

*J2EE Design Patterns*



# J2EE<sup>TM</sup>

# 设计模式

O'REILLY®  
中国电力出版社

*William Crawford & Jonathan Kaplan* 著

刘绍华 毛天露 译



# J2EE 设计模式

OPENUP  
中国好书

中国好书  
中国好书

---

# J2EE™ 设计模式

*William Crawford & Jonathan Kaplan* 著

刘绍华 毛天露 译

**O'REILLY®**

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权中国电力出版社出版

中国电力出版社

## 图书在版编目 (CIP) 数据

J2EE™ 设计模式 / (美) 克劳福德 (Crawford, W.), 卡普兰 (Kaplan, J.) 著; 刘绍华, 毛天露译. - 北京: 中国电力出版社, 2004

(O'Reilly Java 系列)

书名原文: J2EE™ Design Patterns

ISBN 7-5083-3099-4

I. J... II. ①克... ②刘... ③毛... III. JAVA 语言 - 程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 139223 号

北京市版权局著作权合同登记

图字: 01-2004-5049 号

©2003 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2004. Authorized translation of the English edition, 2003 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2003。

简体中文版由中国电力出版社出版 2004。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

书 名/ J2EE™ 设计模式

书 号/ ISBN 7-5083-3099-4

责任编辑/ 陈维宁

封面设计/ Hanna Dyer, 张健

出版发行/ 中国电力出版社 (www.infopower.com.cn)

地 址/ 北京三里河路 6 号 (邮政编码 100044)

经 销/ 全国新华书店

印 刷/ 北京丰源印刷厂

开 本/ 787 毫米×1092 毫米 16 开本 23 印张 373 千字

版 次/ 2005 年 4 月第一版 2005 年 10 月第二次印刷

印 数/ 3501-5500 册

定 价/ 45.00 元 (册)

# 目录

|                     |    |
|---------------------|----|
| 前言 .....            | 1  |
| 第一章 Java 企业设计 ..... | 9  |
| 设计模式 .....          | 9  |
| J2EE .....          | 12 |
| 应用层 .....           | 15 |
| 核心开发概念 .....        | 18 |
| 展望 .....            | 24 |
| 第二章 统一建模语言 .....    | 25 |
| UML 的起源 .....       | 26 |
| “七武士” .....         | 27 |
| UML 与软件开发生命周期 ..... | 28 |
| 用例图 .....           | 29 |
| 类图 .....            | 32 |
| 交互图 .....           | 38 |
| 活动图 .....           | 41 |
| 部署图 .....           | 42 |

|                           |            |
|---------------------------|------------|
| <b>第三章 表达层体系结构 .....</b>  | <b>44</b>  |
| 服务器端表达层 .....             | 45         |
| 应用结构 .....                | 46         |
| 建立一个集中控制器 .....           | 56         |
| <b>第四章 高级表达层设计 .....</b>  | <b>72</b>  |
| Web 应用中的重用 .....          | 73         |
| 扩展控制器 .....               | 73         |
| 高级视图 .....                | 84         |
| <b>第五章 表达层可伸缩性 .....</b>  | <b>98</b>  |
| 可伸缩性与瓶颈 .....             | 99         |
| 内容缓存 .....                | 100        |
| 资源池 .....                 | 113        |
| <b>第六章 业务层 .....</b>      | <b>121</b> |
| 业务层 .....                 | 122        |
| 领域对象 .....                | 126        |
| <b>第七章 层通信 .....</b>      | <b>137</b> |
| 数据传递模式 .....              | 137        |
| <b>第八章 数据库和数据模式 .....</b> | <b>149</b> |
| 数据访问模式 .....              | 149        |
| 主键模式 .....                | 160        |
| 对象 - 关系映射 .....           | 165        |
| <b>第九章 业务层接口 .....</b>    | <b>178</b> |
| 抽象业务逻辑 .....              | 179        |

|                            |            |
|----------------------------|------------|
| 访问远程服务 .....               | 189        |
| 查找资源 .....                 | 198        |
| <b>第十章 企业并发 .....</b>      | <b>205</b> |
| 事务管理 .....                 | 206        |
| 通用的并发模式 .....              | 219        |
| 实现并发 .....                 | 222        |
| <b>第十一章 消息通信 .....</b>     | <b>233</b> |
| 消息通信和集成 .....              | 235        |
| 消息发送模式 .....               | 239        |
| 消息类型 .....                 | 243        |
| 关联消息 .....                 | 245        |
| 消息客户端模式 .....              | 247        |
| 消息通信和集成 .....              | 255        |
| 进一步阅读 .....                | 261        |
| <b>第十二章 J2EE 反模式 .....</b> | <b>262</b> |
| 反模式的起因 .....               | 263        |
| 体系结构反模式 .....              | 264        |
| 表达层反模式 .....               | 269        |
| EJB 反模式 .....              | 277        |
| <b>附录一 表达层模式 .....</b>     | <b>289</b> |
| <b>附录二 业务层模式 .....</b>     | <b>302</b> |
| <b>附录三 消息通信模式 .....</b>    | <b>328</b> |

