

21世纪高等学校计算机**专业**实用规划教材

# Java设计模式

耿祥义 张跃平 著

清华大学出版社

21世纪高等学校计算机**专业**实用规划教材

# Java设计模式

耿祥义 张跃平·著

清华大学出版社  
北京

## 内 容 简 介

本书面向有一定 Java 语言基础和一定编程经验的读者,重点探讨在 Java 程序设计中怎样使用著名的 23 个设计模式。本书的目的是让读者不仅学习怎样在软件设计中使用好设计模式,更重要的是让读者通过学习使用设计模式深刻地理解面向对象的设计思想,以便更好地使用面向对象语言解决设计中的诸多问题。

本书可以作为计算机相关专业高年级学生以及研究生的教材,也可以作为软件项目管理人员、软件开发工程师等专业人员的参考用书,可登录清华大学出版社网站([www.tup.tsinghua.edu.cn](http://www.tup.tsinghua.edu.cn))下载书中的示例代码。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

## 图书在版编目(CIP)数据

Java 设计模式/耿祥义,张跃平著. —北京:清华大学出版社,2009.5

(21 世纪高等学校计算机专业实用规划教材)

ISBN 978-7-302-19804-8

I. J… II. ①耿… ②张… III. JAVA 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 045644 号

责任编辑:魏江江 徐跃进

责任校对:李建庄

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者:北京市清华园胶印厂

装 订 者:三河市李旗庄少明装订厂

经 销:全国新华书店

开 本:185×260 印 张:18.5 字 数:444 千字

版 次:2009 年 5 月第 1 版 印 次:2009 年 5 月第 1 次印刷

印 数:1~4000

定 价:29.00 元

---

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:028646-01

## 相关课程教材推荐

ISBN	书 名	定价(元)
9787302177852	计算机操作系统	29.00
9787302178934	计算机操作系统实验指导	29.00
9787302177081	计算机硬件技术基础(第二版)	27.00
9787302176398	计算机硬件技术基础(第二版)实验与实践指导	19.00
9787302177784	计算机网络安全技术	29.00
9787302109013	计算机网络管理技术	28.00
9787302174622	嵌入式系统设计与应用	24.00
9787302176404	单片机实践应用与技术	29.00
9787302172574	XML 实用技术教程	25.00
9787302147640	汇编语言程序设计教程(第2版)	28.00
9787302131755	Java 2 实用教程(第三版)	39.00
9787302142317	数据库技术与应用实践教程——SQL Server	25.00
9787302143673	数据库技术与应用——SQL Server	35.00
9787302179498	计算机英语实用教程(第二版)	23.00
9787302180128	多媒体技术与应用教程	29.50
9787302185819	Visual Basic 程序设计综合教程(第二版)	29.50

以上教材样书可以免费赠送给授课教师,如果需要,请发电子邮件与我们联系。

## 教学资源支持

敬爱的教师:

感谢您一直以来对清华版计算机教材的支持和爱护。为了配合本课程的教学需要,本教材配有配套的电子教案(素材),有需求的教师可以与我们联系,我们将向使用本教材进行教学的教师免费赠送电子教案(素材),希望有助于教学活动的开展。

相关信息请拨打电话 010-62776969 或发送电子邮件至 [weijj@tup.tsinghua.edu.cn](mailto:weijj@tup.tsinghua.edu.cn) 咨询,也可以到清华大学出版社主页(<http://www.tup.com.cn> 或 <http://www.tup.tsinghua.edu.cn>)上查询和下载。

如果您在使用本教材的过程中遇到了什么问题,或者有相关教材出版计划,也请您发邮件或来信告诉我们,以便我们更好为您服务。

地址:北京市海淀区双清路学研大厦 A 座 708 室 计算机与信息分社魏江江 收

邮编:100084

电子邮件: [weijj@tup.tsinghua.edu.cn](mailto:weijj@tup.tsinghua.edu.cn)

