咕泡学院
gupaoedu.com

【VIP直播课】

各设计模式总结与对比

Tom

做技术人的指路明灯，职场生涯的精神导师。咕泡学院，只为更好的你。
咕泡学院官网：http://www.gupaoedu.com

专 业 互 联 网
I T 教育服务平台

咕泡学院
gupaoedu.com

咕泡学院-Tom
前中电在线技术总监
前超星网架构师
现为咕泡学院联合创始人

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。

专 业 互 联 网
I T 教育服务平台

为什么说来自文艺界？

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

咕泡学院
gupaoedu.com

有兴趣的可以扫码加入书法兴趣小组

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
I T 教育服务平台

关于《21天设计模式强化特训营》

进营条件：完成课后作业，有强化训练需求，乐于奉献，积极学习。

主要模式：训练营是任务闯关形式，建微信群，每天必须打卡，**不打卡会踢群，不完成任务会踢群**。进群要签协议，并缴纳一定的保证金。

训练效果：培养101名成绩优异者进入名企。**提前退出者保证金乐捐，最终胜出者获得奖金。**

有机会成为**咕泡学院助教**。

课程目标

1、简要分析GOF 23种设计模式和设计原则，做整体认知。

2、剖析Spring的编程思想，启发思维，
为之后深入学习Spring做铺垫。

3、了解各设计模式之间的关联，解决设计模式混淆的问题。

1、设计模式在于理解，不只在于形式。

2、不要为了套用设计模式而使用设计模式，而是，在业务上到遇到问题时，很自然地想到设计模式作为一种解决方案。

GOF 23种设计模式简介

设计模式是一门艺术
设计模式来源于生活

孔子人生之道就是模板模式

从出生元婴、

二十加冕、三十而立、

四十不惑、五十知天命、

六十花甲、七十古稀不逾矩、

八、九十耄耋

……

GOF 23种设计模式

分类	设计模式
创建型	工厂方法模式（Factory Method）、抽象工厂模式（Abstract Factory）、建造者模式（Builder）、原型模式（Prototype）、单例模式(Singleton)
结构型	适配器模式(Adapter)、桥接模式（Bridge）、组合模式（Composite）、 装饰器模式（Decorator）、门面模式（Facade）、享元模式（Flyweight）、代理模式（Proxy）
行为型	解释器模式（Interpreter）、模板方法模式（Template Method）、责任链模式（Chain of Responsibility）、命令模式（Command）、迭代器模式（Iterator）、调解者模式（Mediator）、备忘录模式（Memento）、观察者模式（Observer）、状态模式（State）、策略模式（Strategy）、 访问者模式（Visitor）

设计模式之间的关联关系和对比

说说哪些设计模式容易混淆？

工厂类一般就是被设计为单例。
ApplicationContext

工厂模式包含工厂方法和抽象工厂属于创建型模式
策略模式属于行为型模式

工厂模式主要目的是封装好创建逻辑，策略模式接收工厂创建好的对象，从而实现不同的行为

创建：new
行为：invoke

1、策略模式是委派模式内部的一种实现形式，策略模式关注的是结果是否能相互替代。

支付方式：AliPay，WechatPay ...

