

Spring技术内幕——深入解析Spring架构与设计 (揭秘系列丛书)

作者：计文柯

揭秘系列丛书

Spring技术内幕

——深入解析Spring架构与设计原理

计文柯 著

ISBN: 978-7-111-28806-0

本书纸版由机械工业出版社于2009年出版，电子版由华章分社（北京华章图文信息有限公司）全球范围内制作与发行。

版权所有，侵权必究

客服热线：+ 86-10-68995265

客服信箱：service@bbbvip.com

官方网址：www.bbbvip.com

新浪微博 @研发书局

腾讯微博 @yanfabook

本书赞誉

作为一个有近10年历史的成功框架，Spring在Java开发中具有举足轻重的作用。本书从源代码分析入手，对Spring的架构原理和设计思想进行了全面剖析，不仅能让我们更深入、更彻底地认识Spring，领略Spring的架构之美和设计之美，而且更重要的是，它将全面提升我们的Spring开发技能。

——Spring开发者社区

这是值得所有Spring开发者反复研读的一本书。Spring是一个优秀的轻量级企业应用开发框架，是Java开发中最流行的工具之一，也是Java程序员必须熟练掌握的一门技术之一。本书从Spring实现原理的角度揭开了Spring的神秘面纱，使得我们在利用Spring进行开发时，不仅能知其然，还能知其所以然，从本质上提升我们对Spring的理解和开发水平。

——Spring中文用户组

这是所有Java程序员应该认真阅读的一本书。Spring框架主要是用Java语言来实现的，本书对大量经典的、优雅的Spring实现代码进行了赏析，不禁让我们感慨Rod Johnson（Spring之父）对Java语言的运用之精妙。如果能跟随本书穿越Spring源代码的丛林，也许我们对Java语言的理解和掌握将会有质的飞跃。

——中文Java技术网

这是所有软件架构师必备的一本书。成为一名出色的软件架构师，也许是每一位开发者的梦想。在成长为架构师的过程中，在实践中积累并总结经验固然很重要，但是从现有的成功架构中取经也是必不可少的。本书对Spring的各个功能模块的架构原理和设计思想进行了深入剖析，值得所有架构师用心体会和研究，必将受益匪浅。

——架构师中国

前言

作为Java领域中最成功的开源软件之一，Spring在Java开发中，有不可替代的作用和地位。本书以Spring的源代码为依托，从内部实现的角度，对Spring的设计原理、架构和运行机制进行了翔实的分析。

忽如一夜春风来，伴随着Rod Johnson的Expert One-on-One J2EE Design and Development（2002年）一书的出版而正式发布的Spring框架（也就是当年的interface21），经过这几年的发展，已经逐渐成熟起来。吹面不寒杨柳风，Spring带来的崭新开发理念，也早已伴随着它的广泛应用而飞入寻常百姓家。

与此同时，随着Spring的不断成熟和完善，开源社区的成长，以及Rod Johnson的得力领导，以Spring为核心的一系列开源软件产品也越来越丰富，现已发展成为一个包括软件构建、开发、运行、部署整个软件生命周期的产品族群。Spring不但改变了Java EE应用的开发和服务模式，向纯商业软件发起了强有力的挑战，而且已成为Java软件生态链中不可或缺的重要组成部分。它所具备的那种平易近人，但却内涵丰富的品质，对我们这些软件爱好者来说，实在是一个不可多得的学习范本。

简化Java企业级应用开发是Spring框架的目标。其轻量级的开发思想，为开发者提供便利的出发点（for the developer, to the developer and by the developer—这是Rod Johnson在一次演讲中的开场白），以及具有活力的开源社区，所有的这些，都为使用Java开发企业应用和Web应用带来了福音。

