咕泡学院
—gupaoedu—

【VIP直播课】

各设计模式总结与对比

Tom

自我介绍

咕泡学院-Tom

前中电在线技术总监 **前超星网架构师**

现任咕泡学院联合创始人

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

《Spring 5核心原理》作者，《Netty 4核心原理》作者；电子工业出版社《Java架构师系列丛书》签约作者。

格言：不只做一个技术者，更要做一个思考者。

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。担任过学生会宣传部长，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。

QQ扫码加入

Tom老师书法兴趣小组

QQ扫码加入
Tom老师易经兴趣小组

咕泡学院
—gupaoedu—

咕泡学院Java架构师成长丛书

课程目标

1、总结设计原则和GoF 23种设计模式，做整体认知。

2、为之后深入学习源码分析做铺垫。

3、了解各设计模式之间的关联，解决设计模式混淆的问题。

内容定位

1、掌握设计模式的“道”，而不只是“术”。

2、道可道非常道，滴水石穿非一日之功，做好长期修炼的准备。

3、不要为了用设计模式去生搬硬套，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。

七大设计原则

开闭原则(OCP)
Open-Close Principle

依赖倒置原则(DIP)
Dependence Inversion Principle

单一职责原则(SRP)
Simple Responsibility Principle

接口隔离原则(ISP)
Interface Segregation Principle

迪米特法则(LoD)
Law of Demeter

里氏替换原则(LSP)
Liskov Substitution Principle

合成复用原则(CARP)
Composite/Aggregate Reuse Principle

一句话归纳七大设计原则

设计原则	一句话归纳	目的
开闭原则(OCP) (Open-Close)	对扩展开放，对修改关闭	减少维护带来新的风险
依赖倒置原则(DIP) (Dependence Inversion)	高层不应该低层	更利于代码结构的升级 扩展
单一职责原则(SRP) (Simple Responsibility)	一个类只干一件事	便于理解，提高代码可读性
接口隔离原则(ISP) (Interface Segregation)	一个接口只干一件事	功能解耦，高聚合、低耦合
迪米特法则(LoD) (Law of Demeter)	不该知道的不要知道	只和朋友交流，不和陌生人说话， 减少代码臃肿
里氏替换原则(LSP) (Liskov Substitution)	子类重写方法功能发生改变， 不应该影响父类方法的含义	防止继承泛滥
合成复用原则(CARP) (Composite/Aggregate Reuse)	尽量使用组合实现代码复用， 而不使用继承	降低代码耦合

《Design Patterns: Elements of Reusable Object-Oriented Software》（即后述《设计模式》一书）。

由 Erich Gamma、Richard Helm、Ralph Johnson 和 John Vlissides 合著（Addison-Wesley，1995）。

这几位作者常被称为"四人组（Gang of Four）"，而这本书也就被称为"四人组（或 GoF）"书。

GoF的设计模式是Java基础知识和J2EE框架知识之间一座隐性的"桥"

GoF设计模式分类

创建型
Creational

结构型
Structural

行为型
Behavioral

创建型(Creational)设计模式使用频率总结

工厂方法模式
Factory Method

抽象工厂模式
Abstract Factory

建造者模式
Builder

单例模式
Singleton

原型模式
Prototype

结构型(Creational)设计模式使用频率总结

- 适配器模式 Adapter
- 装饰器模式 Decorator
- 代理模式 Proxy
- 门面模式 Facade
- 组合模式 Composite
- 享元模式 Flyweight
- 桥接模式 Bridge

行为型(Behavioral)设计模式使用频率总结

策略模式
Strategy

模板方法模式Template Method

访问者模式
Visitor

观察者模式
Observer

迭代器模式
Iterator

备忘录模式
Memento

责任链模式
Chain of Responsibility

中介者模式
Mediator

状态模式
State

解释器模式
Interpreter

命令模式
Command

一句话归纳创建型设计模式

设计模式	一句话归纳	目的	生活案例	框架源码举例
工厂模式（Factory）	产品标准化，生产更高效	封装创建细节	实体工厂	LoggerFactory、Calender
单例模式（Singleton）	世上只有一个Tom	保证独一无二	CEO	BeanFactory、Runtime
原型模式（Prototype）	拔一根猴毛，吹出千万个	高效创建对象	克隆	ArrayList、PrototypeBean
建造者模式（Builder）	高配中配与低配，想选哪配就哪配	开放个性配置步骤	选配	StringBuilder、BeanDefinitionBuilder