电话:010-62770175-4604

邮购电话:010-62786544

# 出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机专业课程领域,以专业基础课为主、专业课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 反映计算机学科的最新发展,总结近年来计算机专业教学的最新成果。内容先进,充分吸收国外先进成果和理念。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,融合先进的教学思想、方法和手段,体现科学性、先进性和系统性,强调对学生实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。专业基础课和专业课教材配套,同一门课程有针对不同层次、面向不同应用的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机专业实用规划教材  
联系人:魏江江 weijj@tup.tsinghua.edu.cn

# 前 言

---

设计模式是从许多优秀的软件系统中总结出的成功的、可复用的设计方案,已经被成功应用于许多系统的设计中。目前,面向对象程序设计已经成为软件设计开发领域的主流,而学习使用设计模式无疑非常有助于软件开发人员使用面向对象语言开发出易维护、易扩展、易复用的代码。本书面向有一定 Java 语言基础和一定编程经验的读者,重点探讨在 Java 程序设计中怎样使用著名的“23 个设计模式”。本书的目的是让读者不仅学习怎样在软件设计过程中使用好设计模式,更重要的是让读者通过学习深刻地理解面向对象的设计思想,以便更好地使用面向对象语言解决设计中的诸多问题。

本书共 27 章,前 3 章为学习设计模式的一些必要知识,第 4 至第 26 章探讨、讲解“23 个设计模式”,最后一章为模式分类。第 1 章简单介绍模式的基本概念、起源以及学习设计模式的基本方法。第 2 章为面向对象的一些基本原则,了解这些基本原则,有助于理解后续章节中的模式。第 3 章简单介绍了 UML 中的类图。第 4 至第 26 章的每章都包含四部分重要的内容,第一部分为一个概述,用易于理解的问题讲解使用模式的动机;第二部分是模式的结构,在这部分用一个易于理解的示例讲解模式的结构;第三部分阐述模式的优点和使用场景;第四部分结合实际问题,使用设计模式给出一个有一定难度和实用价值的示例。

为了说明一个模式的核心实质,本书精心研究了针对每个模式的示例,以便让读者结合这样的示例能更好地理解和使用模式。本书的全部示例由作者编写完成,并在 JDK 1.6 环境下编译通过。可登录清华大学出版社网站([www.tup.tsinghua.edu.cn](http://www.tup.tsinghua.edu.cn))下载书中的示例代码,本书示例代码及相关内容仅供学习 Java 设计模式使用,不得以任何方式抄袭出版。

希望本书能对读者学习和使用设计模式有所帮助,并请读者批评指正。

编 者

2009 年 3 月

# 目 录

---

<b>第 1 章 设计模式简介</b> .....	1
1.1 什么是设计模式 .....	1
1.2 设计模式的起源 .....	2
1.3 “23 个设计模式”的来源 .....	2
1.4 学习设计模式的重要性 .....	2
1.5 合理使用模式 .....	3
1.6 什么是框架 .....	4
1.7 使用 Java 描述模式的必要性 .....	5
1.8 设计模式资源 .....	5
<b>第 2 章 面向对象的几个基本原则</b> .....	6
2.1 面向抽象原则 .....	6
2.2 开-闭原则 .....	10
2.3 多用组合少用继承原则.....	11
2.4 高内聚-低耦合原则 .....	12
<b>第 3 章 UML 类图简介</b> .....	13
3.1 类.....	13
3.2 接口.....	13
3.3 泛化关系.....	14
3.4 关联关系.....	14
3.5 依赖关系.....	15
3.6 实现关系.....	15
3.7 注释.....	15
<b>第 4 章 命令模式</b> .....	16
4.1 概述.....	16
4.2 模式的结构与使用.....	17
4.3 命令接口中的撤销方法.....	20
4.4 宏命令.....	22



