Java

软件体系结构设计模式标准指南

[美] Partha Kuchana 著

王卫军 楚宁志 等译
吕 军 审校

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Software Architecture Design Patterns in Java

Java 软件体系结构设计模式标准指南

[美] Partha Kuchana 著
王卫军 楚宁志 等译
吕 军 审校

電子工業出版社
Publishing House of Electronics Industry
北京 • BEIJING

内 容 简 介

本书全面介绍了常用的42个设计模式，其中包括23个经典的GoF模式。新增并常用的19个新模式给读者带来更多的参考价值。本书的程序实例均采用Java语言，并且在Internet上可以获得源代码。本书通篇采用了UML标准图表作为描述工具，使得程序代码、设计模式以及文字说明能很好地融为一体。本书的最后一章为"案例研究"，作者把多个模式放在一个实际的应用场合里一起配合工作，这样的安排使读者能更好地对各种模式综合运用。本书内容深入浅出、清楚易懂，大量的Java程序实例和UML图表使内容更加清晰且更具可操作性。

本书适合软件开发和设计人员使用。

版权贸易合同登记号　图字：01-2005-4178

图书在版编目（CIP）数据

Java软件体系结构设计模式标准指南/（美）库察那（Kuchana，P.）著；王卫军等译.—北京：电子工业出版社，2006.2

（SUN核心技术丛书）

书名原文：Software Architecture Design Patterns in Java

ISBN 7-121-01873-X

Ⅰ.J…　Ⅱ.①库…②王…　Ⅲ.Java语言—程序设计　Ⅳ.TP312

中国版本图书馆CIP数据核字（2006）第006565号

责任编辑：朱巍

印　　刷：北京天竺颖华印刷厂

出版发行：电子工业出版社

北京市海淀区万寿路173信箱　邮编：100036

北京市海淀区翠微东里甲2号　邮编：100036

经　　销：各地新华书店

开　　本：787×1092　1/16　印张：23.25　字数：580千字

印　　次：2006年2月第1次印刷

定　　价：37.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077。质量投诉请发邮件至zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

前　言

Partha Kuchana 是一位富有经验的企业级系统体系结构设计师。他体会到模式不仅仅是好的想法，更是对来自实践的知识的俘获。这个来之不易的认识正是作者想和本书的读者一起分享的。下面我想谈谈我对他将要在本书中介绍给读者的内容的喜欢之处。

本书给出了包括 23 个 GoF 模式在内的 42 个设计模式。这些模式分类如下：

7 个基本模式；

5 个创建模式；

4 个集合模式；

11 个结构模式；

11 个行为模式；

4 个并发模式。

针对每个模式的讨论都包括一个用 Java 语言实现的示例程序，而且这些示例程序的源代码在下面这个网址处可以找到：http://www.crcpress.com/e_products/downloads/download.asp。这样的安排还是很不错的。

Partha 选取通常难以理解的素材，却清楚地解释了其中的思想使之易于理解。这样的考虑很重要，它兼顾了初次接触这些材料的新手和那些只想从中迅速获取“马上就能拿来用的”要点的有经验开发人员。针对每个模式的讨论还包括习题部分。这部分主要是供个人提高之用，也适用于那些用本书做教材而时间又比较紧的教员。

Partha 在比较模式之间的异同方面着实花了些时间。例如，在讨论调停者模式时，他给出了一个表格以显示这个模式和外观模式的相似之处和不同之处。读者会发现这种分析比孤立地讲解每个模式更容易理解。文中还有介绍不同模式间关系的内容。

例如，在讨论调停者模式时，就有一段与前面章节中介绍过的命令模式有关的设计用例。

最后，在本书的结尾处，读者会欣喜地发现一个实例研究部分。这部分把一些模式放在一起以说明如何解决一个复杂的问题，以及如何把不同的模式放在一起工作。如果学习过 Christopher Alexander 的成果，读者就会认识到模式并非孤立地被使用，而是在一个特定的问题域中相互协同来解决各种问题的。

自 GoF 的书出版至今已有 10 年之久了。这个期间有很多的模式被确认和研究，也出现了很多这方面的书。本书与 GoF 的书类似，也像是一本目录手册，读者不大可能一次就从头读到尾，但它将会是读者需要经常查阅的一本书。当读者有了很多像“利用 Java 语言这件事该如何做”这样的问题时，一定希望身边有一个专家来回答问题吧。本书将会起到和专家同样的作用。

Linda Rising

Phoenix，AZ

献　辞

谨以此书献给我的家人

作者简介

Partha Kuchana 是一位富有经验的企业级系统体系结构设计师。他有 11 年的从业经验，涉及项目交付管理(本土或离岸方式)和企业级系统体系结构搭建、系统设计、开发、顾问指导以及培训的各个方面。他还是一位 Sun 认证的企业级体系结构设计师。

