

## <<设计模式的艺术>>

### 图书基本信息

书名：<<设计模式的艺术>>

13位ISBN编号：9787302296324

10位ISBN编号：7302296324

出版时间：2013-1

出版时间：清华大学出版社

作者：刘伟

页数：396

字数：619000

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>

<<设计模式的艺术>>

前言

## <<设计模式的艺术>>

### 内容概要

软件开发是一门技术，更是一门艺术。

设计模式是面向对象软件开发的入门功夫，是前人经验的积累，它为构建可维护性和可复用性俱佳的软件而诞生。

《设计模式的艺术：软件开发人员内功修炼之道》结合大量应用实例分析和讲解每一个设计模式，力求通过最通俗易懂的方式让读者学习和理解设计模式，并且在真实项目实例的引导下学会选择和合理运用设计模式。

《设计模式的艺术：软件开发人员内功修炼之道》分为4个部分，共26章内容：第1部分（第1章、第2章）为基础知识，包括设计模式概述、UML，类图与面向对象设计原则等；第2部分（第3~8章）为创建的艺术，包括6种常用的创建型设计模式；第3部分（第9~15章）为组合的艺术，包括7种常用的结构型设计模式；第4部分（第16~26章）为交互的艺术，包括11种常用的行为型设计模式。

《设计模式的艺术：软件开发人员内功修炼之道》可作为一线软件开发人员、高等院校计算机及软件等相关专业师生、IT培训机构讲师和学员、业余软件开发人员、设计模式研究人员以及爱好者的参考用书和自学读物。

## <<设计模式的艺术>>

### 作者简介

刘伟，中南大学计算机应用技术博士，高级工程师，国家认证系统分析师，首批国家认证系统架构设计师，数据库系统工程师，微软认证系统工程师，CSAI顾问团专业顾问及企业内训讲师，CSDN认证专家。

具有十余年软件开发、项目管理及教育培训经验，曾在NIIT（印度国家信息技术学院）担任高级讲师，主持和参与三十多个软件项目的开发工作，现主要致力于软件工程、数据挖掘等领域的教学、推广和研究工作。

