

博文视点原创精品大系

Java 专家

Spring 2.0

技术手册

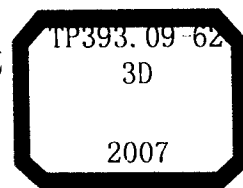
- 介绍Spring 2.0的新特性
- 从实践中了解Spring IoC、AOP
- 完整范例展示，减少摸索时间
- 项目实践：Spring在线书签

林信良 著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>





Spring 2.0 技术手册

林信良 著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书介绍了 Spring 2.0 的新特性, 诸如 Spring 2.0 的新配置、新 AOP 支持、增强的 IoC、JDBC 和 form 标签等新功能。它通过实际完成一个完整的 Spring 项目示例, 展示了与 Spring 相关 API 的使用技巧, 能够显著减少每一位入门者摸索 Spring API 的时间。本书是学习 Spring Web 开发的最佳读物, 它能让读者在示例学习中获得显著提高。

本书内容全面深入, 主要包括 Spring 入门、Bean/消息/事件、Spring AOP、JDBC/事物支持、Spring 与 Hibernate 的整合、Spring Web MVC 框架、View 层方案/Web 框架整合、API 封装、Spring 在线书签完整项目示例等内容。

本书为台湾基峰资讯股份有限公司独家授权发行的中文简体版。本书中文简体字版在中国大陆之专有出版权属电子工业出版社所有。在没有得到本书原版出版者和本书出版者书面许可时, 任何单位和个人不得擅自摘抄、复制本书的一部分或全部以任何方式(包括资料和出版物)进行传播。本书原版版权属基峰资讯股份有限公司。版权所有, 侵权必究。

版权贸易合同登记号 图字: 01-2007-1376

图书在版编目(CIP)数据

Spring 2.0 技术手册 / 林信良著. —北京: 电子工业出版社, 2007.4
ISBN 978-7-121-03985-0

I. S… II. 林… III. JAVA 语言—程序设计—技术手册 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 031880 号

责任编辑: 周 筠 杨绣国

印 刷: 北京市铁成印刷厂

装 订: 北京中新伟业印刷有限公司

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

开 本: 787×980 1/16 印张: 28.75 字数: 550 千字

印 次: 2007 年 4 月第 1 次印刷

定 价: 49.80 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 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; 邮购电话: (010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

导读

这份导读可以让您更了解如何使用本书。

本书许多的范例都通过完整的程序实现来展示，在以下的程序代码范例中：

SpringDemo

HelloBean.java

```
package onlyfun.caterpillar;

public class HelloBean {
    private String helloWord;

    public void setHelloWord(String helloWord) {
        this.helloWord = helloWord;
    }

    public String getHelloWord() {
        return helloWord;
    }
}
```

范例左上角的名称“SpringDemo”表示可以在光盘中找到对应的 SpringDemo 项目，而右上角的名称“HelloBean.java”表示可以在项目中找到 HelloBean.java 文件。

如果按以下方式呈现代码，表示它是一个完整的程序，但不属于范例项目的一部分，主要用来展示如何编写一个完整的文档：

```
helloBean.class=onlyfun.caterpillar.HelloBean
helloBean.helloWord=Welcome
```

如果按以下方式呈现代码，则表示它是个程序代码片段，在编写程序时需要特别注意：

```
...
<beans...>
    <bean id="beanFactoryModifier"
        class="onlyfun.caterpillar.SomeClass"/>
```

```
...  
</beans>  
...
```

由于书籍的页面宽度有限，程序代码有时会因为过长而无法容纳于书籍的一行之中，此时不得不隔行表示，这时会使用“→”符号，表示两行实际上是必须连接在一起的，例如：

```
...  
<bean id="configBean"  
    class="org.springframework.beans.factory.  
        → config.PropertyPlaceholderConfigurer">  
    <property name="location">  
        <value>hello.properties</value>  
    </property>  
</bean>  
...
```

在上面的程序片段中，“class”属性的完整程序代码编写实际上应该是 `org.springframework.beans.factory.config.PropertyPlaceholderConfigurer`。

在本书中会出现以下的对话框：

良葛格的话匣子<<<

在这个对话框中提供的可能是良葛格的一些主观性较强的意见、经验或建议，在这个对话框中也有可能提供一些主题外的常识，或一些额外的参考文件。

本书提供符合 Eclipse 项目结构的源文件，而对于 Web 应用程序的范例，则提供符合 Eclipse Web Tools Platform (WTP) 项目结构的源文件。

