

【VIP直播课】

为什么要从设计模式开始 及工厂模式详解

Tom





前中电在线技术总监 前超星网架构师

10余年Java经验。精通java语言。开发过多套企业UI框架、ORM框架。热衷于分享经验，共同进步。

不只做一个技术者，更要做一个思考者。





书法爱好者、绘画爱好者 编程界字写得最好的 书法界编程最牛逼的

自幼开始练习书法。中学期间，曾获市级青少年杯书法竞赛一等奖，获校园杯美术竞赛工笔画一等奖，获校园征文比赛二等奖。大学担任学生会宣传部长，负责校园黑板报、校园刊物的编辑排版设计。

参加工作后，担任过家具建模、平面设计等工作。亲自设计咕泡学院Logo。



扫码加入书法兴趣小组



群名称:Tom老师书法兴趣小组
群 号:549209007

扫码加入书法兴趣小组



- 1、通过对本章内容的学习，了解设计模式的由来。
- 2、介绍设计模式能帮我们解决哪些问题。
- 3、剖析工厂模式的历史由来及应用场景。



- 1、用设计模式能帮助我们更好地解决实际问题。
- 2、学习前人是如何总结经验的，把经验为自己所用。
- 3、培养将业务需求转换技术实现的能力。





为什么要从设计模式开始

专注互联网IT教育，做技术人的指路明灯，职场生涯的精神导师。
咕泡学院官网：<http://www.gupaoedu.com>



专业互联网
IT教育服务平台

1、开闭原则

2、单一职责原则

3、依赖倒置原则

4、接口隔离原则

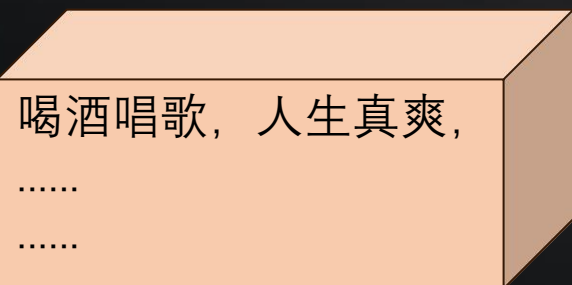
5、迪米特法则

6、里氏替换原则

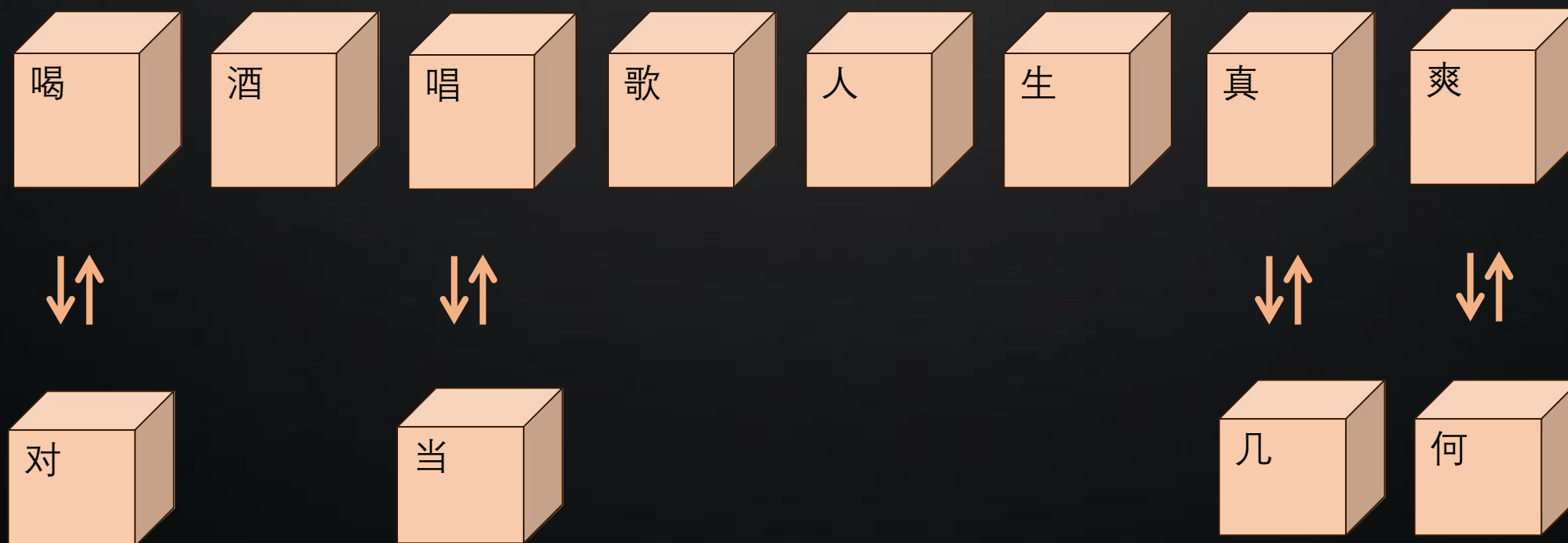
7、合成复用原则



在学设计模式之前，你可能会这样感叹



学完设计模式之后，你可能会这样感叹



大家对比一下前后的区别，有何感受？



- 1、写出优雅的代码
- 2、更好地重构项目
- 3、经典框架都在用设计模式解决问题



工厂模式 BeanFactory
装饰器模式 BeanWrapper
代理模式 AopProxy
单例模式 ApplicationContext
委派模式 DispatcherServlet
策略模式 HandlerMapping
适配器模式 HandlerAdapter
模板方法模式 JdbcTemplate
观察者模式 ContextLoaderListener



- 1、《软件架构设计七大原则》课件，是预习资料。
- 2、维基百科：http://en.wikipedia.org/wiki/Design_Patterns#Patterns_by_Type
- 3、推荐书籍《大话设计模式》
- 4、推荐书籍《Head First设计模式》
- 5、推荐书籍《设计模式-可复用面向对象软件的基础》