一句话归纳结构型设计模式

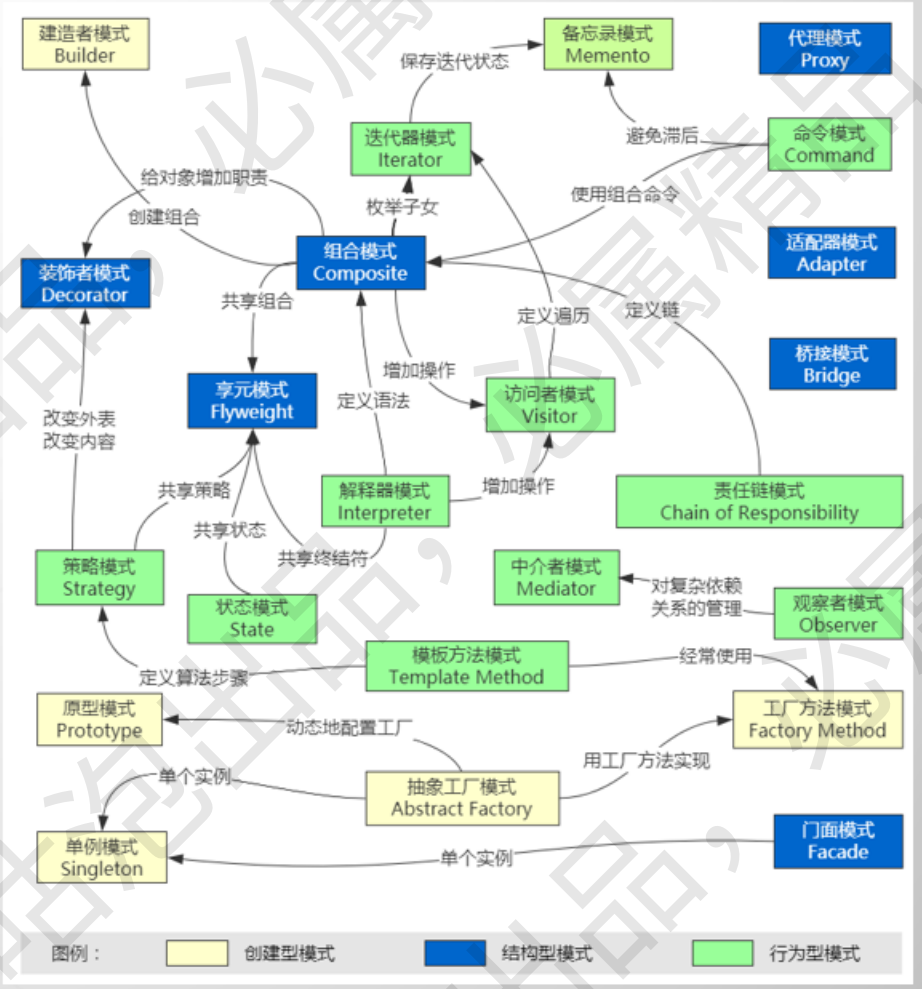
设计模式	一句话归纳	目的	生活案例	框架源码举例
代理模式（Proxy）	没有资源没时间，得找媒婆来帮忙	增强职责	媒婆	ProxyFactoryBean、JdkDynamicAopProxy、CglibAopProxy
门面模式(Facade)	打开一扇门，走向全世界	统一访问入口	前台	JdbcUtils、RequestFacade
装饰器模式（Decorator）	他大舅他二舅，都是他舅	灵活扩展、同宗同源	煎饼	BufferedReader、InputStream
享元模式(Flyweight)	优化资源配置，减少重复浪费	共享资源池	全国社保联网	String、Integer、ObjectPool
组合模式(Composite)	人在一起叫团伙，心在一起叫团队	统一整体和个体	组织架构树	HashMap、SqlNode
适配器模式（Adapter）	适合自己的，才是最好的	兼容转换(求同存异)	电源适配	AdvisorAdapter、HandlerAdapter
桥接模式(Bridge)	约定优于配置	不允许用继承	桥	DriverManager

咕泡学院 —gupaoedu—

一句话归纳行为型设计模式

设计模式	一句话归纳	目的	生活案例	框架源码举例
委派模式（Delegate）	这个需求很简单，怎么实现我不管	只对结果负责	授权委托书	ClassLoader、BeanDefinitionParserDelegate
模板模式（Template）	流程全部标准化，需要微调请覆盖	逻辑复用	把大象装进冰箱的步骤	JdbcTemplate、HttpServlet
策略模式（Strategy）	条条大道通北京，具体哪条你来定	把选择权交给用户	选择支付方式	Comparator、InstantiationStrategy
责任链模式(Chain of Responsibility)	各人自扫门前雪，莫管他人瓦上霜	解耦处理逻辑	踢皮球	FilterChain、Pipeline
迭代器模式(Iterator)	流水线上坐一天，每个包裹扫一遍	统一对集合的访问方式	统一刷脸进站	Iterator
命令模式(Command)	运筹帷幄之中 决胜千里之外	解耦请求和处理	遥控器	Runnable、TestCase
状态模式(State)	状态驱动行为，行为决定状态	绑定状态和行为	订单状态跟踪	Lifecycle
备忘录(Memento)	给我一剂“后悔药”	备份	草稿箱	StateManageableMessageContext
中介者(Mediator)	联系方式我给你，怎么搞定我不管	统一管理网状资源	朋友圈	Timer
解释器模式（Interpreter）	我想说“方言” 一切解释权归我所有	实现特定语法解析	摩斯密码	Pattern、ExpressionParser
观察者模式（Observer）	到点就通知我	解耦观察者与被观察者	闹钟	ContextLoaderListener
访问者模式（Visitor）	横看成岭侧成峰，远近高低各不同	解耦数据结构和数据操作	KPI考核	FileVisitor、BeanDefinitionVisitor

设计模式混用



咕泡学院
—gupaoedu—

教学质量服务监督与反馈邮箱

vip. feedback@gupaoedu. com

咕泡学院
—gupaoedu—

谢谢观看

Tom