4.5	命令模式的优点	25
4.6	适合使用命令模式的情景	25
4.7	命令模式与 Java AWT 事件	25
4.8	应用举例——模拟小电器	29
<b>第 5 章</b>	<b>观察者模式</b>	<b>34</b>
5.1	概述	34
5.2	模式的结构与使用	35
5.3	观察者模式中的“推”数据与“拉”数据	39
5.4	观察者与多主题	42
5.5	观察者模式的优点	46
5.6	适合使用观察者模式的情景	46
5.7	Java API 中的 Observable 类与 Observer 接口	47
5.8	应用举例——统计文本中的单词和数字	51
<b>第 6 章</b>	<b>装饰模式</b>	<b>55</b>
6.1	概述	55
6.2	模式的结构与使用	56
6.3	使用多个装饰者	60
6.4	装饰模式相对继承机制的优势	61
6.5	装饰模式的优点	62
6.6	适合使用装饰模式的情景	62
6.7	Java IO 与装饰模式	62
6.8	应用举例——读取单词表	63
<b>第 7 章</b>	<b>策略模式</b>	<b>66</b>
7.1	概述	66
7.2	模式的结构与使用	68
7.3	策略模式的优点	72
7.4	适合使用策略模式的情景	72
7.5	策略模式相对继承机制的优势	73
7.6	应用举例——加密、解密文件	73
<b>第 8 章</b>	<b>适配器模式</b>	<b>78</b>
8.1	概述	78
8.2	模式的结构与使用	80
8.3	适配器模式的优点	84
8.4	适合适配器模式的情景	84
8.5	单接口适配器	85

8.6	应用举例——Iterator 接口与 Enumeration 接口 .....	86
<b>第 9 章</b>	<b>责任链模式 .....</b>	<b>89</b>
9.1	概述 .....	89
9.2	模式的结构与使用 .....	90
9.3	责任链模式的优点 .....	93
9.4	适合使用责任链模式的情景 .....	94
9.5	应用举例——计算阶乘 .....	94
<b>第 10 章</b>	<b>外观模式 .....</b>	<b>98</b>
10.1	概述 .....	98
10.2	模式的结构与使用 .....	99
10.3	外观模式的优点 .....	102
10.4	适合使用外观模式的情景 .....	102
10.5	应用举例——解析文件 .....	102
<b>第 11 章</b>	<b>迭代器模式 .....</b>	<b>105</b>
11.1	概述 .....	105
11.2	模式的结构与使用 .....	106
11.3	迭代器的 next() 方法与集合的 get(int index) 方法 .....	108
11.4	迭代器模式的优点 .....	109
11.5	适合使用迭代器模式的情景 .....	109
11.6	应用举例——使用多个集合存储对象 .....	110
<b>第 12 章</b>	<b>中介者模式 .....</b>	<b>113</b>
12.1	概述 .....	113
12.2	模式的结构与使用 .....	114
12.3	中介者模式的优点 .....	118
12.4	适合使用中介者模式的情景 .....	118
12.5	使用中介者模式处理组件交互 .....	118
12.6	应用举例——模拟交通信号灯 .....	121
<b>第 13 章</b>	<b>工厂方法模式 .....</b>	<b>127</b>
13.1	概述 .....	127
13.2	模式的结构与使用 .....	128
13.3	Java 集合框架与工厂模式 .....	131
13.4	工厂方法模式的优点 .....	132
13.5	适合使用工厂方法模式的情景 .....	132
13.6	应用举例——创建药品对象 .....	132

