

<<漫谈设计模式>>

图书基本信息

书名：<<漫谈设计模式>>

13位ISBN编号：9787302273028

10位ISBN编号：7302273022

出版时间：2012-1

出版时间：清华大学出版社

作者：刘济华

页数：279

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<漫谈设计模式>>

### 内容概要

模式引入计算机科学领域已经有20余年了，最初人们侧重于面向对象的设计，而现在已经应用于计算机领域的各个方面，它们对计算机产生了深远影响。

若想一本书涵盖所有模式，那么这本书将会非常庞大，以前的设计模式书籍专注于介绍设计模式，虽然读者了解了这些设计模式，但是仍然不知道如何合理地使用它们，往往导致读者为了使用设计模式而设计，而不是从问题出发，使用设计模式优雅地解决这些问题。

本书主要从最基本的设计模式入手，并结合一些J2EE开发过程经常遇见的技术和概念，你将全面理解这10多个设计模式，并在开发过程中，让你真正体会和思考面向对象编程的思想，也只有掌握这些，你才会能成为一名真正的设计专家。

本书不是一本面向对象和Java语言的入门书籍，阅读对象主要是从事Java语言的软件开发人员但不限于Java语言的开发人员。

## <<漫谈设计模式>>

### 作者简介

刘济华现任惠普公司ITSM顾问，曾在电信、金融、保险等多个行业参与开发了多个大型项目和产品，参与过UMS系统、保险核心业务系统、银行系统、CRM系统、ITSM系统软件研发。热衷于编写优良代码以及研究如何提高开发效率，对面向对象开发与设计有较深层次理解，致力于使用OOP为复杂领域问题建模。喜欢钻研并应用一些出色的模式，以及支持软件高效开发设计的流程和方法，是领域驱动开发（Domain-Driven Design）和敏捷（Agile）开发方法的忠实粉丝。

个人博客：<http://redhat.iteye.com/>

## &lt;&lt;漫谈设计模式&gt;&gt;

## 书籍目录

## 第一篇 模式介绍

## 第1章 谈面向对象和模式

- 1.1 什么是对象
- 1.2 面向对象的好处
- 1.3 重用
- 1.4 模式简史
- 1.5 什么是模式
- 1.6 学习设计模式的一些常见问题
- 1.7 本章关键词

## 第2章 第1个模式——模板方法 (template method) 模式

- 2.1 从回家过年说起
  - 2.1.1 dry (don't repeat yourself)
  - 2.1.2 变化+重复, 如何维护
- 2.2 模板方法 (template method) 模式
  - 2.2.1 使用继承
  - 2.2.2 模板方法模式
- 2.3 引入回调 (callback)
- 2.4 总结
- 2.5 本章关键词

## 第二篇 创建对象

## 第3章 单例 (singleton) 模式

- 3.1 最简单的单例
- 3.2 进阶
  - 3.2.1 延迟创建
  - 3.2.2 线程安全
  - 3.2.3 double-check locking
  - 3.2.4 initialization on demand holder
  - 3.2.5 singleton的序列化
- 3.3 总结
- 3.4 本章关键词

## 第4章 工厂方法 (factory method) 模式

- 4.1 工厂方法模式
  - 4.1.1 类图
  - 4.1.2 创建数据库连接对象
- 4.2 静态工厂方法
- 4.3 总结
- 4.4 本章关键词

## 第5章 原型 (prototype) 模式

- 5.1 原型模式
- 5.2 寄个快递
- 5.3 实现
  - 5.3.1 uml静态类图
  - 5.3.2 代码实现
- 5.4 深拷贝 (deep copy)
- 5.5 总结

## &lt;&lt;漫谈设计模式&gt;&gt;

- 5.6 本章关键词
- 第6章 控制反转 (ioc)
  - 6.1 从创建对象谈起
  - 6.2 使用工厂方法模式的问题
  - 6.3 inversion of control (控制反转)
    - 6.3.1 ioc和di (dependency injection, 依赖注入)
    - 6.3.2 service locator (服务定位器)
    - 6.3.3 dependency injection
  - 6.4 总结
  - 6.5 本章关键词
- 第三篇 构建复杂结构
- 第7章 装饰器 (decorator) 模式
  - 7.1 记录历史修改
  - 7.2 open-closed principle (开放—封闭原则, ocp)
  - 7.3 装饰器 (decorator) 模式
    - 7.3.1 类图
    - 7.3.2 实现
    - 7.3.3 一点变化
    - 7.3.4 如何使用
    - 7.3.5 测试
  - 7.4 装饰器模式的优缺点
  - 7.5 总结
  - 7.6 本章关键词
- 第8章 代理 (proxy) 模式
  - 8.1 代理 (proxy) 模式
    - 8.1.1 类图
    - 8.1.2 访问分布式对象
  - 8.2 j2se动态代理
    - 8.2.1 类和接口
    - 8.2.2 调用原理
    - 8.2.3 实现同步
    - 8.2.4 总结
  - 8.3 和装饰器 (decorator) 模式的比较
  - 8.4 总结
  - 8.5 本章关键词
- 第9章 适配器 (adapter) 模式
  - 9.1 打桩
  - 9.2 其他适配器模式
    - 9.2.1 类适配器
    - 9.2.2 双向适配器
  - 9.3 测试

