

<http://www.strolling.cn>

<http://weibo.com/fujianthinking>

jiafu1115

排名： 第1051名

译文: 1篇 评论: 71条

- Design Pattern (7)
- Database|Oracle (20)
- Java|J2SE (445)
- Linux|CentOS (26)
- InfoSystem Program (4)
- Education |Mobile (2)
- Others (27)
- Web Service (25)
- Maven (26)
- JVM (22)
- Site Collection (4)
- Concurrency (4)
- Java IO (23)
- Java Doc (4)
- NetWork (4)
- C++ (271)
- IDE (4)

2011-07-08 17:01 1508人阅读 [评论\(0\)](#)

分类: **Design Pattern (6)**

<http://173806613.iteye.com/blog/907818>

1 <工厂模式与抽象工厂区别>

工厂方法模式：

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。

2 <简单工厂模式、工厂方法模式和抽象工厂模式异同>

简单工厂模式，工厂方法模式和抽象工厂模式都是属于创建型设计模式，这三种创建型模式都不需要知道具体类。我们掌握一种思想，就是在创建一个对象时，需要把容易发生变化的地方给封装起来，来控制变化（哪里变化，封装哪里），以适应客户的变动，项目的扩展。用这三种设计模式都可以实现，那究竟这三种设计模式有什么异同呢？下面根据这三者之间的特点，优点，缺点，适用范围进行比较。

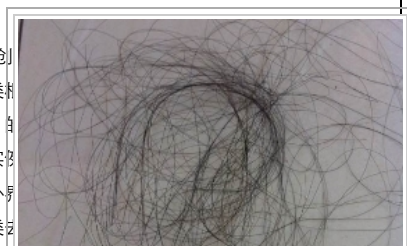
一、特点

关闭

简单工厂模式：专门定义一个类来负责创建其他类的实例，被创建对象都满足共同的接口。静态工厂方法模式。它的实质是由一个工厂类来创建产品类（这些产品类继承自一个父类或接口）的实例。工厂类创建的对象都是充当这个角色的某个具体类的实例。工厂类是创建对象的关键所在。它包含必要的判断逻辑，能够根据外界传入的某种标识（一个字符串、一个整数、一个枚举类型的数字等等）来取得相应的产品类实例。用户在使用时可以直接根据工厂类来创建对象，而不需要知道如何创建以及如何组织的。有利于整个软件体系

工厂方法模式：工厂方法是粒度很小的设计模式，因为模式的表

对象的接口，让子类决定实例化具体的某一个类。



头发为什么会掉



[illegible]

再负责产品的创建，由接口针对不同条件返回具体的类实例，由具体类实例去实现。工厂方法模式是简单工厂模式的衍生，解决了许多简单工厂模式的问题。首先完全实现‘开-闭原则’，实现了可扩展。其次实现更复杂的层次结构，可以应用于产品结果复杂的场合。工厂方法模式是对简单工厂模式进行了抽象。有一个抽象的**Factory**类（可以是抽象类和接口），这个类将不在负责具体的产品生产，而是只制定一些规范，具体的生产工作由其子类去完成。在这个模式中，工厂类和产品类往往可以依次对应。即一个抽象工厂对应一个抽象产品，一个具体工厂对应一个具体产品，这个具体的工厂就负责生产对应的产品。

抽象工厂模式：抽象工厂模式是所有形态的工厂模式中最为抽象和最具一般性的一种形态。抽象工厂模式是指当有多个抽象角色时，使用的一种工厂模式。抽象工厂模式可以向客户端提供一个接口，使客户端在不必指定产品的具体的情况下，**创建多个产品族中的产品对象**。它有多个抽象产品类，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。每一个模式都是针对一定问题的解决方案，工厂方法模式针对的是一个产品等级结构；而抽象工厂模式针对的是多个产品等级结果。

二. 优点

简单工厂模式：工厂类含有必要的判断逻辑，可以决定在什么时候创建哪一个产品类的实例，客户端可以免除直接创建产品对象的责任，而仅仅“消费”产品。简单工厂模式通过这种做法实现了对责任的分割。简单工厂模式能够根据外界给定的信息，决定究竟应该创建哪个具体类的对象。通过它，外界可以从直接创建具体产品对象的尴尬局面中摆脱出来。外界与具体类隔离开来，偶合性低。明确区分了各自的职责和权力，有利于整个软件体系结构的优化。

工厂方法模式：工厂方法模式是为了克服简单工厂模式的缺点（主要是为了满足OCP）而设计出来的。简单工厂模式的工厂类随着产品类的增加需要增加很多方法（或代码），而工厂方法模式每个具体工厂类只完成单一任务，代码简洁。工厂方法模式完全满足OCP，即它有非常良好的扩展性。