---

|                    |     |
|--------------------|-----|
| 附录四 J2EE 反模式 ..... | 339 |
| 源代码索引 .....        | 345 |

---

# 前言

看来，你是打算建立企业应用了？

让时光回到1996年，当时我们正在为波士顿的一家大医院开发一套基于Web的临床数据管理系统。此时Java 1.0.2才刚刚发布，相应的开发工具也开始慢慢地进入市场。Java程序“编写一次，随处可用（Write Once, Run Anywhere）”的口号非常诱人，使得大家的热情丝毫没有因为缺乏能够映射到部署环境的开发环境而降低。而且，这种面向对象的强类型语言还有一套优秀的标准类库。Java语言看起来似乎非常适合我们的应用，至少理论上是这样。所以，在我们的最终应用中开发了几个Java applet和一个原始而较为粗陋的以数据库作为后端的Web网站。使用Java代码替换Perl脚本需要一些复杂的操作，因为当时还没有任何一种Web服务器在内部集成了对Java的支持。此外，程序的性能还遭到置疑，系统维护也存在问题，迫使我们不得不在某些方面退回到使用Perl和C++来开发。该应用本身最终是将Perl CGI脚本、服务器端Java应用（包括为客户端定制的Web服务器的一部分）、HTML和C++ CGI程序等各种元素很奇特地混合在了一起。

我们遇到的首要问题是还不存在能支持应用开发和部署的必备的基础设施。Java对关系数据库的支持还很原始，而且，该语言虽然可以作为Web客户端支持某些特定的操作，但是直到后来引入Java Servlet API，Java才终于具有一个标准的、高性能的集成到Web服务器中的机制。当时的这些限制意味着虽然我们可以使用Java完成某些开发活动，但是这些活动受到了严格的限制。它也意味着我们在当时无法取得像以后在不同环境下使用不同工具集完成项目时那样的成就，所有这些都或多或少表现得有所不同。

现在，Java已经在企业世界中找到了它的位置。性能已经不成问题，甚至可以与任何你所知道的方法竞争。它的API在Java语言随后的四次更迭中逐渐地丰富，这使得Java可以比任何其他语言更广阔地支持企业的活动。Java已经从1996年时乳臭未干的状态

演变成了服务器端业务流程的通用语言。跨平台的图形用户界面曾经被视为是Java最有前景的方面，现在已经逐渐以现实的方式淡出，尤其在用于面向开发人员和系统管理员的产品中。但是，Java已经逐渐地退居幕后，用于构建电子商务网站、企业内联网上的业务应用以及关键性的分布式计算系统。成千上万的人们每天都在与企业级的Java打交道，尽管他们通常不是具体的实现者。

我们即使已经拥有了现代Java环境的所有可用的资源，仍然不能保证我们一定会成功。（准确地说，应用在考虑的时候可以设想得非常周到，但却没有一种我们愿意重复的经验。）人们对工具的使用可能很不充分且很低效。J2EE环境包括了成千上万的类，它纷繁复杂的那么多领域意味着只有极少数的开发人员能够熟知每个领域。J2EE已经建立了标准，使得其实现更加简单，但是基本的设计问题仍然像以前一样存在着。应该说，我们已经有更多的方法可以解决这些问题，也有更多的工具可以减少繁琐的工作。而且，当然了，我们已经有了一个标准的、一致的、广泛部署的平台，可以把那些解决方案构建在它的上面。

这是一本讲述如何使用Java 2企业版（J2EE）为企业环境构建有效应用的书。此书虽然涵盖了J2EE规范中的大多数组件（下面会给出一个简要的概述），但是我们关注的并不是怎样具体地使用J2EE规范中的某个单独的组件，而更关心用于指导有效的软件设计的J2EE设计模式。