2、委派模式更关注分发和调度的过程。

有可能采用if...else...条件分支语句来分发，内部也可以使用策略模式

1、工厂方法是模板方法的一种特殊实现

1、模板方法和策略模式都有封装算法。

2、策略模式是使不同算法可以相互替换，且不影响客户端应用层的使用。

3、模板方法是针对定义一个算法的流程，将一些有细微差异的部分交给子类实现。策略模式算法实现是封闭的。

4、模板模式不能改变算法流程，策略模式可以改变算法流程且可替换。策略模式通常用来代替if...else...等条件分支语句。

1、装饰者模式关注点在于给对象动态扩展、添加方法，而代理更加注重控制对对象的访问。

2、代理模式通常会在代理类中创建被代理对象的实例，而装饰者模式通常把被装饰者作为构造参数。

1、装饰者模式和适配器模式都是属于包装器模式（Wrapper）

2、装饰者模式可以实现被装饰者与相同的接口或者继承被装饰者作为它的子类，而适配器和被适配者可以实现不同的接口

1、适配器可以结合静态代理来实现，保存被适配对象的引用，但不是唯一的实现方式。

1、在适配业务复杂的情况下，利用策略模式优化动态适配逻辑

Spring中常用的设计模式对比

如何让学过的设计模式真正属于自己？

不能死记硬背

穷举法 + 类比法

Spring中常用的设计模式

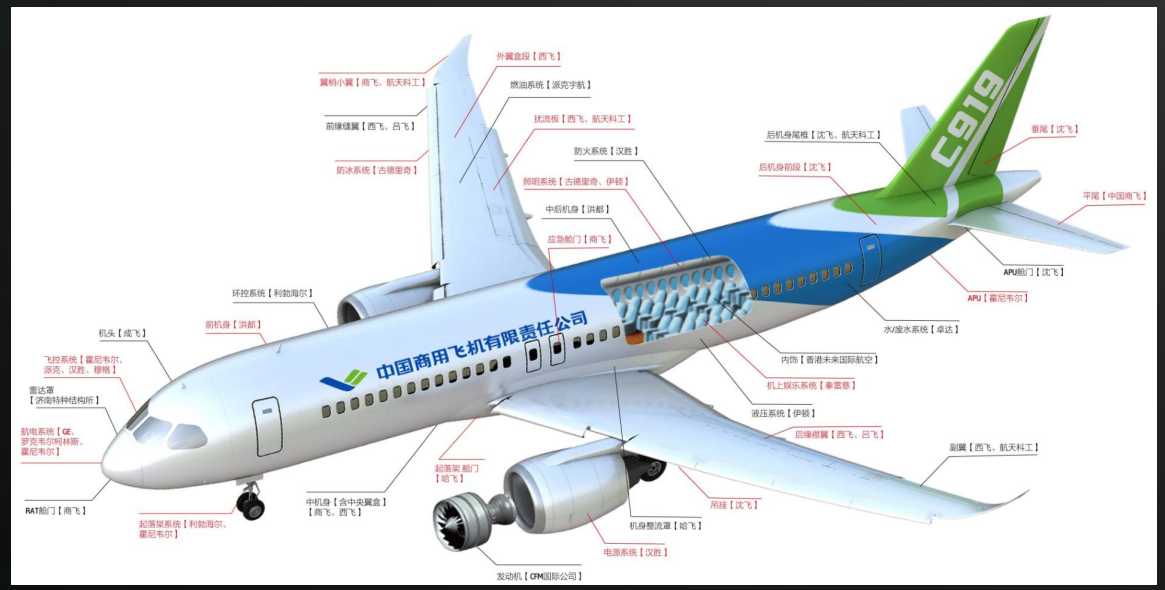
设计模式	一句话归纳	举例
工厂模式（Factory）	只对结果负责，封装创建过程。	BeanFactory、Calendar
单例模式（Singleton）	保证独一无二。	ApplicationContext、Calendar
原型模式（Prototype）	拔一根猴毛，吹出千万个。	ArrayList、PrototypeBean
代理模式（Proxy）	找人办事，增强职责。	ProxyFactoryBean、JdkDynamicAopProxy、CglibAopProxy
委派模式（Delegate）	干活算你的（普通员工），功劳算我的（项目经理）。	DispatcherServlet、BeanDefinitionParserDelegate
策略模式（Strategy）	用户选择，结果统一。	InstantiationStrategy
模板模式（Template）	流程标准化，自己实现定制。	JdbcTemplate、HttpServlet
适配器模式（Adapter）	兼容转换头。	AdvisorAdapter、HandlerAdapter
装饰器模式（Decorator）	包装，同宗同源。	BufferedReader、InputStream、OutputStream、HttpHeadResponseDecorator
观察者模式（Observer）	任务完成时通知。	ContextLoaderListener

Spring中的编程思想总结

Spring中的编程思想总结

Spring思想	应用场景（特点）	一句话归纳
OOP	Object Oriented Programming（面向对象编程）用程序归纳总结生活中一切事物。	封装、继承、多态。
BOP	Bean Oriented Programming（面向Bean编程）面向Bean（普通的java类）设计程序。	一切从Bean开始。
AOP	Aspect Oriented Programming(面向切面编程)找出多个类中有一定规律的代码，开发时拆开，运行时再合并。面向切面编程即面向规则编程。	解耦，专人做专事。
IOC	Inversion of Control（控制反转）将new对象的动作交给Spring管理，并由Spring保存已创建的对象（IOC容器）。	转交控制权（即控制权反转）
DI/DL	Dependency Injection（依赖注入）或者Dependency Lookup（依赖查找）依赖注入、依赖查找，Spring不仅保存自己创建的对象，而且保存对象与对象之间的关系。注入即赋值，主要三种方式构造方法、set方法、直接赋值。	赋值

现实生活中的AOP



飞机组装示意图

汽车组装示意图

AOP的应用场景

Authentication（权限认证）

Auto Caching（自动缓存处理）

Error Handling（统一错误处理）

Debugging（调试信息输出）

Logging（日志记录）

Transactions（事务处理）

……

SpringAOP之必须明白的几个概念

1、Aspect(切面)：通常是一个类，里面可以定义切入点和通知。

2、JointPoint(连接点)：程序执行过程中明确的点，一般是方法的调用。

3、Advice(通知)：AOP在特定的切入点上执行的增强处理，有before、after、afterReturning、afterThrowing、around

4、Pointcut(切入点)：就是带有通知的连接点，在程序中主要体现为书写切入点表达式

SpringAOP之Execution表达式


```
execution(modifiers-pattern? ret-type-pattern declaring-type-pattern? name-pattern(param-pattern) throws-pattern?)
```

modifiers-pattern：方法的操作权限

ret-type-pattern：返回值【必填】

declaring-type-pattern：方法所在的包

name-pattern：方法名 【必填】

parm-pattern：参数名

throws-pattern：异常

咕泡学院
gupaoedu.com

谢谢观看

Tom

Tom老师QQ号：441221062

专 业 互 联 网
I T 教育服务平台

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