在过去的几年里，他在英国和美国的许多客户那里从事过大量客户-服务器、电子商务、Web 门户以及企业应用集成(EAI)方面的项目。这些项目涉及了迭代设计方法学，比如 Rational 统一过程(RUP)和极限编程。

在将设计模式应用于应用程序的体系结构搭建和设计方面他有着广泛的经验。他利用来自不同厂商的 Web 服务、中间件和消息通信产品成功地实现了 B2B 系统和复杂异构系统集成的体系结构搭建和系统的设计。另外，他还出版了几本软件方面的书。

主页：http://members. ITJobsList. com/partha

电子邮件：ParthaKuchana@ITJobsList. com

致　谢

首先感谢我的妻子，感谢她的耐心与支持；感谢她在 UML 和 Java 编程方面承担了我的一部分工作；感谢她在我沮丧时给予的鼓舞。还要感谢我的父母、姐姐、兄弟以及我亲爱的朋友们，你们在我生活中始终给予的支持与鼓励让我掌握了获得成功所需要的各种技能。

感谢 Venu Kuchana 和 D. R. Sudhakar，因为他们贡献了不同的 Java 程序。还要感谢 BalaLingam Kuchana，他在 UML 方面给了我帮助，而且负责书稿的排版工作。

感谢在 Auerbach 出版社的所有同仁。他们对这个项目做出了贡献，并且使之成为了一段不同寻常的经历。特别地，我对参考资料编辑 John Wyzalek 怀有深深的感激，因为他不仅和我分享了忘我的工作，而且提供了很好的建议与帮助。还要感谢主编 Claire Miller 女士给予的宝贵建议，她的安排组织使得本书有着不错的篇章结构。

真诚地感谢 Linda Rising 为本书写了前言。

真诚地感激和感谢下面这些书评撰写人，他们花费宝贵的时间阅读了书稿并提供了反馈信息。

- Pradyumn Sharma，CEO，Pragati Software Pvt. Ltd.
- Carsten Kuckuk，项目带头人，Design Patterns Study Group Stuttgart，RIB Software AG
- Tim Kemper，Boulder Design Patterns Group
- Geoffrey Sparks，CEO，Sparx Systems P/L
- Edward L. Howe，软件体系结构设计师，Employease，Inc.
- Christopher R. Gardner，软件开发工程师，McKesson Information Solutions
- David Deriso，资深软件工程师，Employease，Inc.
- Mike Heinrich，软件工程师，Canada
- Rodney Waldoff，系统体系结构主管，Encyclopedia Brittanica Inc.
- Thomas SMETS，软件工程师，Belgium
- Linda Rising 博士，独立软件顾问，Arizona State University
- Ray Tayek，项目协调人，LAJUG/OCJUG

特别感谢 Pradyumn Sharma、Carsten Kuckuk 和 Tim Kemper，感谢他们提供的很有深度的建议、他们的细致入微以及他们的宝贵建议，其中包括对设计模式有强烈好奇心的读者可能会提的那些疑难问题。

另外还要感谢 Mark Grand 在有关模式书籍书写的很多方面给予的鼓励和建议。

相信在这里我还是漏掉了很重要的一些人，如果是那样，请见谅为盼！

译 者 序

本书全面介绍了软件开发和设计人员在工作和学习中常用的42个设计模式。

也许读者会有这样的疑问，近几年有关设计模式的书面世的已经很多了，为什么还要出版关于同样题材的书呢？译者认为本书有如下特点值得一提。

第一，本书包含了除23个经典的GoF模式之外的许多实用模式。新增的模式有：7个基本模式、4个集合模式、4个结构模式和4个并发模式。这些新增和常用的模式无疑会给读者提供更多的参考价值。

第二，本书的示例程序均采用Java语言，并且可以在Internet上获得源代码。由于经典的GoF原著是在Java尚未出现和流行时出版的，因此原书中的程序代码采用的仅仅是C++。本书弥补了这一缺憾。

第三，本书通篇采用UML标准图表作为描述工具，使得程序代码、设计模式以及文字说明能很好地融为一体；而且，本书还专门用一章的篇幅对UML统一建模语言进行介绍，方便了对UML并不十分了解的一部分初学者使用。