- 6、推荐书籍《Effective Java》

Spring IOC 工厂、单例、装饰器

Spring AOP 代理、观察者

Spring MVC 委派、适配器

Spring JDBC 模板方法



工厂模式详解

专注互联网IT教育，做技术人的指路明灯，职场生涯的精神导师。
咕泡学院官网：<http://www.gupaoedu.com>



专业互联网
IT教育服务平台



原始社会自给自足



农耕社会小作坊



工厂流水线生产



现代产业链代工厂

简单工厂模式（ Simple Factory Pattern ）是指由一个工厂对象决定创建出哪一种产品类的实例。

属于创建型模式，但它不属于GOF，23种设计模式。



工厂类负责创建的对象较少。

客户端只需要传入工厂类的参数，对于如何创建对象的逻辑不需要关心。



只需传入一个正确的参数，就可以获取你所需要的对象
无须知道其创建的细节。



工厂类的职责相对过重，增加新的产品时需要修改工厂类的判断逻辑，违背开闭原则。

不易于扩展过于复杂的产品结构。





工厂方法模式



工厂方法模式（Factory Method Pattern）是指定义一个创建对象的接口，但让实现这个接口的类来决定实例化哪个类，工厂方法让类的实例化推迟到子类中进行。

Define an interface for creating an object, but let subclasses decide which class to instantiate. The Factory method lets a class defer instantiation it uses to subclasses.

属于创建型设计模式。



创建对象需要大量重复的代码。

客户端（应用层）不依赖于产品类实例如何被创建、实现等细节。

一个类通过其子类来指定创建哪个对象。



用户只需关心所需产品对应的工厂，无须关心创建细节。

加入新产品符合开闭原则，提高了系统的可扩展性。



类的个数容易过多，增加了代码结构的复杂度。

增加了系统的抽象性和理解难度。





抽象工厂模式

专注互联网IT教育，做技术人的指路明灯，职场生涯的精神导师。
咕泡学院官网：<http://www.gupaoedu.com>



专业互联网
IT教育服务平台

抽象工厂模式（Abstract Factory Pattern）是指提供一个创建一系列相关或相互依赖对象的接口，无须指定他们具体的类。

Provide an interface for creating families of related or dependent objects without specifying their concrete classes.

