

# 初级认识

工厂模式一般理解就是减少new创建对象的方式，用接口的方式来返回一个对象，而new创建的方式被封装了。这个认识不足以理解到工厂方式的真正目的。

# 整体介绍

- 一： 工厂模式主要是为创建对象提供了接口。工厂模式一般分为三类：
- 1. 简单工厂模式(Simple Factory)
  - 2. 工厂方法模式(Factory Method)
  - 3. 抽象工厂模式(Abstract Factory)
- 这三种模式从上到下逐步抽象，并且更具一般性。
- 还有一种分类，就是将简单工厂模式看为工厂方法模式的一种特例，两个归为一类。两者皆可。
- 二： 什么样的情况下我们应该记得使用工厂模式呢？大体有两点：
- 1.在编码时不能预见需要创建哪种类的实例。
  - 2.系统不应依赖于产品类实例如何被创建、组合和表达的细节

# 简单工厂模式

简单工厂模式的元素：

- 1. 工厂（Factory）角色：接受客户端的请求，通过请求负责创建相应的产品对象。
- 2. 抽象产品（Abstract Product）角色：是工厂模式所创建对象的父类或是共同拥有的接口。可是抽象类或接口。
- 3. 具体产品（ConcreteProduct）对象：工厂模式所创建的对象都是这个角色的实例。

简单工厂模式的演变：

- 1.) 当系统中只有唯一的产品时，可以省略抽象产品。这样，工厂角色与具体产品可以合并。

简单工厂模式的优缺点：

- 1.) 工厂类含有必要的创建何种产品的逻辑，这样客户端只需要请求需要的产品，而不需要理会产品的实现细节。
- 2.) 工厂类只有一个，它集中了所有产品创建的逻辑，它将是整个系统的瓶颈，同时造成系统难以拓展。
- 3.) 简单工厂模式通常使用静态工厂方法，这使得工厂类无法由子类继承，这使得工厂角色无法形成基于继承的等级结构。
- 4.) 简单工厂模式会包含过多的判断条件，维护起来不是特别方便。

# 工厂方法模式

工厂方法模式的元素：

- creator（创建者）
- product（产品）

工厂模式：在基类中定义一个用于创建对象的接口，让子类决定实例化哪一个类，工厂方法要解决的问题是对象的创建时机，它提供了一种扩展的策略，很好地符合了开放封闭原则。工厂方法也叫做虚构造器（Virtual Constructor）

工厂方法模式相比简单工厂模式：

- 1：工厂方法模式克服了简单工厂模式违背开放-封闭原则的缺点，保持了封装对象创建过程的优点。
- 2：要更换对象时，无论用简单工厂模式还是工厂方法模式，都可以不用做大的改动，就可以实现，降低客户程序与产品对象的耦合。
- 3：工厂方法模式是简单工厂模式的进一步抽象和推广，缺点是，每加一个产品，就要加一个产品工厂的类，增加了开发量。

# 抽象工厂模式

一般来说，抽象工厂最简单形态也至少有4个元素：

- 客户端（client）
- 工厂（factory）
- 产品A（product A）
- 产品B（product B）

## 抽象工厂模式与工厂方法模式的联系与区别

联系：“抽象工厂”模式依赖于“工厂方法”模式。抽象工厂强调的是前面的动词“抽象”，也就是说，将工厂方法模式中的工厂方法抽象出来的那个"动作或设计"就是“抽象工程”模式了。

-----

工厂方法模式（一个产品等级结构）：  
一个抽象产品类，可以派生出多个具体产品类。  
一个抽象工厂类，可以派生出多个具体工厂类。  
每个具体工厂类只能创建一个具体产品类的实例。

抽象工厂模式（多个产品等级结构）：  
多个抽象产品类，每个抽象产品类可以派生出多个具体产品类。  
一个抽象工厂类，可以派生出多个具体工厂类。  
每个具体工厂类可以创建多个具体产品类的实例。

区别：工厂方法模式只有一个抽象产品类，而抽象工厂模式有多个。  
工厂方法模式的具体工厂类只能创建一个具体产品类的实例，而抽象工厂模式可以创建多个。

=====

# 总结

以花园种菜为例

- 1.如果一个花园只种蔬菜类,那么就用简单工厂就可以了.
- 2.如果花园蔬菜品种繁多.得用工厂方法才可以,把共有的东西抽象出来.
- 3.如果要扩大后花园的规模,比如一个在北方,一个在南方,这样工厂方法就无法实现了,就应当用抽象工厂,把各种各样的植物,又组成一个后花园.

简单工厂是一个工厂只生产一类的产品,面对的是具体的类,工厂方法是可以生产不同的产品,把公共的方法抽象出来,然后进行创建各种各样的产品.

抽象工厂把几种产品划出共同的东西,把相互依赖的对象抽象出来,只要实现这些接口就可以得到不同的产品.