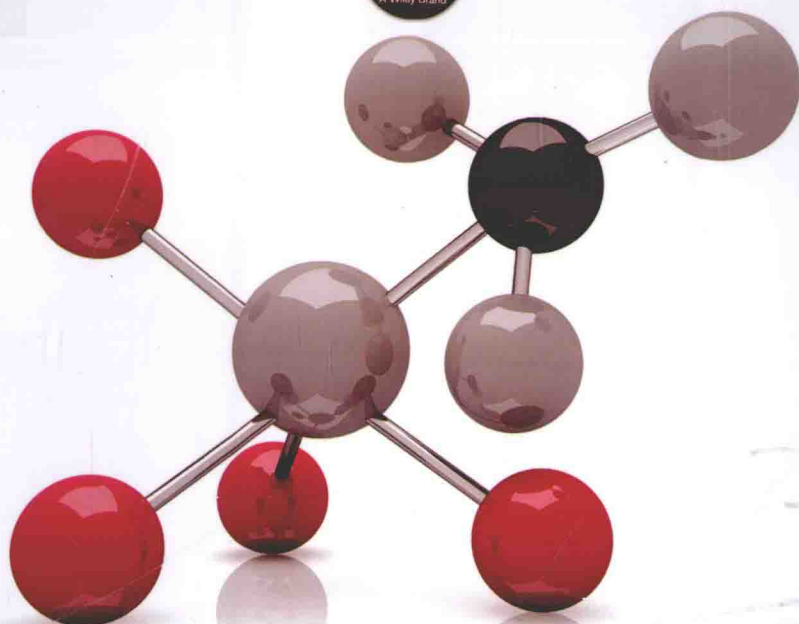




Professional Java EE Design Patterns

Java EE

设计模式解析与应用

[美] Murat Yener
Alex Theedom
张龙

著
译



清华大学出版社

Java EE 设计模式解析 与应用

[美] Murat Yener 著
Alex Theedom
张 龙 译

清华大学出版社

北 京

Murat Yener, Alex Theedom
Professional Java EE Design Patterns
EISBN: 978-1-118-84341-3

Copyright © 2015 by John Wiley & Sons, Inc., Indianapolis, Indiana

All Rights Reserved. This translation published under license.

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Java is a registered trademark of Oracle America, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2015-2357

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

本书封面贴有 Wiley 公司防伪标签, 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 EE 设计模式解析与应用/(美)叶尼(Yener, M.), (美)希多姆(Theedom, A.)著; 张龙译. —北京: 清华大学出版社, 2015
书名原文: Professional Java EE Design Patterns
ISBN 978-7-302-41586-2

I. ①J… II. ①叶… ②希… ③张… III. ①JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 225363 号

责任编辑: 王 军 韩宏志

装帧设计: 牛静敏

责任校对: 曹 阳

责任印制: 沈 露

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 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

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 北京密云胶印厂

经 销: 全国新华书店

开 本: 148mm×210mm

印 张: 10.25

字 数: 276 千字

版 次: 2015 年 10 月第 1 版

印 次: 2015 年 10 月第 1 次印刷

印 数: 1~3000

定 价: 59.80 元

产品编号: 062759-01

译者序

设计模式是一个既“古老”又“时髦”的话题。说它“古老”是因为 GoF 一书已经出版多年，这本巨著开创了设计模式的先河，从那时起才真正有了指导软件架构设计与重用的准则；说它“时髦”是因为时至今日仍有很多人在学习设计模式，利用设计模式构建软件系统。因此从某种意义上来说，设计模式将现代软件开发与传统软件工程有机地结合到一起，前人的诸多经验可以帮助后人设计出健壮性更好、可伸缩性更棒的软件系统。

时至今日，Java 8 已发布一年有余，而 Java EE 7 也早已深入人心。Java 是一门非常讲求设计模式的语言，从发布伊始就在内部大量应用了设计模式，甚至还提供了某些模式的内建实现。传统的 GoF 一书所给出的 23 种模式被称为“经典”设计模式；时过境迁，随着软件开发的不断发展，又有大量新模式涌现出来。在这个时代背景下，Java EE 7 也对诸多模式提供了内建实现，让广大开发者只需通过少量注解与配置即可实现出众多设计模式；这一方面简化了开发工作，另一方面又确保了程序的健壮性。

