**抽象工厂模式**

抽象工厂模式为创建一组相关或相互依赖的对象提供一个接口，而且无需指定他们的具体类。也就是对一组具有相同主题的工厂进行封装 。比如：生产一台PC机，使用工厂方法模式的话，一般会有cpu工厂，内存工厂，显卡工厂...但是使用抽象工厂模式的话，只有一个工厂就是PC工厂，但是一个PC工厂涵盖了cpu工厂，内存工厂，显卡工厂等要做的所有事。工厂模式：定义一个用于创建对象的借口，让子类决定实例化哪一个类。这个区别在于产品，如果产品单一，最合适用工厂模式，但是如果有多个业务品种、业务分类时，通过抽象工厂模式产生需要的对象是一种非常好的解决方式。再通俗深化理解下：工厂模式针对的是一个产品等级结构 ，抽象工厂模式针对的是面向多个产品等级结构的。

其实无论是简单工厂模式，工厂方法模式，还是抽象工厂模式，他们都属于工厂模式，在形式和特点上也是极为相似的，他们的最终目的都是为了解耦。在使用时，我们不必去在意这个模式到底工厂方法模式还是抽象工厂模式，因为他们之间的演变常常是令人琢磨不透的。经常你会发现，明明使用的工厂方法模式，当新需求来临，稍加修改，加入了一个新方法后，由于类中的产品构成了不同等级结构中的产品族，它就变成抽象工厂模式了；而对于抽象工厂模式，当减少一个方法使的提供的产品不再构成产品族之后，它就演变成了工厂方法模式。 所以，在使用工厂模式时，只需要关心降低耦合度的目的是否达到了。