抽象工厂模式：抽象工厂模式主要在于应对“新系列”的需求变化。分离了具体的类，抽象工厂模式帮助你控制一个应用创建的对象类，因为一个工厂封装创建产品对象的责任和过程。它将客户和类的实现分离，客户通过他们的抽象接口操纵实例，产品的类名也在具体工厂的实现中被分离，它们不出现在客户代码中。它使得易于交换产品系列。一个具体工厂类在一个应用中仅出现一次——即在它初始化的时候。这使得改变一个应用的具体工厂变得很容易。它只需改变具体的工厂即可使用不同的产品配置，这是因为一个抽象工厂创建了一个完整的产品系列，所以整个产品系列会立刻改变。它有利于产品的一致性。当一个系列的产品对象被设计成一起工作时，一个应用一次只能使用同一个系列中的对象，这一点很重要，而抽象工厂很容易实现这一点。抽象工厂模式有助于这样的团队的分工，降低了模块间的耦合性，提高了团队开发效率。

三. 缺点

简单工厂模式：当产品有复杂的多等级结构时，工厂类只有自己，以不变应万变，就是模式的缺点。因为工厂类集中了所有产品创建逻辑，一旦不能正常工作，整个系统都要受到影响。系统扩展困难，一旦添加新产品就不得不修改工厂逻辑，有可能造成工厂逻辑过于复杂，违背了“开放--封闭”原则(OCF)。另外，简单工厂模式通常使用静态工厂方法，这使得无法由子类继承，造成工厂角色无法形成基于继承的等级结构。

工厂方法模式：不易于维护，假如某个具体产品类需要进行一定的修改，很可能需要修改对应的工厂类。当同时需要修改多个产品类的时候，对工厂类的修改会变得相当麻烦（对号入座已经是个问题了）。

抽象工厂模式：抽象工厂模式在于难于应付“新对象”的需求变动。难以支持新种类的产品。难以扩展抽象工厂以生产新种类的产品。这是因为抽象工厂几乎确定了可以被创建的产品集合，支持新种类的产品就需要扩展该工厂接口，这将涉及抽象工厂类及其所有子类的改变。

a030703130: java正则表达式教程 学习地址: http://www.java3z.com/cwbwebhome/...
replaceAll ()/appendReplacemer
a030703130: 正则表达式matcher类 学习地址: http://www.java3z.com/cwbwebhom...

评论排行

知识都学杂了 (8)

Java解惑笔记<不断更新: (4)

JAVA多态与类型转化分析 (3)

读完《重构》的一些感想 (3)

关于Coding的学习与思考 (3)

instanceOf 与 isInstance (3)

简单工厂模式、工厂方法 (3)

java+方法覆盖必须不减 (3)

Access restriction : The t (3)

java 正斜杠与反斜杠之分 (3)

好友博客

良师益友 Java Tips
http://www.javamex.com

四. 适用范围

简单工厂模式：工厂类负责创建的对象比较少，客户只知道传入了工厂类的参数，对于始何创建对象（逻辑）不关心。

工厂方法模式：当一个类不知道它所必须创建对象的类或一个类希望由子类来指定它所创建的对象时，当类将创建对象的职责委托给多个帮助子类中的某一个，并且你希望将哪一个帮助子类是代理者这一信息局部化的时候，可以使用工厂方法。

抽象工厂模式：一个系统不应当依赖于产品类实例如何被创建、组合和表达的细节，这对于所有形态的工厂模式都是重要的。这个系统有多于一个的产品族，而系统只消费其中某一产品族。同属于同一个产品族的产品是在一起使用的，这一约束必须在系统的设计中体现出来。系统提供一个产品类的库，所有的产品以同样的接口出现，从而使客户端不依赖于实现。

其实，无论是简单工厂模式、工厂模式还是抽象工厂模式，它们本质上都是将不变的部分提取！的部分留作接口，以达到最大程度上的复用。究竟用哪种设计模式更适合，这要根据具体的业务需求来决定。

顶0

踩0

上一篇 项目管理“九五”矩阵

下一篇 信息系统管理基础知识点整理

我的同类文章

Design Pattern（6）					
• 欢迎使用CSDN-markdown...	2015-07-10	阅读 84	• 抽象工厂和Builder模式区别	2011-08-08	阅读 978
• 对代理模式与Java动态代理...	2011-07-30	阅读 211	• 代理模式和装饰者模式的异...	2011-07-09	阅读 358
• Java设计 生成器模式	2011-07-09	阅读 271	• Java设计 单例模式	2011-07-09	阅读 249

主题推荐

工厂模式preclass

猜你在找


设计模式课程	C#抽象工厂模式的几种实现方法及比较
Android设计模式精解(第7课)：Adapter模式	抽象工厂模式第一步-传统实现方法
Android设计模式精解(第10课)：状态模式(State pa	C#抽象工厂模式的几种实现方法及比较
C语言系列之 进程通讯与相关设计模式	C#抽象工厂模式的几种实现方法及比较
MySQL SQL优化及高可用公开课视频分享	玩转抽象工厂模式

关闭

查看评论

- 2楼 felixit0120 2011-08-16 10:03发表
 转载的也非常不错，讲得非常全面，UML图，说明，异同点都有，看了之后，非常明白，理解很透彻。
^ _ ^
- 1楼 felixit0120 2011-08-15 17:39发表
 写得不出，非常详细，谢啦

Re: [jiafu1115](#) 2011-08-15 20:56发表

 回复felixit0120: 都是抄袭的，多多交流，哈

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服 杂志客服 微博客服 webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司[技术支持](#)

京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved 

关闭