本书主要介绍的就是如何通过 Java EE 实现设计模式。相对于“经典”设计模式，Java EE 对模式的实现更简洁、更健壮，可读性也更好。在市场上浩如烟海的设计模式图书中，本书可谓是独树一帜，每一章都专注于一种特定的设计模式，首先介绍传统实现方法，接下来介绍如何通过 Java EE 进行实现，最后又给出设计模式的最佳应用场景。本书涵盖了最常用的设计模式，如工厂模式、单例模式、观察者模式、装饰模式、MVC 模式、异步编程、门面模式、

数据访问模式、定时器服务等，并对每种模式都进行了细致入微的讲解，是广大软件开发者不可多得的设计模式宝典。

本人在初学设计模式时就从单元测试框架的鼻祖 JUnit 开始，随着对 JUnit 源代码的层层分析，越来越感叹二位作者 Kent Beck 和 Erich Gam 在模式上的造诣。建议各位读者能静下心来好好研究一下 JUnit 的源代码，相信会让大家对设计模式有更深刻的理解。

有模式就有反模式，虽然设计模式能够帮助我们设计出更棒的系统，但对设计模式的大量堆砌却导致了反模式的出现。模式并非用得越多就越好，而是要在合理的场景下选择最恰当的模式才能起到最好的效果。本书也通过作者的切身经历揭示出反模式存在的根源以及该如何避免常见的反模式。

翻译技术图书是一项艰苦的劳动，不仅有体力的付出，更有大量脑力的付出。在此，请允许我感谢清华大学出版社的编辑们，非常感谢他们在本书翻译过程中对我的包容与鼓励；其次，我要感谢我的父母，没有你们的精心抚育与教诲就不会有今天的我；再次，我将深深的感激之情送给我的妻子张明辉，谢谢她在本书翻译过程中给予我持续不断的支持与理解，本书的顺利出版有她一半的功劳；最后，将本书送给我亲爱的孩子张梓轩小朋友，愿他未来能够开开心心，茁壮成长。

全书由张龙翻译。参加本书翻译的人员还有张彤辉、张兴国、焦伟、张淑贤、王辉、李志芹、马元贺、王凯、周锐、单会明、高鹏、张淑华、王冠等，在此一并表示感谢。

尽管在翻译过程中译者已经尽了最大的努力力求保证译文的准确性和流畅性，但囿于水平有限，读者在阅读过程中如果发现了错误或不准确之处请及时与我联系，以便再版时修正。我们的邮箱是 wkservice@vip.163.com。最后，诚挚希望本书能够带领读者走进设计模式之门，帮助大家设计出健壮性更棒、可伸缩性更好的软件系统。

作者简介

Murat Yener 是一位代码极客与开源软件提交者，就职于 Intel New Devices Group，是一名 Android 开发者。他在 Java、Web 框架、Java EE 与 OSGi 应用开发上经验丰富；此外，他还教授课程并指导学生。Murat 是 Eclipse 提交者，同时也是 Eclipse Libra 项目的最初提交者之一，他目前致力于通过 HTML5 与 mGWT 构建原生与混合移动应用。

Murat 从 2009 年开始就是 GDG Istanbul 用户组的领导者，组织、参与并在各种大会上演讲。他还是 JavaOne、EclipseCon 与 DevOxx 大会的常客。

Linkedin——www.linkedin.com/in/muratyener

twitter——[@yenerm](https://twitter.com/yenerm)

blog——www.devchronicles.com

Alex Theedom 是一位资深 Java 开发者，就职于 Indigo Code Collective(indigocodecollective.com，隶属于 E-scape Group)，他在公司中主要负责一款基于微服务、用户彩票与即时双赢游戏平台的架构设计与开发工作。

在此之前，他为一家国际化的西班牙银行开发过 ATM 软件，也为一家软件咨询公司开发过一款代码质量分析软件。

Alex 在 Java Web 应用开发方面颇具经验，涉及很多领域，包括财务、在线学习、彩票与软件开发等。对开发的强烈爱好促使他的项目遍及并超出了整个欧洲。他是 alextheedom.com 的一位博主，

并且经常参与在线论坛帮助他人解决问题。

Linkedin—www.linkedin.com/in/alextheedom

Twitter—[@alextheedom](https://twitter.com/alextheedom)

Blog—www.alextheedom.com

技术编辑简介

Mohamed Sanaulla 是一位拥有 5 年以上专业软件开发经验的软件开发者。他目前供职于印度最大的一家电子商务公司，同时也是 JavaRanch 论坛的版主。在工作的闲暇时间，他总是会陪伴自己可爱的女儿。他经常会在 <http://blog.sanaulla.info> 上分享自己关于软件开发的经验与想法。