在Java企业应用中，与我们熟悉的企业应用服务器一样，Spring也希望能够集成管理企业应用资源，以及为应用开发提供平台支持。在这一点上，Spring与UNIX和Windows等传统意义上的操作系统在计算系统中起到的作用是类似的。不同点在于，传统操作系统关心的是存储、计算、通信、外围设备等这些物理资源的管理，并在管理这些资源的基础上为应用程序提供一个统一平台和服务接口；而Spring关心的是如何为开发者集中管理在Java企业应用和Web应用中涉及的数据持久化、事务处理、消息中间件、分布式计算等抽象资源，并在此基础上为应用提供了一个基于POJO的开发环境。尽管二者面向的资源、管理的对象、支持的应用，以及使用的场景不同，但它们在整個计算系统中的定位，却有着可以类比和相互参考之处。所以，笔者根据对传统操作系统的认识方法，粗浅地把对Spring框架的实现划分为核心、组件和应用这三个基本层次，通过对这三个层次中的一些主要特性的实现来剖析Spring的工作原理和运作机制。同时，也把这样的认识逻辑用来组织本书中需要阐述的内容。

在这样的层次划分中，首先看到的是对IoC容器和AOP这两个核心模块的工作原理的分析，它们都是Spring平台实现的核心部分；同时，它们也是Spring的其他模块实现的基础。虽然大多数开发者都只是在此基础上进行相关的配置和使用外部功能，但是深入理解这两个核心模块的工作原理和运作机制对于我们更好地使用Spring进行开发是至关重要的。因为Spring简化Java EE开发是通过对POJO开发提供支持来实现的。Spring通过为基于POJO的开发模式提供支持，从而让应用开发与复杂的Java EE服务实现解耦，并由此通过提高单元测试覆盖率（也就是应用系统的可测试性）来有效地提高Spring应用的开发质量。在这样的开发场景下，需要把为POJO提供支持的各种Java EE服务支持抽象到Spring应用平台中去，并将其封装起来。具体地说，这一系列的封装工作在Spring及其应用实现中离不开IoC容器和AOP这两个核心模块的支持，它们很大程度上体现了Spring作为应用开发平台的核心价值。它们的实现是Rod Johnson在他的另外一本著作Expert One-on-One J2EE Development without EJB中，所提到“Without EJB设计思想”的具体体现，同时，也深刻地体现了Spring背后的设计理念。

其次，在IoC容器和AOP这两个核心模块的支持下，Spring为了简化Java EE开发，为应用开发提供了许多现成的用户态系统组件，比如事务处理、Web MVC、DBC、O/R映射、远端调用等，通过这些系统组件为企业应用服务的实现提供驱动支持。这些由Spring或者其生态系统（其本身、子项目或者社区）提供的、类似于驱动模块般的系统组件是开发应用时经常会用到的Java EE服务抽象。Spring让用户可以用POJO来开发具体的应用，而这些应用往往需要Java EE服务的有力支撑。通过使用Spring提供的这些类似于驱动组件的中间产品，通过这一层Java EE服务的抽象，用户可以通过使用简单的开发接口或者应用模板很方便地使用各种Java EE服务和灵活地选取提供这些服务的各种不同的具体实现方案。比如，可以在各种第三方开源软件或者商业产品中自由地选择。

Spring作为一个开源项目，它本身就是一个开放的生态系统。对于与Spring相关的一些项目，可以把它们看作某个领域的用户应用，因为它们与Spring的实现紧密相关；或者，它们本身就作为Spring框架的应用案例，体现了许多使用Spring的技巧。这些都是我们开发应用的理想参考，比如ACEGI安全性框架和petclinic应用案例。一方面，可以把这些实现作为应用的一个基本方案加以裁剪，以满足特定领域的需求；另一方面，通过剖析这些应用，可以为应用开发提供很好的参考，提高开发效率。

从更深层次的技术层面上来看，因为Spring是基于Java语言的应用平台，如果我们能够对作为Spring的运行环境的Java计算模型（比如JVM的实现原理）有一些了解，将会加深我们对Spring实现原理的理解。反射机制、代理类、字节码技术等这些JVM特性都是在Spring实现中会涉及的一些Java计算环境的底层技术。一般的应用开发人员可能不会直接从事与JVM底层实现相关的工作，但是这些计算环境的底层知识对我们深入理解Spring是不可缺少的。

在本书的写作过程中，VMware公司收购了Spring的运营者Spring Source。此次商业收购一方面反映了开源软件中蕴含着的巨大商业价值，另一方面反映了Spring的发展趋势：与云计算融合，以及Spring为自己规划的云计算战略——成为PaaS（Platform As a Service）服务的有力竞争者。在云计算这个全新的计算环境中，如何在发挥Spring在企业级应用开发中已经具备的优势的基础上，

