java设计模式（1）：工厂模式

1. 面向对象

一切事物皆对象，通过面向对象的方式，将现实世界的事物抽象成对象，现实世界中的关系抽象成类、继承，帮助人们实现对现实世界的抽象与数字建模。面向对象是指一种程序设计范型，同时也是一种程序开发的方法。对象指的是类的集合。它将对象作为程序的基本单元，将程序和数据封装其中，以提高软件的维护性、复用性、灵活性和扩展性。

面向对象的分析设计编程思想，通过封装、继承、多态降低程序，用设计模式使程序更加灵活，容易修改，并且易于复用。编程时，尽量将重复的代码抽象出来，降低耦合度，不然维护起来，非常困难。

“编程是一门技术，更是一门艺术。”不能只满足于写完代码运行结果正确就完事，还要时常考虑如何让代码更简练，更加容易维护，容易扩展和复用。

二、工厂模式

（1）简单工厂模式

简单工厂模式的实质是由一个工厂类根据传入的参数，动态决定应该创建哪一个产品类（这些产品类继承自一个父类或接口）的实例。由于简单工厂很容易违反高内聚责任分配原则，因此一般只在很简单的情况下应用，否则对于简单工厂的压力太大了。

（2）工厂方法模式

为了避免一个简单工厂可以生产各种东西，导致简单工厂压力过大的情况，产生了工厂方法模式。工厂方法模式就是把某一类/某一种东西交由一个工厂生产，同时增加某一类”东西“并不需要修改工厂类，只需要添加生产这类”东西“的工厂即可，使得工厂类符合开放-封闭原则。

（3）抽象工厂模式

当要产生一系列相似的产品时，我们将构建一系列类似的工厂。由于这些工厂有很多相似的地方，我们将这一系列的工厂抽象化为一个统一的构建工厂的标准，对于每一个具体的工厂，有其具体的实现。这就利用了抽象工厂模式。

在简单工厂和工厂方法中理论上都是可以新增任意”东西“的，但是由于抽象工厂模式有一个统一的标准，生产的”东西“已经被限定住了，如果想要额外生产其他的东西，会导致一系列的工厂发送改变，将造成不可估计的影像。

抽象设计模式讲的比较好的博客：<http://men4661273.iteye.com/blog/1637880>

三、abstract class和interface的区别

抽象类(abstract class)可以有抽象方法,也可以有具体是方法,抽象类只能支持单继承。

interface 接口可以有抽象的方法,不存在具体的方法,接口可以多继承(多实现)。

abstract class在Java语言中体现了一种继承关系，要想使得继承关系合理，父类和派生类之间必须存在"is-a"关系，即父类和派生类在概念本质上应该是相同的。对于interface来说则不然，并不要求interface的实现者和interface定义在概念本质上是一致的， 仅仅是实现了interface定义的约定而已。