第四，作者在本书中花了很多精力对不同但相关或相似的模式进行比较或对照。这样的安排使得读者能更牢固地掌握每个设计模式的本质思想。

第五，本书的最后一章为“案例研究”，作者把多个模式放在一个实际的应用场合里一起配合工作。这样的安排使读者能更好地对各种模式综合运用。

第六，几乎在每一章中，作者都给出了习题，以加强读者对所学内容的理解和对模式的实际使用能力。因此，从这个角度上看，这本书更像是一本教科书。

参加本书翻译工作的有Sun中国工程研究院的王卫军、楚宁志、李志宏、曲靖、邹林志、孙阳和吕军，由吕军负责协调和统稿。翻译工作得到了王垦耀院长、交流对外合作部佟佳珏经理和项目经理荣文铮的大力支持和帮助，在此一并表示感谢。由于时间和水平有限，书中难免出现错误或笔误，恳请专家和读者不吝指正。

目　录

第一部分　设计模式简介

第二部分　UML

第三部分　基 本 模 式

第四部分 创建模式

第五部分 集合模式

第六部分 结构模式

第七部分 行 为 模 式

第八部分 并发模式

第九部分 案例研究

第一部分　设计模式简介

第 1 章　设计模式的起源与历史

在 20 世纪 70 年代后期，一个叫 Christopher Alexander 的建筑师在建筑模式领域开展了现在所知最早的一项工作。有一次在试图发现和描绘一些高质量建筑设计的完整性和其中体现的艺术活力时，Alexander 和他的同事们研究了为解决同一个问题而设计出的不同建筑结构，他发现了那些高质量的设计中的相似性。于是，在下面这些书中他用“模式”这个术语来指代这种相似性。

- 一种模式语言：城市、建筑业与建筑物（牛津大学出版社，1977 年）
- 永久的建筑风格（牛津大学出版社，1979 年）

由 Alexander 发现并记录的这些模式仅仅是涉及诸如建筑物、花园和道路之类的建筑学方面的模式。

从建筑学模式到软件设计模式

1987 年，受 Alexander 作品的影响，Kent Beck 和 Ward Cunningham 把建筑学上的模式观点应用于软件的设计和开发。他们利用 Alexander 的一些观点开发了一系列模式，利用 Smalltalk 语言实现了雅致的用户界面。利用这个工作成果，他们在 1987 年召开的面向对象的编程系统、语言和应用程序（OOPSLA）研讨会上做了一个以《在面向对象编程中使用模式语言》为题的演讲。从那以后，面向对象领域中的许多名人发表了很多与模式相关的论文和演讲。

1994 年，由 Erich Gamma、Richard Helm、Ralph Johnson 和 John Vlissides 合作的以《设计模式：可复用的面向对象软件的基本原理》为题目的书籍出版了。这本书解释了模式的用处，同时也使得设计模式得到广泛的普及。这四位作者被合称为“四人小组”（GoF）。在书中，作者们记录了在他们近四年半的工作中发现的 23 个模式。

从那儿以后，许多涉及设计模式和其他软件工程最佳实践方面的书陆续出版。

何谓设计模式

一个设计模式就是一个已被记录的最佳实践或一个解决方案，这个最佳实践或解决方案已被成功应用在许多环境中，它解决了在某种特定情境中重复发生的某个问题。

建筑师 Christopher Alexander 把一个建筑模式描绘成是“针对在某个特定背景和作用力系统中发生的通用问题的常见解决方案”。在这个定义中，术语“背景”指的是一个给定模式能够适用的条件/情境；而作用力系统指的是在这个特定背景下的约束条件集合。

关于设计模式的更多定义

- 设计模式是表达或传达从高质量的设计中所得到的学习成果的一种有效途径。这里

的学习成果是：

——一种共享语言，它用来传达在处理和对待那些常见问题及其解决方案方面获得的经验；

——一种为讨论问题解决方案时方便之用而在系统设计基本原理中采用的通用词汇；一种复用已获得认识并基于这种认识构建系统的方式，这种方式能在软件可维护性和可复用性方面带来质量上的提高。

- 一个设计模式并非一项新发明，而是一种文档化的表达方式，用来记录通过在对大量的软件系统的研究学习和构建过程中观察或发现得出的针对某个具体问题的最佳解决方法。
- 对于设计模式的若干常见错误看法中，有一种看法认为设计模式仅适用于面向对象的环境。事实上，尽管关于设计模式的讨论在典型情况下指的是面向对象的软件开发，但在其他领域它们同样适用。如果对其做一些小的变化，一个对设计模式的具体描述可以指代某些通用的软件设计模式。从前面讲到的设计模式的起源与历史中可以看出，模式在建筑业的早期已经存在，而且远远早于面向对象的设计和面向对象的编程时代。
- 设计模式并非理论构想。一个设计模式可以被看做一个对于一种可复用的解决方案的包装，而被包装的解决方案已被成功地用来解决一类常见的设计问题。
- 尽管设计模式指的是已知的解决很多问题的最好方式，然而并非所有的这些最佳实践都被认为是模式。一个最佳实践必须满足所谓“三次规定”才能被认为是一个设计模式。“三次规定”指出：一个给定的解决方案必须被验证是一个重复出现的现象，最好至少已经在三个已知系统中存在。否则，这个解决方案不会被认为是一个模式。这么做的目的是保证社区中的软件专业人员确实将某个模式中描述的解决方案应用在了实际的软件设计问题中。满足“三次规定”意味着一个设计模式确实是针对现实世界中的问题而给出的实用解决方案。
- 设计模式并不能对现实世界中在软件设计和开发中发现的每个问题都给出解决方案，而是针对某种“特定上下文”中通常都会遇到的某种软件开发问题而给出的简练和可重用的一些解决方案。这就是说，某个设计模式在某种“特定上下文”中能给出最佳解决方案，然而在“不同的上下文”中可能并不能给出一种有效的解决方案。有些情况下，某个设计模式中提出的解决方案可能在一个不同的上下文中根本无法适用。