致 谢

本书的合著者 Alex 总是对我说, 我们想要撰写一本自己梦寐以求并且喜欢阅读的书。首先, 我要感谢 Alex, 感谢他的耐心、勤奋的工作以及广博的知识。如果没有他, 这本书肯定不会像现在这样出色。

非常感谢之前的策划编辑 Mary James, 她曾联系过我希望我能编写一本关于 Spring 的书, 后来听取了我的建议, 这些建议形成了本书的基础。要是没有她的支持与指导, 这本书是不可能问世的。任何语言都无法表达我对 Adaobi Obi Tulton 的感激之情, 他耐心地处理了本书的所有细节, 同时尽最大可能帮助我们缓解进度的压力。当然了, 还要感谢 Wrox/Wiley 中每一个在本书出版过程中帮助过我们的朋友。此外, 非常感谢 Reza Rahman 对我的鼓励。

感谢在软件开发职业生涯中给予我巨大帮助的重要的 3 个人。

首先, 感谢我的父亲 Mustafa Yener, 在我还要模型赛车这么小的年龄就给我买了第一台电脑 C64。我就是在这台电脑上编写了人生最初的代码。

接下来, 感谢我的论文导师 Mahir Vardar 教授, 他的指导帮助我开启了职业生涯。

最后, 感谢我一生的导师与朋友(也是我前老板)Naci Dai, 他教授了我关于成为一名专业的软件开发所需的一切。

——Murat Yener

我们对自己撰写的第一本书感到非常骄傲和自豪，希望你在阅读时能够从中获得最大的收益。我们在撰写本书时希望达成这样的目标：如果自己没写而别人写了，那么我们也会买一本，现在这个愿望终于实现了。

不过，要是没有那些对本书出版直接或间接地付出了巨大的心力、耐心与理解的人，本书是不可能问世的。我们要向 Wiley Publishing 敬业而又颇具经验的团队所做出的贡献表示深深的敬意，感谢他们与我们的风雨相伴，并且对我们非常信任。将特别感谢送给本书的策划编辑 Mary James，正是她一如既往的支持才使得本书能够与各位读者见面。感谢 Adaobi Obi Tulton，他的耐心与推动力让我们能够保持节奏，他对细节的专注让我们少走了很多弯路。我还要感谢本书的合著者 Murat Yener，他的灵感与幽默成就了本书的独一无二；最后谢谢我的妻子 Maria Eugenia García García，感谢她在本书撰写之际给予我的全力支持与理解。谢谢你们。

——Alex Theedom

序 言

愚者提出智者早在 1000 年前就已经回答过的问题。

——Johann Wolfgang von Goethe

设计模式将我们与过去和未来连接起来。它们构成了一种基础语言，是由才华横溢的工程师针对常见问题提出的易于理解的解决方案，并形成了集体智慧。设计模式或蓝图以各种方式存在于每一个工程领域中，软件开发也不例外。事实上，设计模式可能是通往工程之路最切实可行的链接，而非工匠或手艺人这种组织性更强但严密性更差的世界。设计模式的艺术与科学被带到软件工程(特别是企业级 Java)的世界是由 GoF 一书带动的。从 J2EE、Spring 到现代轻量级 Java EE 这一路上都有设计模式相伴，这是事出有因的。服务端 Java 开发者编写的通常都是任务关键的应用，这些应用需要经得起时间的考验，因此会从设计模式所代表的原则中获益匪浅。

撰写一本关于设计模式的图书非常不易，更不必说是如何在 Java EE 应用中使用设计模式了。作者不仅需要掌握关于 API 的基础知识与模式本身，还需要对辛苦获得的经验有深刻的洞察力，同时又要能将复杂的概念以优雅的方式表述出来。我很高兴 Java EE 领域能有 Murat 与 Alex 来完成这个壮举。

本书很好地完成了上述目标。此外，本书还走在了前沿，涵盖了 Java EE 7(而不仅是 Java EE 6 或 Java EE 5)。事实上，本书所介绍的很多模式，如单例、工厂、模型-视图-控制器(MVC)、装饰与观察者等现在都已经融合进了 Java EE 平台中。其他诸如门面、数

据访问对象(DAO)与数据传输对象(DTO)等模式也都优雅地跃然纸上。Murat 与 Alex 介绍每一种模式,解释实际动机,并且讨论如何通过 Java EE 来实现。

