

# 三种工厂模式区别总结

2013-10-07 10:40 15647人阅读 评论(1) 收藏<sup>[1]</sup> 举报

工厂模式分为三种：简单工厂、工厂模式和抽象工厂模式。三者之间存在哪些异同呢？先分别看看各个模式的特点

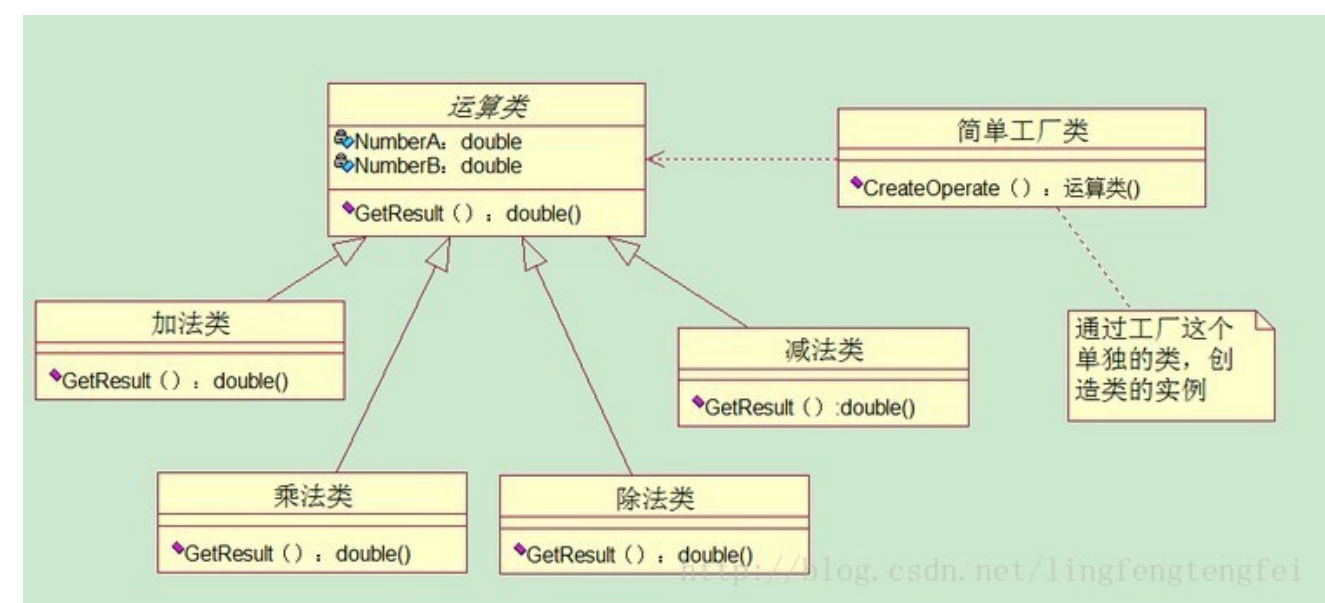
简单工厂模式：

实现了算法<sup>[2]</sup>和界面的分离，也就是将业务逻辑和界面逻辑分开了，降低了耦合度。

算法的封装：定义一个抽象的算法接口，提供不同算法的公共接口方法。其他具体算法继承这个抽象类，并实现具体的算法。

简单工厂类：作为一个独立的类，实现了针对不同的算法进行实例化。

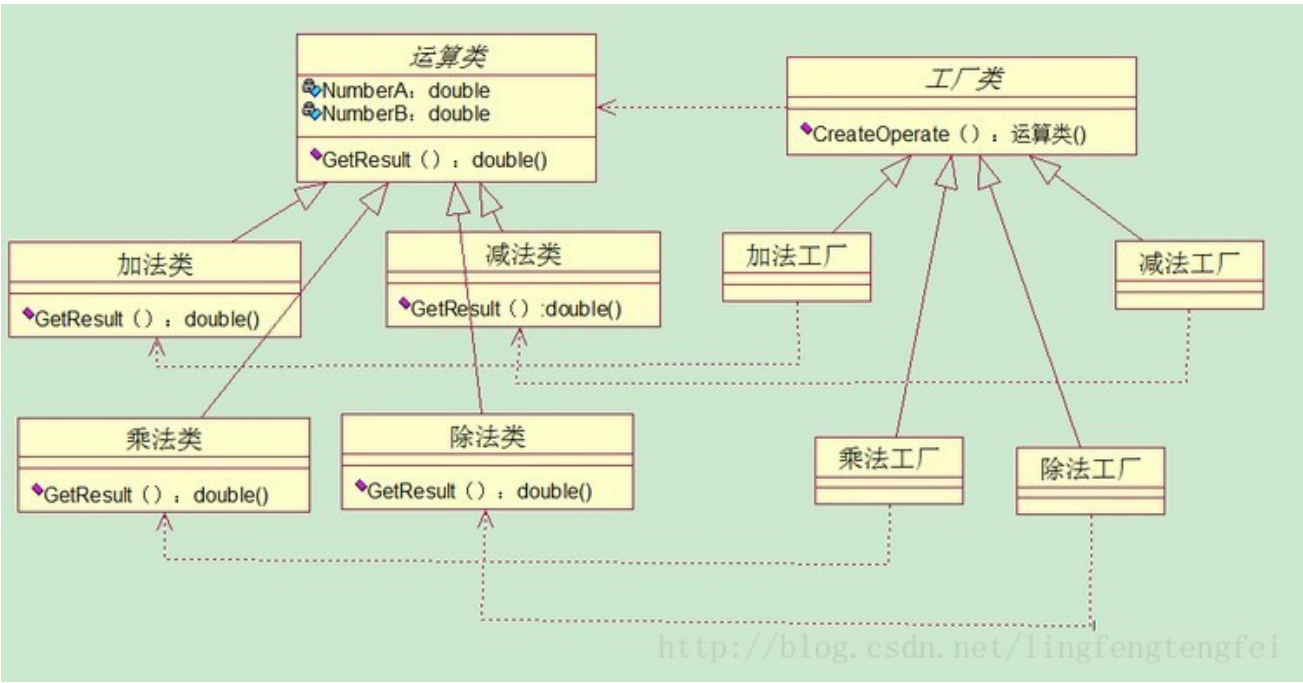
简单工厂的



工厂模式：

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。 遵循了开放—封闭原则。

工厂模式的



简单工厂与工厂模式的共同优点：

都集中封装了对象的创建，使得要更换对象时不需要做大的改动就可实现，降低了客户端程序与产品对象的耦合。

简单工厂模式：

工厂模式的优点：

工厂模式是简单工厂模式的进一步抽象和推广。它遵循了“开放—封闭”原则。