软件框架（Frameworks）和设计模式易被混淆。它们两者是紧密相关的。表 1.1 列出了两者的相似点和不同点。

表 1.1　设计模式与框架

设计模式	框架
设计模式是在某种特定上下文中，针对一个软件应用程序生命过程中出现的问题而给出的多次适用的解决方案	一个框架是一组软件组件，它们互相协作提供了针对某个给定的问题领域中的应用程序所用到的一种可复用的体系结构
根本目标是： • 用来帮助提高以下方面的软件质量：可复用性、可维护性、可扩展性等 • 节省开发时间	根本目标是： • 用来帮助提高以下方面的软件质量：可复用性、可维护性、可扩展性等 • 节省开发时间

（续表）

设计模式	框架
模式本质上是逻辑概念	框架本质上更为具体可见，它们作为软件的形式而存在
模式描述通常是独立于编程语言或具体实现的	由于是作为一种软件的形式而存在，框架是针对具体实现的
模式本质上更为一般笼统，可在几乎任何应用程序中应用	框架提供的是与特定领域相关的功能
一个模式本身并不是以软件组件的形式存在的，它需要在每次使用时被具体实现	框架本身并不是完整的应用程序，然而完整的应用程序可以通过直接继承这些软件组件来构建
模式提供了一种方式用来完成一个“良好的”设计，而且模式也可用来辅助设计框架	设计模式可在一个框架的设计和实现过程中运用，或者说框架是一些设计模式的典型体现

关于本书

本书的目的在于通过轻松易懂的方式结合简单实例来讨论设计模式。本书讨论了包括 GoF 提出的 23 个模式在内的 42 个设计模式。这些模式分为 6 类：

- 7 个基本模式——第三部分（第 3 章到第 9 章）；
- 5 个创建模式——第四部分（第 10 章到第 14 章）；
- 4 个集合模式——第五部分（第 15 章到第 18 章）；
- 11 个构造模式——第六部分（第 19 章到第 29 章）；
- 11 个行为模式——第七部分（第 30 章到第 40 章）；
- 4 个并发模式——第八部分（第 41 章到第 44 章）。

关于每个模式的讨论都以关于这个模式的一段说明开始，后面紧跟一个用 Java 编程语言实现的例子。一个给出的模式如何被应用在这个实例中是结合代码段和 UML 图表（类图、序列图）来具体讨论的。在针对每个模式的讨论结束时，书中给出了几个习题以便加强读者对这个模式的理解。只要有可能，书中对相似的模式进行了比较。

本书中的实例简单易懂，目的是通过实例来强化针对每个模式的解释以使读者更容易理解。

UML 章节部分给出了统一建模语言的概述，并讨论了类图和序列图中的各种元素。

书末的实例学习部分展示了在现实世界中的应用程序设计场景中不同设计模式的整体综合运用。这一部分讨论了如何使用模式来为一个假想的 Web hosting 公司设计一个可重用的应用程序框架。

源代码

本书中的所有实例程序源代码在以下网址可以得到：

http：//www. crcpress. com/e _ products/downloads/default. asp

源代码声明

作者和出版商未做关于软件适用性方面的任何明指或暗指的表示或保证，其中包括并且不限于暗指的销路方面的保证、针对某种特定目的的适用性或不侵权。作者和出版商均不对使用、修改或分发软件或其衍生软件所带来的破坏性后果负责。

Java 是 Sun Microsystems 公司的商标。Windows 是 Microsoft 公司的注册商标。

第二部分　UML

OMG 是一家为企业应用制定计算机工业标准和规范的非赢利组织。UML 是 OMG 的一个应用建模规范。UML 的主要目的就是简化复杂的软件工程流程。使用 UML，人们能够详细地描叙、可视化和建立软件或非软件系统的模型。但是必须注意的是，UML 只是一种建模语言，就是说，它只定义了一些语言和语法，但是并没有去强制规定一些建模的流程。