为这样一本重要的图书编写序言本身就是一件非常荣幸的事情,我希望这本书能够成为每个优秀的 Java EE 开发者案头的必备之作。我也希望你能够喜欢本书,它会帮助你编写出更棒、更优秀的企业级 Java 应用。

M. Reza Rahman

Oracle 公司 Java EE/GlassFish 技术讲师

前 言

本书介绍知名的 GoF 一书最初所谈到的经典设计模式¹，然后探讨如何在 Java EE 6 与 7 中实现它们。

每一章开始都会介绍每一种模式的传统实现，然后展示如何通过 Java EE 特有的语义来实现它。

我们通过完整的代码示例演示模式的传统实现与 Java EE 实现，并通过实际的示例来说明模式的用法(或是误用)。

我们介绍了每一种模式的优缺点及其用法。每一章都会通过一些练习来考察你对模式在 Java EE 中实现的理解。

本书读者对象

本书面向各个层次的软件开发人员，涵盖了关于模式的一切内容，从对其他图书的引用，到基本的 Java 实现代码，再到 Java EE 实现，最后通过实际示例介绍了如何以及何时该使用模式。本书还通过作者的切身经历谈到了关于模式的优秀以及糟糕的实践。

对设计模式与 Java EE 有基本的了解可以帮助你更好地阅读本书。

如果对模式与基本的 Java 实现已经具备了一些经验，那么你可以直接跳到 Java EE 实现上来，这对于重温设计模式知识是非常有帮助的。

本书涵盖的内容

除了一些新模式外，本书还介绍了 Java EE 提供的作为标准实现一部分的所有经典设计模式，覆盖范围从 Java EE 5 到最新版的 Java EE 7。

我们希望本书作为你的日常参考，能够一直在你书架上占据一席之地。

本书的组织结构

每一章都重点关注一种设计模式。如果是经典模式，那么会在介绍完该模式后给出一个简单的 Java 实现。每一章都会通过作者的亲身经历介绍关于这一章所重点关注的模式的优秀实践与糟糕实践。作者亲身经历之后是 Java EE 实现、示例与说明。每个代码示例都可以独立运行。每一章最后都会给出何时以及如何能高效使用模式的说明。

阅读本书的前提条件

操作系统上装有 Java 虚拟机(JVM)实现的任何现代计算机都可以运行书中的示例。出于简化编码的需求，你需要一个集成开发环境(IDE)。书中示例可以运行在任何流行的现代 IDE 上，如 Eclipse、NetBeans 及 IntelliJ。

你需要 Java Development Kit(JDK) for Java EE 7 才能编译和运行代码示例，不过一些代码示例也可以运行在旧版本的 Java EE JDK 上。

可使用任何 Java EE 7 兼容的应用服务器来运行示例。我们在

Glassfish 和 TomEE 上运行了全部代码示例，前者是参考实现服务器，后者是流行的 Java Web 服务器 Tomcat 的 Java EE 版本。你可以使用任何服务器，不过由于 Glassfish 是参考实现，因此最好使用它。

要运行书中示例，你需要：

- 安装有 JDK for Java EE 7 的操作系统，如 Linux、Mac OS X 或 Windows
- Java EE 7 JDK
- 一个 IDE，如 Eclipse for Java EE Developers、NetBeans 或 IntelliJ
- 与 Java EE 7 兼容的应用服务器，如 GlassFish 或 TomEE

示例源代码可以从 Wrox 网站下载，网址是：www.wrox.com/go/projavaeedesignpatterns。

本书的写作动机

2011 年 11 月，在一场关于项目应该使用 Java EE 还是 Spring 的争论过后，我回到我的工位，写下了一篇题为“Java EE 6 and the Ewoks”的博文²，这篇博文很快就传开了。故事是根据电视节目 How I Met Your Mother 展开的。其中，纨绔子弟 Barney 介绍了一个关于依娃族的理论，依娃族是一种类似于泰迪熊的生物，在《星际大战》第 6 部中出现。粉丝对于依娃族的感受可谓是泾渭分明。

根据 Barney 所述，出生在 1973 年 5 月 25 日(那时《星球大战》第 3 部：武士复仇刚刚上映)之前的人觉得依娃族非常幼稚，因此很讨厌他们。不过，之后出生的人觉得依娃族非常可爱，因为他们总是会让人想起泰迪熊。