<b>第 14 章 抽象工厂模式</b> .....	136
14.1 概述 .....	136
14.2 模式的结构与使用 .....	137
14.3 工厂方法模式的优点 .....	141
14.4 适合使用抽象工厂模式的情景 .....	141
14.5 应用举例——存款凭证 .....	142
<b>第 15 章 生成器模式</b> .....	149
15.1 概述 .....	149
15.2 模式的结构与使用 .....	150
15.3 生成器模式的优点 .....	153
15.4 适合使用生成器模式的情景 .....	154
15.5 应用举例——日历牌 .....	154
<b>第 16 章 原型模式</b> .....	160
16.1 概述 .....	160
16.2 java.lang.Object 类的 clone 方法 .....	161
16.3 Serializable 接口与克隆对象 .....	163
16.4 模式的结构与使用 .....	164
16.5 原型模式的优点 .....	166
16.6 适合原型模式的情景 .....	166
16.7 应用举例——克隆容器 .....	167
<b>第 17 章 单件模式</b> .....	170
17.1 概述 .....	170
17.2 模式的结构与使用 .....	170
17.3 单件模式的优点 .....	173
17.4 适合单件模式的情景 .....	173
17.5 应用举例——多线程争冠军 .....	173
<b>第 18 章 组合模式</b> .....	178
18.1 概述 .....	178
18.2 模式的结构与使用 .....	180
18.3 组合模式的优点 .....	184
18.4 适合组合模式的情景 .....	185
18.5 应用举例——苹果树的重量及苹果的价值 .....	185

<b>第 19 章 桥接模式</b> .....	190
19.1 概述 .....	190
19.2 模式的结构与使用 .....	191
19.3 桥接模式的优点 .....	194
19.4 适合桥接模式的情景 .....	195
19.5 应用举例——制作电视节目 .....	195
<b>第 20 章 状态模式</b> .....	200
20.1 概述 .....	200
20.2 模式的结构与使用 .....	202
20.3 状态切换 .....	204
20.4 共享状态 .....	207
20.5 状态模式的优点 .....	209
20.6 适合状态模式的情景 .....	210
20.7 应用举例——模拟咖啡自动售货机 .....	210
<b>第 21 章 模板方法模式</b> .....	215
21.1 概述 .....	215
21.2 模式的结构与使用 .....	216
21.3 钩子方法 .....	219
21.4 模板方法模式的优点 .....	222
21.5 适合模板方法模式的情景 .....	222
21.6 应用举例——数据库的连接与记录查询 .....	222
<b>第 22 章 代理模式</b> .....	228
22.1 概述 .....	228
22.2 模式的结构与使用 .....	229
22.3 远程代理 .....	231
22.4 虚拟代理 .....	235
22.5 代理模式的优点 .....	239
22.6 适合代理模式的情景 .....	239
22.7 应用举例——使用远程窗口阅读文件 .....	239
<b>第 23 章 享元模式</b> .....	242
23.1 概述 .....	242
23.2 模式的结构与使用 .....	243
23.3 享元模式的优点 .....	247
23.4 适合享元模式的情景 .....	247

23.5	应用举例——创建化合物 .....	248
<b>第 24 章</b>	<b>访问者模式 .....</b>	<b>251</b>
24.1	概述 .....	251
24.2	模式的结构与使用 .....	252
24.3	双重分派 .....	256
24.4	访问者模式的优点 .....	256
24.5	适合访问模式的情景 .....	256
24.6	应用举例——评价体检表 .....	256
<b>第 25 章</b>	<b>备忘录模式 .....</b>	<b>261</b>
25.1	概述 .....	261
25.2	模式的结构与使用 .....	262
25.3	使用内部类创建备忘录 .....	266
25.4	备忘录模式的优点 .....	266
25.5	适合使用备忘录模式的情景 .....	267
25.6	应用举例——使用备忘录实现 undo 操作 .....	267
<b>第 26 章</b>	<b>解释器模式 .....</b>	<b>270</b>
26.1	概述 .....	270
26.2	解释器模式的结构与使用 .....	271
26.3	解释器模式的优点 .....	272
26.4	适合使用解释器模式的情景 .....	272
26.5	应用举例——简单的英文翻译器 .....	273
<b>第 27 章</b>	<b>模式分类 .....</b>	<b>278</b>
27.1	创建型模式 .....	278
27.2	行为型模式 .....	278
27.3	结构型模式 .....	279
<b>参考文献</b>	<b>.....</b>	<b>280</b>