为云计算应用的开发提供高可靠性、高可用性、高可扩展性和高性能的应用平台，是Spring团队面

临的全新挑战。Spring在这个崭新领域的雄心和作为让我们充满好奇和期待，我们拭目以待！

闲话说了这么多，很多读者可能已经有些迫不及待了，也许只有对Spring的实现身临其境地接触才是真实的，这里太多的文字已经成为一种累赘。本书将带领你到Spring的核心实现这个茂密而又充满生机的源代码丛林中去大胆地探寻和小心地求证。在这里，你会惊奇地发现：这个过程就像是阅读优美的散文或情节跌宕起伏的小说一样，是与开源软件开发者以及开发者社区的一种畅快淋漓的交流和对话，让人如痴如醉。

本书面向的读者

学习Java语言和Java EE技术的中高级开发者

Spring是使用Java语言实现的，很多功能的源码实现都极其优秀，非常具有研究和参考价值。对这部分读者来说，不仅可以从本书中了解到Spring的实现原理，而且还能通过Spring的源代码，掌握一流的Java编码技巧和Java EE开发技术。

Spring应用开发人员

如果要利用Spring进行高级应用开发，抑或是相关的优化和扩展工作，仅仅掌握Spring的配置和基本使用是远远不够的，必须要对Spring框架的设计思想、架构和运作机制有一定的了解。对这部分读者而言，本书将带领他们全面了解Spring的实现原理，从而加深对Spring框架的理解，提高自己的开发水平。同时，本书可以作为他们定制和扩展Spring框架的参考资料。

开源软件爱好者

Spring是开源软件中的佼佼者，它在实现的过程中吸收了很多开源领域的优秀思想，同时也有很多值得其他开源软件学习的创新。尤为值得一提的是，本书分析Spring源代码的方式也许值得所有想分析源代码的爱好者们学习和借鉴。通过阅读本书，这部分读者不仅能领略到开源软件的优秀思想，而且还能掌握分析源代码的方法和技巧，从而进一步提高使用开源软件的效率和质量。

平台开发人员和架构师

前面已经反复强调，Spring的设计思想、架构和实现都非常优秀，是平台开发人员和架构师们不可多得的参考资料。

如何阅读本书

本书共分为三部分，分别剖析了Spring的核心、组件和经典应用的实现机理。阅读本书时，首先建议读者建立一个源代码阅读环境，这样一方面可以追踪最新的源代码实现，另一方面可以在阅读的过程中进行各种方式的索引和动手验证，加深对开源软件开发方式的体会。

第1章对如何建立源代码环境做了简要介绍。这些知识不但适用于建立Spring的源代码研究环境，而且还适用于其他的Java开源项目，有一定的普遍性和参考意义。对于不同的项目，其具体

使用的源代码管理工具、代码的位置、权限配置会有一些不同，但是整个源代码的获取过程与Spring是类似的。

第一部分详细分析了IoC容器和AOP的实现，这部分内容是理解Spring平台的基础，适合对Spring的运行机理有深入了解需求的读者阅读。在对AOP实现模块的分析中涉及一些JVM底层技术，这也是读者需要具备的背景知识。

第二部分深入阐述了基于IoC容器和AOP的Java EE组件在Spring中的实现。在这部分内容中可以看到，每一个组件实现的内容基本上都是相对独立的，读者可以结合自己的应用需求选读。比如，如果对Spring Web MVC的实现原理感兴趣，可以阅读第4章；如果对Spring提供的数据库操作的实现机制感兴趣，可以阅读第5章；如果对Spring提供的统一事务处理的实现感兴趣，可以阅读第6章；如果对Spring提供的各种不同的远端调用实现感兴趣，可以阅读第7章。