回到我的故事上来。与客户争论关于 Java EE 与 Spring 的选择

问题让我认识到这非常类似于依娃族理论。那些在项目中使用过 J2EE 1.4(EJB 1.0/2.0/2.1)的人经历过缓慢、低效的开发环境,以及占内存且 Bug 重重的 IDE,服务器要花好几分钟才能启动。其架构有些过于工程化,失败的可能性很高,因此他们迁移到了 Spring。无论使用何种版本,这些人都非常憎恶 Java EE。Java EE 5 的发布被低估了,并未给人留下深刻的印象。

Java EE 再也不是 J2EE。它现在更加开放,拥有庞大的社区,并吸收了 Spring 与 Hibernate 等框架的优秀理念来重新打磨自己。第 1 个重大变化就是其架构与编码风格。Enterprise JavaBeans(EJB)采用了轻量级的 Plain Old Java Object(POJO)模型,几乎没什么用处的实体 Bean 被 Java Persistence API(JPA)所取代,REST 与 Web Service 成为标准以及运行时的组成部分,注解取代了 XML 配置。不过,有些人会说 Java EE 5 并未对大型迁移做好准备,因为它不像 Spring 那样成熟,开发环境的响应性也没有那么好。使用 Spring 与 Tomcat 而非应用服务器与 EJB 和 Java EE 5 会极大提升开发效率,不过 Java EE 5 依然从头开始,向企业级 Java 平台的设计、使用与架构上迈出了重要一步。

Java EE 6 与 7 延续了这一转变,它们使用了与 Java EE 5 相同的原则和理念。Java EE 是开发的一个很好的选择,但争论并未结束,这都要拜依娃族理论所赐。

某年炎热的仲夏,我第一次接到 Wrox/Wiley 的电话,询问我是否有兴趣写一本关于 Spring 的图书。我在 Spring 开发方面颇具经验,也非常有自信,不过市场上已经有太多关于 Spring 的图书了,这使得再写一本新书的价值变得没那么大了。

此外,在版本 6 发布之前我就一直在使用 Java EE。考虑到 Spring 与 Java EE 之间的争论、我的博客以及伊娃族,我打算撰写一本关于 Java EE 的图书。不过,就像 Spring 一样,我承认市场上已经有太多关于 Java EE 的优秀图书了。我总觉得 Java EE 的一些属性被低

估了。Java EE 提供了很多设计模式的不错的内建实现，只需通过几个简单的注解即可。

GoF 一书中所列出的经典模式被广泛应用在各种语言、框架与平台中。J2EE 与 Java EE 也不例外。事实上，Java EE 更进一步，为很多模式提供了默认实现，不过依然有很多颇具经验的开发者低估了这些开箱即用的实现的价值。

我已经为这些模式写了将近一年的博客，因此决定写一本关于如何在 Java EE 中实现“经典”设计模式的图书。在阅读这本书时，我相信你一定会给予积极的反馈。

本书填平了 Java EE 平台与 GoF 一书中所列出的经典设计模式之间的鸿沟，同时还谈及了一些新模式。这样，我们实际上并未撰写另一本 Java EE 图书，而是以 Java EE 的方式介绍了设计模式。

作为该平台的忠实信徒，我撰写博客、讨论如何通过 Java EE 实现设计模式，从而扩展我的知识面与经验，因此对于我个人来说，撰写这样一本书最棒的事情就是我有机会写下我真的怀有热情的一些东西。虽然我的博客提供了一些更简单的示例，不过他们是在我需要时用作参考的，撰写一本书则更加正式，格式也要更合规，同时遵循着相同的理念，这真是很棒的机会。

本书合著者与我所撰写的每一章都有相同的目标：内容一定要是我们自己都喜欢看的。我们希望这本书能够成为大家的参考书籍。

我们希望你能喜欢阅读本书，就像我们喜欢撰写本书一样。

本书约定

为了帮助你从本书中获得最大收益并跟上节奏，全书通篇使用了大量约定。



注意：说明表示注释、技巧、提示、诀窍或是当前所介绍内容的旁注。

我们以如下两种方式呈现代码：

- 对于大部分代码示例来说，我们使用等宽字体并且没有突出显示。
- 我们使用**粗体**强调当前上下文中特别重要的代码，或是显示出与之前代码片段不同的代码。

p2p.wrox.com

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于你张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给你传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接。
- (2) 阅读使用协议，并单击 Agree 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 Submit 按钮。
- (4) 你会收到一封电子邮件，其中的信息描述了如何验证账户，