## &lt;&lt;漫谈设计模式&gt;&gt;

- 9.4 和代理 (proxy) 模式的比较
- 9.5 总结
- 9.6 本章关键词
- 第10章 外观 (facade) 模式
  - 10.1 外观 (facade) 模式
  - 10.2 least knowledge principle (最少知识原则)
  - 10.3 懒惰的老板请客
  - 10.4 ejb里的外观模式
  - 10.5 总结
  - 10.6 本章关键词
- 第11章 组合 (composite) 模式
  - 11.1 组合模式概述
    - 11.1.1 类图
    - 11.1.2 使用组合 (composite) 模式
    - 11.1.3 测试
  - 11.2 透明的组合模式
  - 11.3 安全的组合模式vs透明的组合模式
  - 11.4 还需要注意什么
  - 11.5 总结
  - 11.6 本章关键词
- 第四篇 行为模式
  - 第12章 策略 (strategy) 模式
    - 12.1 既要坐飞机又要坐大巴
    - 12.2 封装变化
    - 12.3 策略模式
    - 12.4 还需要继承吗
    - 12.5 优先使用合成而非继承
    - 12.6 总结
    - 12.7 本章关键词
  - 第13章 状态 (state) 模式
    - 13.1 电子颜料板
    - 13.2 switch-case实现
    - 13.3 如何封装变化
    - 13.4 状态模式
    - 13.5 使用enum类型
    - 13.6 与策略 (strategy) 模式的比较
    - 13.7 总结
    - 13.8 本章关键词
  - 第14章 观察者 (observer) 模式
    - 14.1 股票价格变了多少
    - 14.2 观察者模式
      - 14.2.1 如何实现
      - 14.2.2 观察者模式

## &lt;&lt;漫谈设计模式&gt;&gt;

14.2.3 java标准库的观察者  
模式

14.3 总结

14.4 本章关键词

第五篇 终点还是起点

第15章 面向切面的编程 ( aop )

15.1 记录时间

15.2 aop ( aspect-oriented  
programming )

15.2.1 一些重要概念

15.2.2 oop实现横切

15.2.3 aop实现技术

15.3 aop框架介绍

15.4 aop联盟  
( aop alliance )

15.5 使用aop编程的风险

15.6 oop还是aop

15.7 总结

15.8 本章关键词

第16章 面向对象开发

16.1 写在面向对象设计之前

16.2 汲取知识

16.3 横看成岭侧成峰

16.4 提炼模型

16.5 应用设计模式

16.6 不能脱离实现技术

16.7 重构

16.8 过度的开发 ( over-engineering )

16.9 总结

16.10 本章关键词

第17章 结语

17.1 感悟

17.2 面向对象的开发范式

17.3 一些原则

17.4 写在模式之后

17.5 本章关键词

附录 推荐阅读资源

1 java语言相关学习图书

2 j2ee技术相关图书

3 面向对象设计相关图书

4 给agile ( 敏捷 ) 开发人员推荐的书籍

5 网站和论坛

参考文献

## <<漫谈设计模式>>

### 章节摘录

版权页：插图：对象易于理解和抽象：例如马是一个类，一匹马是一个对象，跑是马的行为。

正是由于这个特性，我们很容易把客观世界反映到计算机里，极大地方便了编程设计。

对象的粒度更大，模块化程度也更高：与方法（函数）和结构体相比，对象是一组方法和数据的单元，所以粒度更大，这样更方便控制和使用；而模块化程度越高，也越容易抽象。

更加容易重用代码：只要使用继承，就可以拥有父类的方法；只要创建这个对象，就可以使用它们的公有属性和方法；只要使用多态，就可以使用相同的逻辑处理不同类型的对象。

具有可扩充性和开放性：OOP天生就具有扩展性和开放性。

代码易于阅读：在阅读代码过程中，可以不去关注那些具体实现类，只要关注接口的约定即可，这样更容易侧重重点。



## <<漫谈设计模式>>

### 编辑推荐

《漫谈设计模式:从面向对象开始》特色：从面向对象的眼光分析问题，并引入模式。

给出J2EE实例，这些例子都是J2EE应用中被经常使用的。

类比相近的模式，使读者能够掌握它们之间的区别与联系。

融入软件开发设计原则，指导读者在实践中正确使用这些原则。

如果您：反复研习面向对象编程，然而并未开发出真正面向对象的系统。

应用了面向对象模式，但并未享受到面向对象为软件带来的好处。

对模式和面向对象的理解停留在表面，不会使用面向对象的观点分析问题。

那么您应该阅读《漫谈设计模式:从面向对象开始》。

没有谈神马，也没有讲浮云，漫谈设计模式，也漫谈面向对象（OO），放弃过程式编程（FP/PP），玩转面向对象编程（OOP），辅以面向切面编程（AOP）不要if-else，不要重复自己（DRY）反转控制（IOC），反转依赖（DI），工单例模式，工厂方法模式，饰器模式，原型模式，爱编程，也爱琢磨，爱钻研，更爱分享，我是程序员，漫谈设计模式背后的软件设计思想。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>