除了多年构建企业级Java系统的12年经验之外，本书的作者也像大多数开发人群一样，是十余年在面向对象建模、软件设计方法学以及企业级体系结构等方面研究和开发工作的受益人。在这一时期赢得瞩目的最重要的一个思想就是软件设计模式：一种用于解决通用问题的可重复解决方案的描述方法。本书将把设计模式与企业级软件设计融合在一起。

## 本书的读者对象

读者应该对Java语言本身非常了解，具备良好的操作经验。我们并不要求读者完全精通J2EE，但是我们希望读者能够在较高的层次上把握J2EE技术框架中有哪些可用的东西（我们会在第一章中为还不了解J2EE的读者们提供一个简要的概述，总结每种技术分别用来解决什么问题）。在表达层部分，读者将需要一些Java Servlet和HTTP以及一般性的Web协议的基础知识，但是我们在某些地方提到这些名词时也会作些必要的解释。由于企业应用趋向于数据库密集型，所以业务层部分（从第六章到第十章）会频繁地用到数据库。

读者如果想对于所有的 J2EE 技术有个快速的了解，可以参阅同样由 O'Reilly 出版的由 William Crawford、Jim Farley 和 David Flanagan 合著的《Java Enterprise in a Nutshell》一书（注 1）。

## 本书的组织结构

本书为开发者提供了许多种设计模式，用于在有效的时间内基于 J2EE 平台构建出可扩展的、可伸缩的、可靠的企业系统。一旦你读完此书，你将理解基本的 J2EE 设计模式，并且明白如何将这些模式组合在一起，开发出一个完整的应用。

至少可以用三种方式阅读此书。最简单的方式就是从这里开始，一直读到全书的结束。另一种方式就是选择性地阅读：本书的许多章节是自包含的；如果只对业务层感兴趣，那么可以把注意力放在本书的第二部分，而如果需要一些为数据库中的表创建主键的技巧，那么就只需要读第八章。或者，你也可以先阅读此书的四个附录，然后再回过头去阅读书中的具体内容。因为附录中会给你一个通用的大纲，介绍了不同的模式分别可以做什么事情，同时也说明了怎样使用这些模式来帮助你完成当前的工作，所以可以根据这个大纲回过头来更加深入地阅读对应的章节。

### 第一章

提供了一个设计模式和企业设计的简单入门。除了介绍设计模式本身之外，我们将讨论模式如何帮助达到四个主要的设计目标：可扩展性、可靠性、可伸缩性和时效性。当然，不必一蹴而就：我们并不是奇迹的创造者。

### 第二章

包括了一个统一建模语言（UML）的入门，这是一种标准化的图形化描述应用目标、结构和处理的方法。我们将贯穿本书的剩余章节使用 UML 描述各个设计模式。我们已经努力在不花几百页篇幅的基础上使该入门尽可能完整，以便你清楚地了解 UML 能够做什么，以及你可以从何处入手学习更多的东西。

本书的剩下部分引入了大量的 J2EE 设计模式，并将它们放在用到它们的设计环境的上下文中。与其创建一个传统的模式目录，不如选择以叙述的风格来介绍这些模式，这使每一个部分都足够独立，以便你可以轻易地聚焦于模式本身。可伸缩性、可扩展性和可靠性的底层主题贯穿于每一个章节。有些章节可以独立存在：在应用的 Web 层上工作的程序员不必要知道 MQ 消息通信系统的模式，反之亦然。但是，我们相信，成熟而全面的开发人员会希望自己熟悉这整个体系。我们将会从表达层着手。

---

注 1： 建议为朋友、亲戚和爱人购买一些。

### 第三章

概述了构造面向Web端的J2EE应用时涉及的通用模式,提供了一个可扩展性框架。

### 第四章

继续对可扩展性的讨论,引入了额外的关注于创建更灵活的表达层逻辑的模式。

### 第五章

着眼于表达层的可伸缩性、用于改善资源管理和通常性能的表达策略等。

接下来,我们把表达层放在一边,转向业务层。

### 第六章

介绍了业务层,着眼于应用的领域模型。

### 第七章

提供了为创建表达层和业务层之间更高效和可扩展的接口所设计的模式——前者与用户进行交互,后者与应用数据进行交互。

### 第八章

关注于数据库,提供了一组用于实现你的应用中的持久化的模式。

### 第九章

讨论了集中表达你的应用的业务逻辑的方法。

### 第十章

讨论了并发性,换句话说,即任何可能存在两个或者更多用户试图同时做同一件事情的应用所面临的问题。

### 第十一章

介绍了一组用于企业消息通信系统的模式。

### 第十二章

调整了模式章节的方向,并介绍了反模式(antipattern)的概念:重现了企业应用开发中最糟糕的做法,这些做法在任何时刻即使不考虑成本也必须尽量避免。

我们以一组附录结束,在附录中以编目的形式概述了本书中介绍的每种模式。这个编目提供了本书中各种模式的一个快速索引,同时也是高效探索当前问题潜在解决方案的一种方法。

## 进一步阅读

因为设计模式本质上全是记录解决方案的,所以存在大量与该主题相关的文献。下面是模式领域中的其他一些重要书目:

- 《Patterns of Enterprise Application Architecture》，Martin Fowler 著，Addison-Wesley 出版，介绍了 Java 和 .NET 企业架构，作者是该领域的引领者之一。
- 《Core J2EE Patterns》，第二版，Deepak Alur、John Crupi 和 Dan Malks 著，Prentice Hall PTR 出版，Sun 贡献的模式作品。
- 《Design Patterns》，Erich Gamma、Richard Helm、Ralph Johnson 和 John Vlissides 合著，Addison-Wesley 出版，原创的软件设计模式书籍，包含了对于企业开发和独立系统都很有用的模式。
- 《Pattern Oriented Software Architecture》，第 1 卷和第 2 卷，Frank Buschman 等著，Wiley 出版，关注于从头开始构造较大的体系结构，超越了 Java 的世界。
- 《EJB Design Patterns》，Floyd Marinescu 著，Wiley 出版，用于 EJB 环境的设计模式。
- 《Bitter Java》，Bruce Tate 著，Manning 出版，对于 Java 反模式的极好介绍。

而在 Java 这一端，O'Reilly 的 Java 丛书提供了一系列关注于底层实现技术的书籍。特别是，我们认为本书是下面这些书籍的最佳伴侣：

- 《Java Enterprise in a Nutshell》，第二版，William Crawford、Jim Farley 和 David Flanagan 合著，O'Reilly 出版，你想知道而又害怕去问的关于 J2EE API 的所有东西。
- 《Java Enterprise Best Practices》，O'Reilly 的 Java 作者们合著，O'Reilly 出版，关于 J2EE API 的高级技巧。

此外还有：

- 《Java Servlet Programming》，第二版，Jason Hunter 和 William Crawford 合著，O'Reilly 出版。
- 《Java Web Services》，David Chappell 和 Tyler Jewell 合著，O'Reilly 出版。
- 《Java Message Service》，Richard Monson-Haefel 和 Dave Chappell 合著，O'Reilly 出版。
- 《Enterprise JavaBeans》，第三版，Richard Monson-Haefel 著，O'Reilly 出版。

## 排版约定

本书中使用了如下这些排版印刷约定：

**斜体 (*Italic*)**

用于文件名、目录、着重以及技术术语的第一次使用。

**等宽体 (*Constant width*)**

用于代码举例，也用于 Java 类的名字和 Bean 的名字。

**等宽斜体 (*Constant width italic*)**

表示一个术语，它应该被你的程序中的一个实际的值替换。

**粗体 (*Constant width bold*)**

用于文本形式的用户输入，以及在例子中表示输入和输出。

## 建议与评论

本书的内容都经过测试，尽管我们做了最大的努力，但错误和疏忽仍然是在所难免的。如果你发现有什么错误，或者是对将来的版本有什么建议，请通过下面的地址告诉我们：

美国：

O'Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472

中国：

100080 北京市海淀区知春路 49 号希格玛公寓 B 座 809 室  
奥莱理软件（北京）有限公司

本书有一个网页，列出了勘误、例子或者任何额外的信息。你可以访问如下页面：

<http://www.oreilly.com/catalog/j2eedp/>

若要为本书提意见和问题，请发邮件给：

[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)

如果向了解书籍、会议、资源中心以及 O'Reilly 网络的信息，请参见如下 O'Reilly 网站：

<http://www.oreilly.com>

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

作者们也为本书维护了一个主页，参见<http://www.j2eepatterns.info>。另外有一个极有价值的模式资源是在<http://www.java.net>的模式社区。