本书是面向有一定 Java 语言基础和一定编程经验的读者,本书重点是探讨在 Java 程序设计中怎样使用著名的“23 个设计模式”。<sup>[1]</sup>

本章简要介绍设计模式,包括设计模式的起源,GOF 著作的贡献以及设计模式与框架的区别。

## 1.1 什么是设计模式

一个设计模式是针对某一类问题的最佳解决方案,而且已经被成功应用于许多系统的设计中,它解决了在某种特定情景中重复发生的某个问题,因此,可以这样定义设计模式(pattern):“设计模式是从许多优秀的软件系统中总结出成功的可复用的设计方案”。文献[1]中引用了建筑大师 Alexander 关于设计模式的经典定义:“每一个设计模式描述一个在我们周围不断重复发生的问题,以及该问题的解决方案的核心。这样,你就能一次又一次地使用该方案而不必做重复劳动”。“尽管 Alexander 所指的是城市和建筑设计模式,但他的思想也同样适用于面向对象设计模式,只是在面向对象的解决方案里,我们用对象和接口代替了墙壁和门窗。两类模式的核心都在于提供了相关问题的解决方案”<sup>[1]</sup>。

记录一个设计模式需有四个基本要素<sup>[1]</sup>:

### 1. 名称

一个模式的名称高度概括该模式的本质,有利于该行业统一术语、便于交流使用。

### 2. 问题

描述应该在何时使用模式,解释设计问题和问题存在的前因后果,描述在怎样的环境下使用该模式。

### 3. 方案

描述设计的组成部分、它们之间的相互关系及各自的职责和协作方式。

### 4. 效果

描述模式的应用效果及使用模式应当权衡的问题。主要效果包括使用模式对系统的灵活性、扩充性和复用性的影响。

例如,中介者模式<sup>[1]</sup>:

- 名称 中介者。
- 问题 用一个中介者来封装一系列的对象交互。中介者使各对象不需要显式地相互引用,从而使其耦合松散,而且可以独立地改变它们之间的交互。
- 方案 中介者(Mediator)接口、具体中介者(Concrete Mediator)、同事(Colleague)、

具体同事(Concrete Colleague)。

- 效果 减少了子类的生成,将各个同事解耦,简化了对象协议,控制集中化。

## 1.2 设计模式的起源

软件领域的设计模式起源于建筑学。1977年,建筑大师 Alexander 出版了《A Pattern Language: Towns, Building, Construction》<sup>[2]</sup>一书, Alexander 在该著作中将其建筑行业中许多问题的最佳解决方案记录为 200 多种模式,这些模式为房屋与城市的建设制定了一些规则。Alexander 著作中的思想不仅在建筑行业影响深远,而且很快影响到了软件设计领域。1987年,受 Alexander 著作的影响, Kent Beck 和 Ward Cunningham 将 Alexander 在建筑学上的模式观点应用于软件设计,开发了一系列模式,并用 Smalltalk 语言实现了雅致的用户界面。Kent Beck 和 Ward Cunningham 在 1987 年举行的一次面向对象的会议上发表了论文《在面向对象编程中使用模式》,该论文发表后,有关软件的设计模式论文以及著作相继出版。

## 1.3 “23 个设计模式”的来源

目前,被公认在设计模式领域最具影响力的著作是 Erich Gamma、Richard Helm、Ralph Johnson 和 John Vlissides 在 1994 年合作出版的著作《Design Patterns: Elements of Reusable Object-Oriented Software》(中译本《设计模式:可复用的面向对象软件的基本原理》由机械工业出版社在 2000 年出版),该书的四位作者在其著作中记录了他们在四年多的工作中所发现的 23 个模式。四位作者的著作成为空前的畅销书,对软件设计人员学习、掌握和使用设计模式产生了巨大的影响。《设计模式》一书被广大喜爱者昵称为 GOF(Gang of Four)之书(四人帮之书),被认为是学习设计模式的必读著作,GOF 之书已经被公认为是设计模式领域的奠基之作。