发表相关论文多篇，已编写出版《设计模式》、《设计模式实训教程》、《设计模式的艺术——软件开发人员内功修炼之道》等著作。

## <<设计模式的艺术>>

### 书籍目录

#### 第1部分基础知识

##### 第1章从招式与内功谈起——设计模式概述

1.1设计模式从何而来

1.2设计模式是什么

1.3设计模式有什么用

1.4个人观点

1.5知识回顾

##### 第2章预备知识——UML类图与面向对象设计原则

2.1UML概述

2.2类与类的UML图示

2.3类之间的关系

2.4面向对象设计原则概述

2.5单一职责原则

2.6开闭原则

2.7里氏代换原则

2.8依赖倒转原则

2.9接口隔离原则

2.10合成复用原则

2.11迪米特法则

2.12知识回顾

#### 第2部分创建的艺术——创建型模式

##### 第3章确保对象的唯一性——单例模式

3.1单例模式的动机

3.2单例模式概述

3.3负载均衡器的设计

3.4饿汉式单例与懒汉式单例的讨论

3.5一种更好的单例实现方法

3.6单例模式总结

3.7知识回顾

##### 第4章集中式工厂的实现——简单工厂模式

4.1图表库的设计

4.2简单工厂模式概述

4.3完整解决方案

4.4方案的改进

4.5创建对象与使用对象

4.6简单工厂模式的简化

4.7简单工厂模式总结

4.8知识回顾

##### 第5章多态工厂的实现——工厂方法模式

5.1日志记录器的设计

5.2工厂方法模式概述

5.3完整解决方案

5.4反射与配置文件

5.5重载的工厂方法

5.6工厂方法的隐藏

## <<设计模式的艺术>>

5.7工厂方法模式总结

5.8知识回顾

第6章产品族的创建——抽象工厂模式

6.1界面皮肤库的初始设计

6.2产品等级结构与产品族

6.3抽象工厂模式概述

6.4完整解决方案

6.5开闭原则的倾斜性

6.6抽象工厂模式总结

6.7知识回顾

第7章对象的克隆——原型模式

7.1大同小异的工作周报

7.2原型模式概述

7.3完整解决方案

7.4带附件的周报

7.5原型管理器的引入和实现

7.6原型模式总结

7.7知识回顾

第8章复杂对象的组装与创建——建造者模式

8.1游戏角色设计

8.2建造者模式概述

8.3完整解决方案

8.4关于Director的进一步讨论

8.5建造者模式总结

8.6知识回顾

第3部分组合的艺术——结构型模式

第9章不兼容结构的协调——适配器模式

9.1没有源码的算法库

9.2适配器模式概述

9.3完整解决方案

9.4类适配器模式

9.5双向适配器模式

9.6缺省适配器模式

9.7适配器模式总结

9.8知识回顾

第10章处理多维度变化——桥接模式

10.1跨平台图像浏览系统

10.2桥接模式概述

10.3完整解决方案

10.4适配器模式与桥接模式的联用

10.5桥接模式总结

10.6知识回顾

第11章树形结构的处理——组合模式

11.1设计杀毒软件的框架结构

11.2组合模式概述

11.3完整解决方案

11.4透明组合模式与安全组合模式

## <<设计模式的艺术>>

11.5Sunny公司组织结构

11.6组合模式总结

11.7知识回顾

第12章扩展系统功能——装饰模式

12.1图形界面构件库的设计

12.2装饰模式概述

12.3完整解决方案

12.4透明装饰模式与半透明装饰模式

12.5装饰模式注意事项

12.6装饰模式总结

12.7知识回顾

第13章提供统一入口——外观模式

13.1文件加密模块的设计

13.2外观模式概述

13.3完整解决方案

13.4抽象外观类的引入

13.5外观角色设计补充说明

13.6外观模式总结

13.7知识回顾

第14章实现对象的复用——享元模式

14.1围棋棋子的设计

14.2享元模式概述

14.3完整解决方案

14.4带外部状态的解决方案

14.5单纯享元模式和复合享元模式

14.6关于享元模式的几点补充

14.7享元模式总结

14.8知识回顾

第15章对象的间接访问——代理模式

15.1收费商务信息查询系统的设计

15.2代理模式概述

15.3完整解决方案

15.4远程代理

15.5虚拟代理

15.6Java动态代理

15.7代理模式总结

15.8知识回顾

第4部分交互的艺术——行为型模式

第16章请求的链式处理——职责链模式

16.1采购单的分级审批

16.2职责链模式概述

16.3完整解决方案

16.4纯与不纯的职责链模式

16.5职责链模式总结

16.6知识回顾

第17章请求发送者与接收者解耦——命令模式

17.1自定义功能键

## &lt;&lt;设计模式的艺术&gt;&gt;

- 17.2命令模式概述
- 17.3完整解决方案
- 17.4命令队列的实现
- 17.5撤销操作的实现
- 17.6请求日志
- 17.7宏命令
- 17.8命令模式总结
- 17.9知识回顾
- 第18章自定义语言的实现——解释器模式
- 18.1机器人控制程序
- 18.2语法规则和抽象语法树
- 18.3解释器模式概述
- 18.4完整解决方案
- 18.5再谈Context的作用
- 18.6解释器模式总结
- 18.7知识回顾
- 第19章遍历聚合对象中的元素——迭代器模式
- 19.1销售管理系统中数据的遍历
- 19.2迭代器模式概述
- 19.3完整解决方案
- 19.4使用内部类实现迭代器
- 19.5JDK内置迭代器
- 19.6迭代器模式总结
- 19.7知识回顾
- 第20章协调多个对象之间的交互——中介者模式
- 20.1客户信息管理窗口的初始设计
- 20.2中介者模式概述
- 20.3完整解决方案
- 20.4中介者与同事类的扩展
- 20.5中介者模式总结
- 20.6知识回顾
- 第21章撤销功能的实现——备忘录模式
- 21.1可悔棋的中国象棋
- 21.2备忘录模式概述
- 21.3完整解决方案
- 21.4实现多次撤销
- 21.5再谈备忘录的封装
- 21.6备忘录模式总结
- 21.7知识回顾
- 第22章对象间的联动——观察者模式
- 22.1多人联机对战游戏的设计
- 22.2观察者模式概述
- 22.3完整解决方案
- 22.4JDK对观察者模式的支持
- 22.5观察者模式与Java事件处理
- 22.6观察者模式与MVC
- 22.7观察者模式总结