简单工厂模式的优点：

简单工厂模式最大的优点在于工厂类中包含了必要的逻辑判断，根据客户端的选择条件动态实例化相关的类，对于客户端来说，去除了与具体产品的依赖。

工厂模式的缺点：

工厂方法把简单工厂的内部逻辑判断转移到了客户端代码来执行；每增加一产品就要增加一个产品工厂的类，增加了额外的开发量。

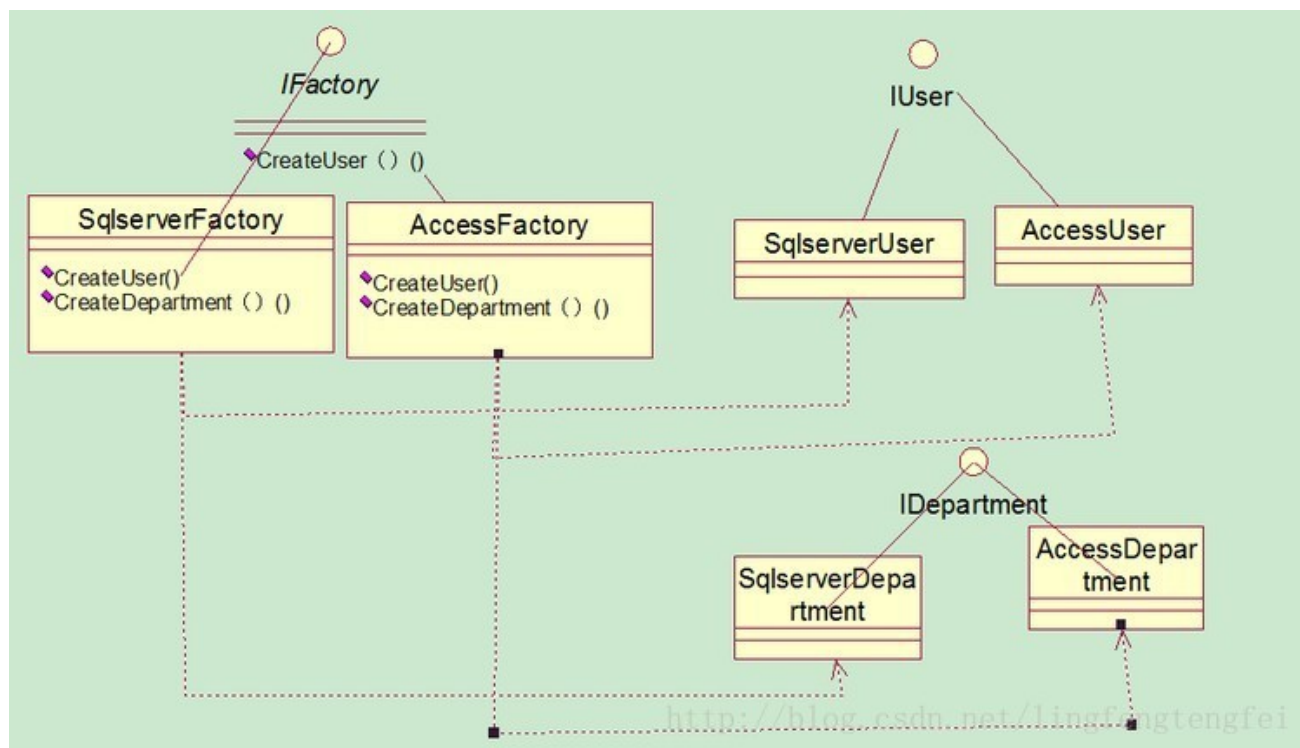
简单工厂的缺点：

是没有遵守开放—封闭原则。所谓的“开放封闭”原则就是开放接口，封闭修改。如果将

来需要添加一个开方的算法，那么，在简单工厂模式中，就必须在简单工厂类中添加相应的判断语句！另外，在简单工厂类中利用了Switch语句，这对程序的扩展本身就不不利。

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。



工厂模式和抽象工厂的区别：

操作类的时候，只需要工厂方法模式；但数据库<sup>[3]</sup>中有很多的表，而access又是两大不同的分类，所以就延伸到了抽象工厂模式

抽象工厂模式的优点：

、易于交换产品系列，由于具体工厂类，在一个应用程序中只需要在初始化的时候出现一次，这就使得改变一个应用的具体工厂变得非常容易，它只需改变具体工厂即可使用不同的产品配置。

、它让具体的创建实例过程与客户端分离，客户端是通过他们的抽象接口操纵实例，产品的具体类名也被具体工厂的实现分离，不会出现在客户端代码中。

抽象模式的缺点：

、抽象模式虽然便于两数据库之间的切换，但是不便于增加需求功能。

个调用数据库访问的类，就需要多次实例化此具体工厂类。

抽象工厂模式、反射以及配置文件：

反射方法的实质是在对象实例化的时候传引用，将程序由编译时转为运行时，通过字符串变量来处理，去除了、switch判断的麻烦。但是如果数据库在更换时，还需要去修改程序（字符串的值）重编译。

通过添加配置文件可以解决更改DataAccess

<http://lvxingzhelimin.blog.163.com/blog/static/170716550201110852956542/><sup>[4]</sup>

## Links

1. javascript:void(0);
2. <http://lib.csdn.net/base/datastructure>
3. <http://lib.csdn.net/base/mysql>
4. <http://lvxingzhelimin.blog.163.com/blog/static/170716550201110852956542/>