## 致谢

作者要感谢曾经花费时间阅读过本书全部或者部分草稿的每一个人。我们特别要感谢 Ted Chen、William Connor、Liz Evans、Jim Farley、Eric Friedman、Phil Haigh、Dave Henry、Ron Hitchens、James “JC” Norman、Paul Tecca 和 Dave Wood，他们仔细阅读过本书中的每一个章节。我们还要感谢所有的作者、理论学家和程序员，他们为我们全面理解设计模式和企业架构做出了贡献。当然，我们的误解、错误和不足之处也在所难免，恳请读者批评指正。

在 O'Reilly 公司，Colleen Gorman 完成了一项非常成功的工作，在开发过程的最后将所有东西整合到了一起。当然，我们的编辑 Mike Loukides 也为项目倾注了心血，从刚开始就给了我们自由度，直到最后整理成我们所能想像的最好著作。

## William Crawford

首先，我要感谢 Jon Kaplan 与我共同完成该项目，也要感谢 Tracy Leeds Kaplan 让他这么做，即使他们的婚礼恰好在撰写计划的时间内。结果就产生了这么好的佳作。

在本书完成期间，我曾经为两个人工作，他们的帮助使本书得以问世。Invantage 的 Martin Streeter 和 Perceptive Informatics 公司的 Howard Foster 都努力确保我有足够的时间，并且支持我成功完成了项目。Perceptive 公司在 Boulder、Birmingham 和 Boston 的非常有能力的工程队伍也为该项目的成功做出了巨大的贡献：所有这些都构成了一个很好的环境。

Mike Loukides 非常善解人意，在把这本书提交之后，当 Invantage 公司成为 Perceptive Informatics 公司一部分的时候，我的社交圈变化极大。我没有计划上半年累积花一个半月在英格兰度过，而 Mike 不仅展示了他的高尚品性，更显示出几乎像上帝一样的耐心。

没有我的家庭的帮助和支持的话，我刚开始就不会进入这种事业，随着计算机越来越多，我的家庭相当耐心地给予我支持。当然，我还要非常感激 Rebecca Goetz (<http://rebecca-goetz.blogspot.com>) 在项目进行的自始至终都不断地给我鼓励，这是完全不可替代的：感谢你容忍我，亲爱的。

## Jonathan Kaplan

我要感谢我的合著者 Will Crawford 和编辑 Mike Loukides 给我这样一次撰写本书的机会。Will 给了我线索，自始至终给予我指导和意见。Mike 很通情达理地包容了我的无数错误、重写和错过期限，还一直为我提供建议和帮助。

这种书籍实际上是一个团队的成果，它包括了我们的技术评审们的不懈努力，特别是 Dave Wood 和 O'Reilly 的专业团队。我还想要感谢 Sun 公司的同事们维护了一个非常有益于思考和构建创新软件的环境。

在撰写本书的过程中，我不断地得到来自家庭的支持——包括我的父母、祖父母、兄弟以及 Leeds 一家人，不管我多少次说“再写一遍”时，他们始终都未失去过信任。

最后，我想要感谢我的妻子 Tracy。没有她的爱、鼓励和耐心，我就不可能写出这本书。

— Cambridge, MA

2003 年 8 月

---

# 第一章

# Java 企业设计

在我们深入到企业应用体系结构的特定设计模式之前,有必要奠定一个良好的知识基础。在本章中,我们将进一步了解设计模式的出现背景,然后,简单地概述一下J2EE。接下来,我们将揭示企业应用的各个不同层次,从客户端到上层的业务逻辑和企业级服务。我们会看到基于组件的设计,以及用于保证企业系统开发的四个重要的主题:可扩展性、可伸缩性、可靠性和时间限制。

在下一章,我们会转变思路,关注一下UML,即统一建模语言,它为描述设计模式和完整的系统提供了一个标准的词汇集。

本书后面部分的组织主要围绕构建优良的企业体系结构的实践以及用于支持这些体系结构的设计模式等话题。

## 设计模式

设计模式是对于重复出现的问题的一种可以重用的解决方法。从编程的角度来说,一个模式提供了一系列特定的交互,这些交互可以应用于某些通用的目标,以解决某个众所周知的问题。好的模式会在保证有效求解问题的基础上,突破它们所能解决的问题的规模。

最简单的模式概括起来,可能仅仅是一个或者两个句子。使用数据库为网站存储信息,这就是一个模式,虽然它相当概括,但确实是一个模式。更复杂的模式需要更多的解释,可能包括使用建模语言或者各种其他格式作更长的描述。

设计模式起源于计算机行业之外,早期出现在传统的(与计算机系统相反)建筑行业中。建筑的架构与软件的架构比一般人想像的还要接近,它们之间具有非常多的相似之处。