

设计模式 - 设计模式伊始

寂然

强调一点：设计模式是不依赖语言的

系列课程内容体系

学习内容

篇章一：设计模式原则

设计模式原则是设计模式的基础，依据

七大设计模式原则，单一职责，接口隔离，里氏替换，，，

篇章二：UML类图&类的六大关系

画原理类图

篇章三：创建型模式（5）

单例模式，工厂模式，抽象工厂模式，建造者模式，原型模式

篇章四：结构型模式（7）

适配器模式，桥接模式，装饰者模式，组合模式，外观模式，代理模式，享元模式

篇章五：行为型模式（11）

模板模式，责任链模式，迭代器模式，，，，，

学习路线

提出案例需求 -> 设计模式 -> 原理类图 -> 重构案例需求 -> 注意事项 -> 框架源码分析

观看说明

本节内容目标

- 设计模式产生的背景&概念
- 为什么要学习设计模式?

设计模式的背景

设计模式最早出现在建筑领域，是克里斯托弗.亚历山大(Christopher Alexander) 对环境中不断出现的问题，总结

出这些问题的解决方案。以后再遇到这些问题时，可以重用这些方案来解决

四人帮 (GOF)

1994年，有四位作者：**Erich Gamma**，**Richard Helm**，**Ralph Johnson**和**John Vlissides** 发表了一本名为

《设计模式 - 可重用的面向对象软件元素》的图书，该书在软件开发中开创了设计模式的概念，第一次将设计模式

提升到理论高度，并将之规范化

四人帮



设计模式的概念

设计模式 (design pattern) 是对软件设计中普遍存在（反复出现）的各种问题，所提出的解决方案

这个术语是由埃里希·伽玛 (Erich Gamma) 等人在 1990 年代从建筑设计领域引入到计算机科学的

为什么要学设计模式

开发任务 --》实现功能 -》一周发一个版本

??

案例引入

茅草小屋



摩天大厦



总结

第一个，设计模式可以让你的程序体现出更好的代码重用性，可靠性，可读性，可维护性，可以让程序体现出高内聚，低耦合的特性

- 代码重用性：即相同功能的代码，一次编写，多处使用
- 可读性：即编码的规范性
- 可扩展性：即可维护性，当需要增加新功能的时候，成本非常低，可扩展性就非常高
- 可靠性：即代码健壮性，当需要增加新功能的时候，对原来的程序没有影响

高内聚&低耦合

- 内聚：主要描述的是模块内部
- 耦合：主要描述的是模块之间的关系
- 高内聚：一个功能模块内部的元素，关联性非常强
- 低耦合：模块之间耦合性非常低

第二个，掌握了设计模式，去看一些源码会非常清晰

第三个，设计模式也是系统合理重构的指南针

? 不改变原来软件的功能--》通过调整程序代码 --》趋于合理

设计模式原则引出

？何为设计模式的原则

- 程序员在编程的时候，应当遵守的准则
- 设计模式的基础，依据
- 不是孤立存在的，相互依赖的，相互补充的，每一种都很重要

名称	设计原则简介	重要性
单一职责原则	类的职责要单一，不能将太多的职责放在一个类中	★★★★☆
开闭原则	软件实体对扩展是开放的，但对修改是关闭的，即在不修改一个软件实体的基础上去扩展其功能	★★★★★
里氏代换原则	在软件系统中，一个可以接受基类对象的地方必然可以接受一个子类对象	★★★★☆
依赖倒置原则	要针对抽象层编程，而不要针对具体类编程	★★★★★
接口隔离原则	使用多个专门的接口来取代一个统一的接口	★★☆☆☆
合成复用原则	在系统中应该尽量多使用组合和聚合关联关系，尽量少使用甚至不使用继承关系	★★★★☆
迪米特法则	一个软件实体对其他实体的引用越少越好，或者说如果两个类不必彼此直接通信，那么这两个类就不应当发生直接的相互作用，而是通过引入一个第三者发生间接交互	★★★☆☆

下节预告

单一职责原则