自 GOF 的《设计模式》出版后,受其影响,陆续出版了许多具有一定影响力的书籍,比如,1998 年,Alpert、Brown 和 Woolf 出版《The Design Pattern Smalltalk Companion》<sup>[3]</sup>,该书使用 Smalltalk 语言讲解了 GOF 之书书中的 23 个模式;2000 年,James W. Cooper 出版《Java Design Patterns: A Tutorial》<sup>[4]</sup>,该书使用 Java 语言讲解了 GOF 之书书中的 23 个模式(中译本《Java 设计模式》由中国电力出版社在 2003 年出版),该书尤其侧重使用 GUI 程序设计来讲解怎样使用 GOF 之书书中的 23 个模式;特别要提到的是 Eric Freeman 等在 2004 年出版的《Head First Design Pattern》(中译本《Head First 设计模式》<sup>[5]</sup>由中国电力出版社在 2007 年出版),该书使用 Java 语言重点讲解 GOF 之书书中的部分模式(13 个模式),书中图文并茂,独具匠心的写作风格更是令人耳目一新,语言叙述及结构安排非常适合初学者;我们在参考文献部分还列出了部分具有一定影响力的有关设计模式的著作。

## 1.4 学习设计模式的重要性

一个好的设计系统往往是易维护、易扩展、易复用的,有过一定代码编写量的程序开发人员,可能会逐渐思考程序设计问题,想知道一些优秀的设计者或团队是怎样设计出一个好

的软件系统的。有经验的设计人员或团队知道如何使用面向对象语言编写出易维护、易扩展和易复用的程序代码,《设计模式》一书正是从这些优秀的设计系统中总结出的设计精髓,因此学习好设计模式对提高设计能力无疑是非常有帮助的。尽管 GOF 之书并没有收集全部的模式(这似乎是不可能的,也是不必要的),但所阐述的 23 种模式无疑是使用频率最高的模式。

设计模式的目的是不是针对软件设计和开发中的每个问题都给出解决方案,而是针对某种特定环境中通常都会遇到的某种软件开发问题给出的可重用的一些解决方案,因此学习设计模式不仅可以使我们用好这些成功的模式,更重要的是可以使我们更加深刻地理解面向对象的设计思想,非常有利于我们更好地使用面向对象语言解决设计中的问题。另外,学习设计模式对于进一步学习、理解和掌握框架是非常有帮助的,比如 Java EE 中就大量使用了《设计模式》一书中的模式,对于熟悉设计模式的开发人员,很容易理解这些框架的结构,继而很好地使用框架来设计他们的系统。《设计模式》一书所总结的成功模式不仅适合于面向对象语言,其思想及解决问题的方式也适合于任何和设计相关的行业,因此学习掌握设计模式无疑是非常有益的。

## 1.5 合理使用模式

不是软件的任何部分都需要套用模式来设计的,必须针对具体问题合理地使用模式。

### 1. 正确使用

当设计某个系统,并确认所遇到的问题刚好适合使用某个模式,就可以考虑使用该模式到自己的系统设计中,毕竟该模式已经被公认是解决该问题的成功方案,能使设计的系统易维护、可扩展性强、复用性好,而且这些经典的模式也容易让其他开发人员了解你的系统和设计思想。