属于创建型设计模式。



客户端（应用层）不依赖于产品类实例如何被创建、实现等细节。

强调一系列相关的产品对象（属于同一产品族）一起使用创建对象需要大量重复的代码。

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具体实现。



具体产品在应用层代码隔离，无须关心创建细节

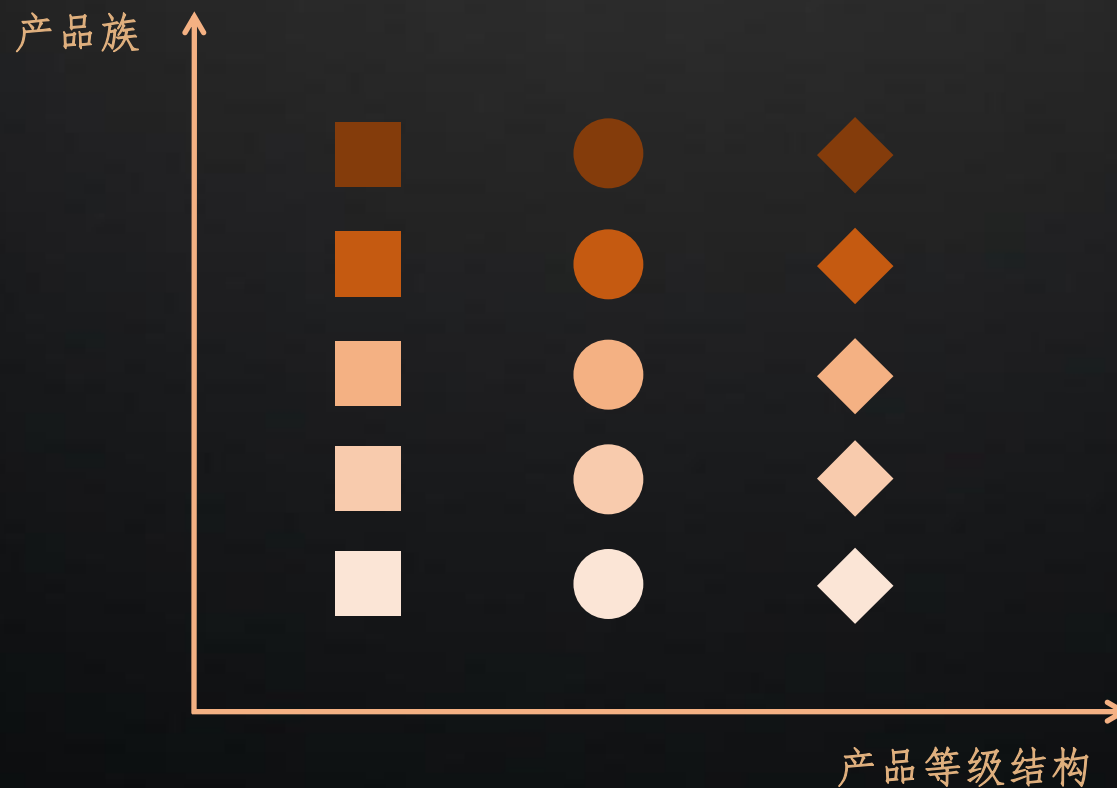
将一个系列的产品族统一到一起创建。

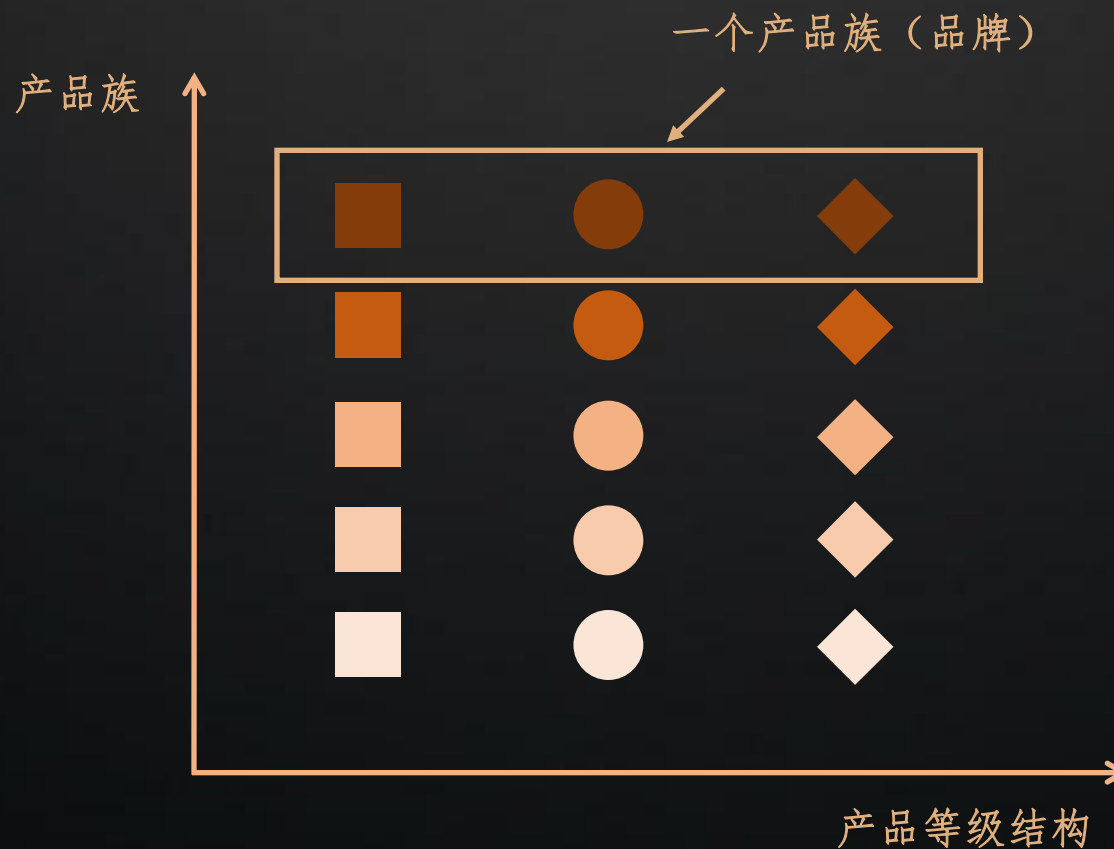


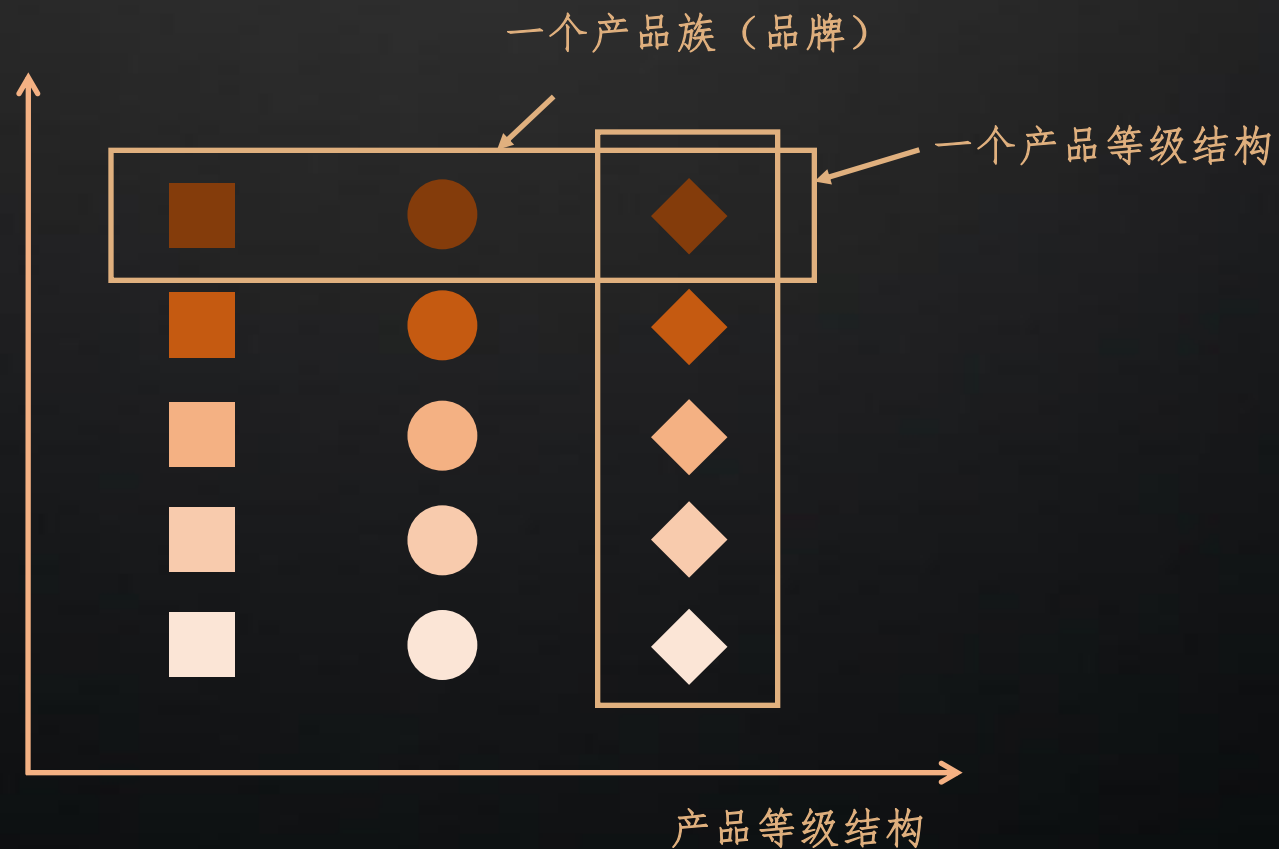
规定了所有可能被创建的产品集合，产品族中扩展新的产品困难，需要修改抽象工厂的接口。