## <<设计模式的艺术>>

22.8知识回顾

第23章对象状态及其转换——状态模式

23.1银行系统中的账户类设计

23.2状态模式概述

23.3完整解决方案

23.4共享状态

23.5使用环境类实现状态转换

23.6状态模式总结

23.7知识回顾

第24章算法的封装与切换——策略模式

24.1电影票打折方案

24.2策略模式概述

24.3完整解决方案

24.4策略模式的两个典型应用

24.5策略模式总结

24.6知识回顾

第25章定义算法的框架——模板方法模式

25.1银行利息计算模块

25.2模板方法模式概述

25.3完整解决方案

25.4钩子方法的使用

25.5模板方法模式总结

25.6知识回顾

第26章操作复杂对象结构——访问者模式

26.1OA系统中员工数据汇总

26.2访问者模式概述

26.3完整解决方案

26.4访问者模式与组合模式联用

26.5访问者模式总结

26.6知识回顾

参考文献

## &lt;&lt;设计模式的艺术&gt;&gt;

## 章节摘录

版权页：插图：在开闭原则的定义中，软件实体可以指一个软件模块、一个由多个类组成的局部结构或一个独立的类。

任何软件都需要面临一个很重要的问题，即它们的需求会随时间的推移而发生变化。

当软件系统需要面对新的需求时，应该尽量保证系统的设计框架是稳定的。

如果一个软件设计符合开闭原则，那么可以非常方便地对系统进行扩展，而且在扩展时无须修改现有代码，使得软件系统在拥有适应性和灵活性的同时具备较好的稳定性和延续性。

随着软件规模越来越大，软件寿命越来越长，软件维护成本越来越高，设计满足开闭原则的软件系统也变得越来越重要。

为了满足开闭原则，需要对系统进行抽象化设计，抽象化是开闭原则的关键。

在Java、C#等编程语言中，可以为系统定义一个相对稳定的抽象层，而将不同的实现行为移至具体的实现层中完成。

在很多面向对象编程语言中都提供了接口、抽象类等机制，可以通过它们定义系统的抽象层，再通过具体类来进行扩展。

如果需要修改系统的行为，无须对抽象层进行任何改动，只需要增加新的具体类来实现新的业务功能即可，实现在不修改已有代码的基础上扩展系统的功能，达到开闭原则的要求。

在本书所要介绍的24种设计模式中，大部分设计模式都符合开闭原则，在对每一个模式进行优缺点评价时，都会将开闭原则作为一个重要的评价依据，以判断基于该模式设计的系统是否具备良好的灵活性和可扩展性。

2.7 里氏代换原则 里氏代换原则由2008年图灵奖得主、美国第一位计算机科学女博士Barbara Liskov教授和卡内基·梅隆大学Jeannette Wing教授于1994年提出。

其严格表述如下：如果对每一个类型为S的对象o1，都有类型为T的对象o2，使得以T定义的所有程序P在所有的对象o1都代换成o2时，程序P的行为没有变化，那么类型S是类型T的子类型。

这个定义比较拗口且难以理解，因此一般使用它的另一个通俗版定义：里氏代换原则表明，在软件中将一个基类对象替换成它的子类对象，程序将不会产生任何错误和异常，反过来则不成立，如果一个软件实体使用的是一个子类对象的话，那么它不一定能够使用基类对象。

例如，我喜欢动物，那我一定喜欢狗，因为狗是动物的子类；但是我喜欢狗，不能据此断定我喜欢动物，因为我并不喜欢老鼠，虽然它也是动物。

## <<设计模式的艺术>>

### 编辑推荐

《设计模式的艺术:软件开发人员内功修炼之道》编辑推荐：(1) 实例驱动：《设计模式的艺术:软件开发人员内功修炼之道》提供了30多个项目实例，每一个模式的讲解和练习都基于项目实例，让读者在项目实例的引导下学会正确选择和合理运用设计模式，真正理解和掌握每一个设计模式。

(2) 问题导入：《设计模式的艺术:软件开发人员内功修炼之道》的大多数设计模式都通过一个有问题的解决方案来导入，对使用模式之前的设计方案与使用模式重构之后的设计方案进行对比分析，加深对模式的理解。

(3) 结构严谨：《设计模式的艺术:软件开发人员内功修炼之道》每一章都从项目实例分析入手，逐步引出模式定义和结构，再提供一套应用模式的完整解决方案，然后对模式进行深入讲解，最后总结模式优缺点和适用场景，内容编排符合学习习惯，力求通过最通俗易懂的方式来讲解设计模式。

(4) 内容全面：对GoF 23个常用设计模式的讲解全面且细致，深入浅出，难度适中，可满足不同层次读者的需求。

(5) 边学边练：《设计模式的艺术:软件开发人员内功修炼之道》每一章都内嵌了适量的思考题和练习题，所有练习题也都基于项目实例，让读者能够在学习的同时通过思考题和练习题来消化和进一步巩固所学知识。

(6) 博客支持：《设计模式的艺术:软件开发人员内功修炼之道》作者在CSDN开辟了设计模式专栏博客，为广大读者提供一个灵活的沟通和交流平台，读者也可以很方便地与作者进行交互。

<<设计模式的艺术>>

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>