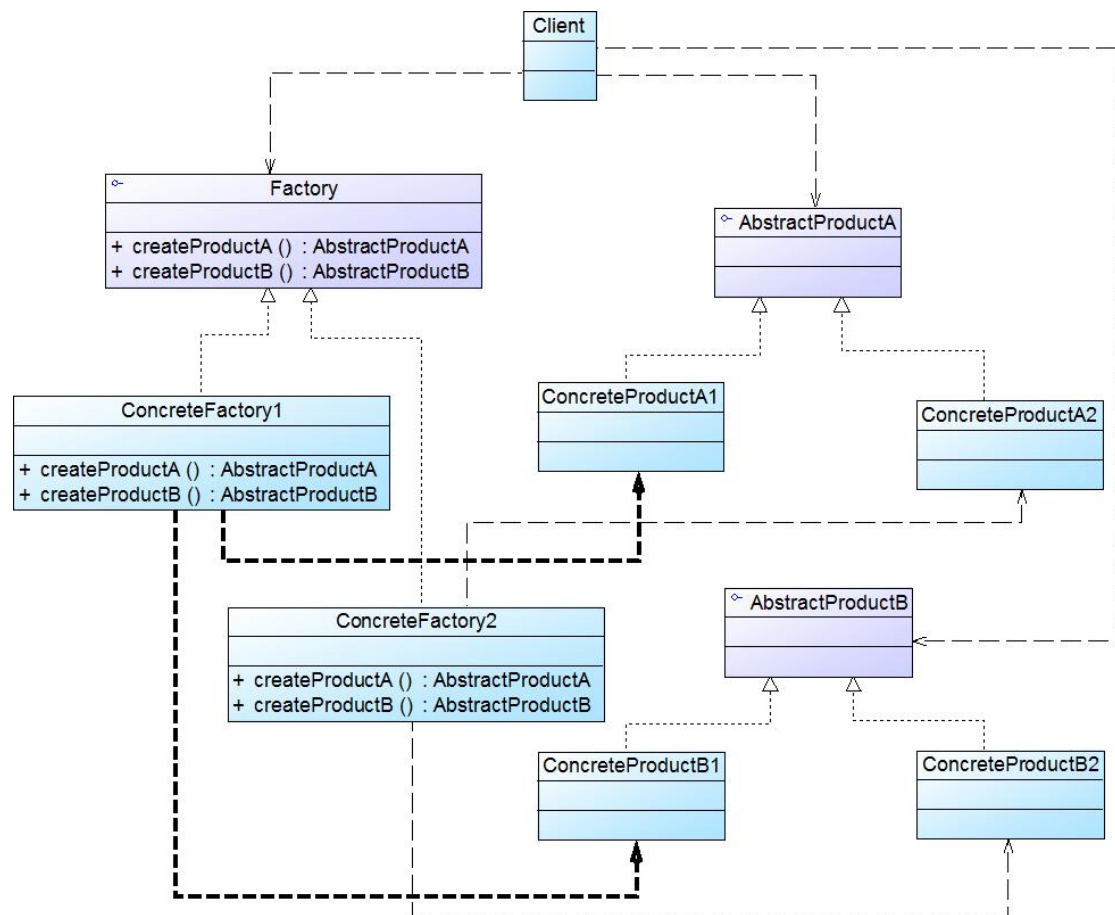
抽象工厂模式的产品划分示意图：



**抽象工厂模式与工厂方法模式最大的区别**在于，工厂方法模式针对的是一个产品等级结构，而抽象工厂模式则需要面对多个产品等级结构，一个工厂可以负责多个处于相同产品等级结构中的产品对象的创建。

比如有一个工厂用于创建人类，在工厂方法模式中，抽象工厂只有一个创建人类的方法，必须派生出六个具体类分别用于创建男性黄色人种、女性黄色人种、男性白色人种、女性白色人种、男性黑色人种、女性黑色人种。而在抽象工厂模式中，抽象工厂应该有三个方法，分别用于创建黄色、白色、黑色人种，再派生出两个具体工厂：男性工厂和女性工厂，这样将人类分为黄色、白色、黑色三个产品结构，再分为男女两个产品族。

抽象工厂模式类图：



注意：工厂方法模式常常潜伏在抽象工厂模式中。抽象工厂模式中的每个工厂都能够生产多个产品，而每个生产产品的方法往往是由工厂方法模式来实现的。在工厂方法模式中，需要为每个实体类都创建一个具体工厂类，而在抽象工厂模式中，对产品进行了整合划分，形成了不同的产品结构，组成了不同的产品族。

在抽象工厂模式中，开闭原则具有倾斜性，增加新的产品族很容易，只需要从抽象工厂中重写派生出一个子类即可，但是增加新的产品等级结构很麻烦，需要修改工厂接口及其所有子类。也就是说，在上面的例子中，增加一个生产中性人类的工厂很容易，但是增加一个棕色人种的产品结构很麻烦。

模式扩展：当抽象工厂模式中每一个具体工厂类只创建一个产品对象，也就是只存在一个产品等级结构时，抽象工厂模式退化成工厂方法模式；当工厂方法模式中抽象工厂与具体工厂合并，提供一个统一的工厂来创建产品对象，并将创建对象的工厂方法设计为静态方法时，工厂方法模式退化成简单工厂模式。