读者对《Spring 技术手册¹》的赞誉

“我很喜欢第 2、3、6、8 这几章。在看此书之前我看了一些 Spring 的资料，可有些东西还是不很清楚。看过本书后有种豁然开朗的感觉，原来不懂的一些东西一下子就全明白了，感觉很不错。”

——沈阳市读者

“最喜欢第 3、4、5、6、9、10 章，以及第 8 章的部分内容。此书易懂，真的很棒，个人觉得《Spring 技术手册》是市面上最易懂的 Spring 书籍（注意，没有之一）。”

——大连市读者

“最喜欢‘Bean、消息、事件’这一章，内容充实，实例鲜明！”

——哈尔滨市读者

“很喜欢第 6、8 章，因为这两章涉及 Spring 和 Hibernate、JSF 相结合的内容，对现实应用有很好的帮助。”

——北京市读者

“其实，这本书的每章都非常棒。特别是里面的示例，很简单也很容易实现，很棒！”

——南京市读者

“我最喜欢第 7 章中讲的 Spring Web MVC 框架，它让我弄清楚了从请求到请求分派到 Controller 实现类的过程，谢谢！”

——广州市读者

“我最喜欢第 4 章的 AOP，以及第 7 章 Spring 的 MVC 内容。”

——汕头市读者

¹ 本书是对畅销书《Spring 技术手册》的全新改版，涵盖 Spring 2.0 的新特性。

“林信良的作品肯定不错，非常适合初学者，刚入门 Spring 的程序员非常适合看这本书！这本书带我进入了 Spring 的世界。”

——www.dearbook.com 读者

我看了 4 本已出版的与 Spring 相关的书籍：《Spring 高级框架设计》（自购）、《Spring in Action》（借阅）、《Spring 技术手册》（自购）、《精通 Spring》。

对于像我这样的 Spring 初学者（懂点设计模式，也有一定的代码积累和一点项目经验）来说，最理想的书可能就是《Spring 技术手册》。可不要被书名迷惑，这可决不是一本册子这么简单，它对 IoC、AOP 的分析是这 4 本书中最容易懂的了（作者在讲解之前做了一些技术和概念的铺垫），如果你是初学者，强烈推荐。

——www.china-pub.com 读者

夏昕、林信良谈 Spring 框架

CSDN、博文视点名家讲坛邀请夏昕先生和林信良先生针对 Spring 框架进行了在线访谈。现将谈话内容整理成文，以飨读者。

采访人：《程序员》杂志的技术编辑霍泰稳

受访人：夏昕、林信良

采访方式：CSDN 聊天室在线访谈 <http://chat.csdn.net>

♣ 主持人：

大家好，随着夏昕老师的译作《Spring 专业开发指南》、林信良老师的《Spring 2.0 技术手册》的上市，Spring 继续成为大家关注的热点。今天我们有幸请到两位老师来参加 CSDN F2F 俱乐部和博文视点资讯有限公司共同举办的名家讲坛活动，本次活动会就“轻量级框架——Spring”这一话题与大家在线交流。

这里先简单介绍一下两位专家的背景：

夏昕是业界知名的技术专家、金融行业的技术顾问、OpenDoc 项目发起人及《DrBobb's Journal China》的常任编委，著有广受好评的畅销书《深入浅出 Hibernate》。

林信良老师（网名为良葛格）曾是台湾 SUN 教育训练中心讲师，著有《Spring 2.0 技术手册》、《JAVA 学习笔记》。他是 SUN 官方教材的主要编写成员之一。林老师的博客地址是：http://blog.csdn.net/caterpillar_here/。

下面请夏昕老师、林信良老师开始回答网友们的提问。

♣ 主持人：

第一个问题，新手到达什么程度可以开始 Spring 的学习？

♣ 夏昕：

一般来说，如果拥有一年多持续的 Java Web 应用开发经验，学习 Spring 基本上就没太大问题。目前公司团队中，几乎所有成员都是从进入公司开始学习应用 Spring 的。从程度来看，一年的实际 Web 开发经验已足以打下理解 Spring 基础结构的基础。这时候，只要有适当的资料加以辅助，学习起来就不会太难了，呵呵。

♣ 主持人:

网友提问, Spring 能够粘合不同层面的解决方案, 但无论是 MVC 层还是 ORM 层, Java 领域从来都不缺少好的框架, 我们在使用 Spring 时, 在这些框架中作选择的依据是什么?