第三部分讲述了一些基于Spring的典型应用的实现。如果读者对在Spring应用中如何满足应用资源的安全性需求方面的内容感兴趣，可以阅读第8章，本章对为Spring应用提供安全服务的ACEGI框架的实现进行了分析，在深入了解这部分内容的基础上，读者可以根据自己的应用需要定制自己的安全系统。如果想了解一般企业应用的典型实现，比如Web MVC层的应用、数据库操作、O/R映射等特性在Spring应用中的具体使用，可以阅读第9章，本章中的petclinic应用为Spring应用开发提供了一个现实的应用实例，虽然简单，但却相对完整。这个应用实例是Spring团队的作品，是Spring项目的一部分。

读者可登录本书网站（<http://www.springagile.cn>）进行技术交流。

由于水平有限，再加上写作时的疏漏，书中难免还会存在许多需要改进之处。在此，欢迎读者朋友们指出书中存在的问题，并提出指导性意见，不甚感谢。如果大家有任何与本书相关的内容需要与我探讨，请申请加入华章俱乐部^[1]并提出你的问题和看法，我会及时给予回复。最后，衷心希望本书能给大家带来帮助，祝大家阅读愉快！

^[1]<http://groups.google.com/group/HZBOOK-for-readers-and-authors>

致谢

感谢父母、家人和各位好友，他们无私的关爱，是我能够完成本书写作的最基本动力。

感谢机械工业出版社华章分社的杨福川编辑和李俊竹编辑，本书得以呈现在广大读者面前，离不开他们的敬业精神和一丝不苟的努力。

同时，希望能够借此机会，感谢许许多多通过互联网结识的朋友。我们一起通过网络，对Spring的实现原理进行了广泛而深入的讨论。坦率地说，正是这些互动与支持，使我能够坚持下来，将本书的内容逐渐完善。

本书的最初内容以及得以出版的各种机遇，来源于发表在JavaEye（<http://www.javaeye.com>）上的一系列帖子和博客（博客地址：<http://jiwenke.javaeye.com>），在这里，请允许我向立志成为“最棒的软件开发交流社区”的JavaEye表示由衷的敬意。

计文柯

2009年11月

第1章 准备源代码环境

半亩方塘一鉴开，天光云影共徘徊。

问渠哪得清如许，为有源头活水来。

——【宋】朱熹《观书有感》

1.1 安装JDK

在开发Spring应用和分析Spring源代码之前，需要做一些准备工作，其中搭建Java环境是必不可少的。Spring 3.0要求Java 5以上版本，JDK需要1.5或1.5以上版本。如果JDK的版本低于1.5，则可进入网站 <http://Java.sun.com/javase/downloads> 下载最新的JDK安装程序。

提示安装完后，要检查JDK是否配置正确。某些第三方的程序会把自己的JDK路径加到系统PATH环境变量中。这样，即便安装最新版本的JDK，系统还是会使用第三程序所带的旧版本JDK。这种情况下，Java环境可能无法正常运行，手工修改系统PATH路径可以解决这个问题，比如重新设置PATH路径指向新JDK的安装路径。

1.2 安装Eclipse

运行Spring只需要JDK，但是一个好的开发环境和源代码阅读环境可以事半功倍。Eclipse是最流行的Java集成开发环境之一，Eclipse的JDT提供了良好的代码分析功能，它们是在分析Spring的实现原理过程中会用到的基本工具。比如，可以用Eclipse分析Java类和接口的继承关系、查看Java方法的调用关系、搜索代码等。

在Eclipse中分析Java类和接口的继承关系的具体做法如下：在代码区中选择需要的类和接口定义，然后使用右键选取Open Type Hierarchy或按快捷键【F4】，可以在Hierarchy View中看到继承关系，如图1-1所示。

在Eclipse中分析Java方法的调用关系的具体做法如下：在代码区中选择相应的方法定义，然后用右键选取Open Call Hierarchy或者按下快捷键【CTRL+ALT+H】，可以在Call Hierarchy视图中看到方法的调用关系，提供了逐层的方法调用追溯的功能，对查找方法的相互调用关系非常有用，如图1-2所示。

在Eclipse中使用搜索功能，使用菜单上的Search可以打开搜索对话框，搜索结果在Search View中显示。如果双击Search View中的搜索结果列表中的某一项，就可以直接打开该搜索结果对应的出处，具体如图1-3所示。