### 2. 避免教条

模式不是数学公式、也不是物理定律、更不是软件设计中的“法律”条文,一个模式只是成功解决某个特定问题的设计方案,完全可以修改模式中的部分结构以符合自己的设计要求。

### 3. 模式挖掘

模式不是用理论推导出来的,而是从真实世界的软件系统中被发现、按着一定规范总结出来的可以被复用的方案。目前,许多文献或书籍里阐述的众多模式实际上都是 GOF 书中经典模式的变形,这些变形模式都经过所谓的“三次规则”,即该模式已经在真实世界的三个方案中被成功地采用。也可以从某个系统中洞察出某种新模式,只要经过“三次规则”就会被行业认可。需要注意的是,在寻找新的模式之前,必须先精通现有的模式,尤其是 GOF 之书 23 个模式,因为许多模式事实上只是现有模式的变种。通过研究学习现有的模式,不仅可以比较容易地识别模式,而且也能使你学会怎样综合地使用各种模式,即使用复合模式。如果认为自己真的发现了一种新的模式,那么就可以按着 GOF 书中提供的格式将你的“准模式”写成一份文档,按 1.1 节中给出的模式定义,该文档至少需要包括名称、问题、方案和效果四个方面,当然“准模式”需要经过“三次规则”才能成为真正的模式。



#### 4. 避免乱用

不是所有的设计中都需要使用模式,因为模式不是发明出来的,而是总结出来的,事实上,真实世界中的许多设计实例都没有使用过 GOF 书中的经典模式。在进行设计时,尽可能用最简单的方式满足系统的要求,而不是费尽心机地琢磨如何在这个问题中使用模式,一个设计中,可能并不需要使用模式就可以很好地满足系统的要求,如果牵强地使用某个模式可能会在系统中增加许多额外的类和对象,影响系统的性能,因为大部分设计模式往往会在系统中加入更多的层,这不但会增加复杂性,而且系统的效率也会下降。

#### 5. 了解反模式

所谓反模式就是从某些软件系统中总结出的不好的设计方案,反模式就是告诉你如何采用一个不好的方案解决一个问题。既然是一个不好的方案,为何还有可能被重复使用呢?这是因为,这些不好的方案表面上往往有很强的吸引力,人们很难一眼就发现它的弊端,因此,发现一个反模式也是非常有意义的工作。在有了一定的设计模式的基础之后,可以用搜索引擎查找有关反模式的信息,这对学习好设计模式也是非常有帮助的。

## 1.6 什么是框架

框架不是模式,框架是针对某个领域,提供用于开发应用系统的类的集合,程序设计者可以使用框架提供的类设计一个应用程序,而且在设计应用程序时可以针对特定的问题使用某个模式。

### 1. 层次不同

模式比框架更抽象,模式是在某种特定环境中,针对一个软件设计出现的问题而给出的可复用的解决方案,不能向使用者提供可以直接使用的类,设计模式只有在被设计人员使用时才能表示为代码,例如,GOF 描述的中介者模式:“用一个中介对象来封装一系列的对象交互。中介者使各对象不需要显式地相互引用,从而使其耦合松散,而且可以独立地改变它们之间的交互”,该模式在解决方案中并没有提供任何类的代码,只是说明设计者可以针对特定的问题使用该模式给出的方案。框架和模式不同,它不是一种可复用的设计方案,它是由可用于设计解决某个问题的一些类组成的集合,程序设计人员通过使用框架提供的类或扩展框架提供的类进行应用程序的设计,例如,在 Java 中,开发人员使用 Swing 框架提供的类设计用户界面;使用 Set(集合)框架提供的类处理数据结构相关的算法等。

### 2. 范围不同

模式本质上是逻辑概念,以概念的形式而存在,模式所描述的方案独立于编程语言。Java 程序员、C++ 程序员或 SmallTalk 程序员都可以在自己的系统设计中使用的某个模式。框架应用的范围是很具体的,它们不是以概念的形式而存在,而是以具体的软件组织而存在,只能被特定的软件设计者使用,比如 Java 提供的 Swing 框架只能为 Java 应用程序所使用。

### 3. 相互关系

一个框架往往会包括多个设计模式,它们是面向对象系统获得最大复用的方式,较大的面向对象应用会由多层彼此合作的框架组成,例如,Java Web 设计中的 Struts、Spring 和 Hibernate 等框架。框架变得越来越普遍和重要,导致许多开源框架的出现,而且一个著名