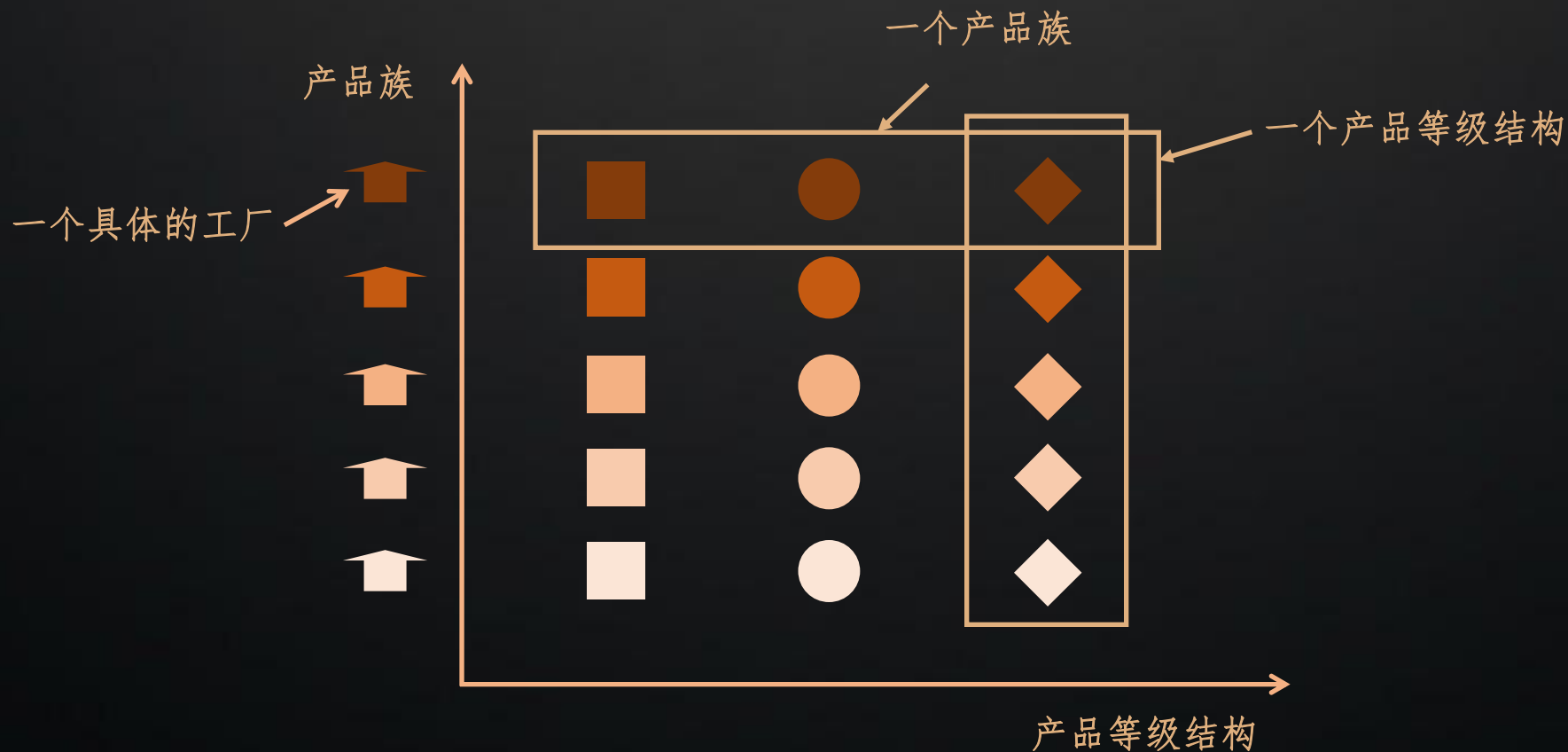
增加了系统的抽象性和理解难度。











谢谢观看

Tom



Tom老师QQ号：441221062