- 1-1 在Eclipse中查看类的继承关系
- 1-2 在Eclipse中参看方法调用关系
- 1-3 在Eclipse中使用搜索功能

1.3 安装辅助工具

作为集成的IDE环境，Eclipse为许多相关的插件提供了应用平台。例如，SVN的Eclipse插件为开发者提供了很好的源代码管理功能，Spring的Eclipse IDE插件为开发Spring应用提供了很好的帮助。对应的插件在Eclipse中安装以后，这些功能就可以直接在Eclipse IDE环境中使用；通常这些插件是作为Eclipse的一个View或者一个Perspective出现在集成IDE环境中的。下面我们来看看这些插件在Eclipse中的安装。我们先对在Eclipse中安装SVN插件做一个简要的说明，关于Spring Eclipse IDE的安装和简要使用，将在第8章中详细介绍。

关于源代码管理工具的具体使用，本书就不做详细的说明了，记得早在10几年前我使用的版本管理工具是CVS。当时是在一台SUN Ultra60工作站上与版本管理工具CVS有了第一次亲密接触。它有一个简单的图形界面，非常直观，使用起来也很方便，从那时候起，我才开始有了源代码管理的基本概念。后来发现，这些概念在其他工具上也是同样适用的，例如版本、分支、基线、合并、比较等。

慢慢更深入地接触了软件产品开发以后，知道这些操作都是在进行软件配置管理计划时需要定义的基本过程，这些基本过程保证了软件产品在协同开发和构建过程中的一致性，成为软件开发中不可缺少的变更管理的重要组成部分。无论是使用CVS、SVN还是ClearCase，是使用命令行方式还是图形界面的方式，基本的概念都需要深刻的理解和掌握。ClearCase在大型商业软件开发领域被普遍使用，而CVS和SVN却是开源领域和中小型企业应用开发领域的主角。

值得注意的是，CVS和SVN本身也是开源软件，提供的功能已经足以满足开源软件开发的需求。这里有个有趣的问题就是，不知道CVS和SVN本身的开发项目是不是用自己做的代码管理？感兴趣的读者不妨去做个调研来告诉大家。如果读者对开源软件的源代码管理感兴趣，可以阅读Open Source Development with CVS（作者Karl Fogel和Moshe Bar）这本书，书中详细介绍了如何使用CVS对开源软件的源代码进行管理。

Eclipse中自带的源代码管理客户端就是CVS的客户端，Eclipse使用者可以直接使用。此外，这个CVS客户端因为有Eclipse的各种视图的支持，所以使用起来也是非常的方便。但是，由于Spring 3.0的源代码是用Subversion来进行管理的，而在Eclipse的默认安装中是不带SVN客户端这个插件的，所以这里有必要对SVN客户端的安装和使用做简要的介绍。这些源代码管理工具是开源软件开发中常用的，所以掌握这些工具的基本使用对我们理解开源软件的开发方式有很大的帮助。我们在这里使用的是一个开源的SVN客户端—Subclipse，一个在Eclipse IDE中使用的SVN插件。其官方网站是<http://subclipse.tigris.org>，其下载页面如图1-4所示。

□ 1-4 SVN客户端下载

在Eclipse的Help菜单下选择Software Updates，如图1-5所示。

打开Software Updates and Add-ons对话框，选择Available Software面板，可以看到在我们的Eclipse IDE环境中已经安装的插件，如图1-6所示。

单击Add Site.....按钮打开Add Site对话框。具体的Eclipse更新地址可以在<http://subclipse.tigris.org>/单击Download and Install页面找到，我们在Location中输入插件更新地址，结果如图1-7所示。

- 1-5 在Eclipse中安装插件
- 1-6 在Eclipse中安装插件
- 1-7 在Eclipse中安装插件

单击OK按钮关闭Add Site对话框。Eclipse获取到插件列表，更新Software Updates and Add-ons对话框的内容，并选择Subclipse update sites，如图1-8所示。

- 1-8 在Eclipse中安装插件

欢迎访问：电子书学习和下载网站（<https://www.shgis.cn>）

文档名称：《Spring技术内幕——深入解析Spring架构与设计》计文柯.epub

请登录 <https://shgis.cn/post/1852.html> 下载完整文档。

手机端请扫码查看：