♥ 林信良:

选择框架时, 主要是考虑稳定性。例如 Struts, 虽然它是个相对古老的框架, 但稳定性高。也许就现在的设计观点来说它旧了, 但开发不是追求流行, 不是说最新的框架就是最好的。另外, 选择框架的一个重点常被忽略, 那就是人员培训。一个好但复杂的框架或一个好但缺乏文档的框架, 能发挥它的功用吗? 开发不是一两个人的事, 而是众人的事。我看过很多公司为了追求流行而忽略了这一点。

♣ 主持人:

网友提问, 与仅仅使用 JSP 或 Struts 等简单框架进行开发相比, 使用 Spring 的 Web 应用性能如何?

♣ 夏昕:

客观来讲, Spring 的微内核容器的确会在一定程度上消耗额外的系统性能。不过, 除非是开发具有并发性能(比如超大规模论坛)、实时性能(比如超大规模信用卡在线交易系统)等关键业务系统, 否则我还是推荐使用 Spring。Spring 为我们带来的系统结构和开发上的帮助, 远远可以抵消这部分损失。

♣ 主持人:

网友提问, 夏老师是金融行业的顾问, 想问一下, 您是怎么说服金融行业的人员使用开源产品的, 我所遇到的一些金融行业从业人员对开源甚至 Java 都不愿意接受, 他们更喜欢古老但坚固的 C++。

♣ 夏昕:

这可能是传统思路的延续吧, 跟国内金融系统一贯的风险规避思路有关。我为花旗银行亚太数据中心工作的时候, 看到很多特关键的业务, 其后台都是 Tomcat 在运行。所以, 开源产品本身并没有什么问题, 关键还在于思路的改变, 这需要时间。

♣ 主持人:

网友提问, 学习 J2EE 开发的人, 不学习 Spring 可以么?

♥ 林信良:

可以啊! 并没有规定一定要学过 J2EE 才可以学习 Spring, 也没有规定一定要学过 Spring 才可以学习 J2EE。在学习 Spring 甚至是一些 opensource project 时, 重点是了解其精神。学习 Spring 有助于理解 J2EE 开发中的设计, 事实上, opensource 界在 Java 的设计精神上取得了很大的胜利,

在未来的 Java EE 中, 有很多都会借鉴 open-source 界的经验, 比如采用 Spring/Hibernate 的一些设计精神。在这种情况下, 学习 Spring, 或者学过 Spring/Hibernate 等, 多少都会有助于了解将来的 Java EE。

♣ 主持人:

网友提问, 什么情况下适合将 Spring 纳入项目中加以应用(最适合 Spring 应用的场景有哪些)?

♥ 林信良:

使用 Spring, 最怕的是为了 Spring 而 Spring, 为了 IoC 而 IoC, 为了 AOP 而 AOP, 这样的话, Spring 对您没有好处。有几点可以参考:

1. 当组件出现重用, 再开始考虑使用 Spring IoC。
2. 当组件用到外部资源时, 使用 Spring IoC。
3. 当组件需要 AOP 时, 再考虑设置 Spring xml。
4. 能不用 Spring IoC 就不用!
5. 不要为 Spring 而 Spring。
6. 不要为 IoC 而 IoC。
7. 不要为 AOP 而 AOP。

简单地说, 采用 XP 的方式可以让您考虑何时该采用 Spring, 何时该采用 IoC、AOP 等。这里有个案例可以参考: http://www.javaworld.com.tw/roller/page/ingramchen?entry=2006_2_1_KeepSpringIoCcoarseGrained。另外, Spring 在 JDBC 这方面所提供的封装也值得分来看看, 可以省去不少功夫。

♣ 主持人:

网友提问, Spring 里面的接口编程有哪些好处? 如果用类来实现 Service, DAO 层同样可以实现注入吗? 为什么要接口呢? 有的资料说可以重用代码, 但是对于一般的 Service, DAO 的接口就只被一个类实现, 何谈重用呢?

♣ 夏昕:

这就跟电脑上的 USB 接口一样, 一般这个接口只会插鼠标, 如果什么时候需要用 U 盘了, 插上也可以。Spring 提供了这种实现的基础, 至于重用性本身是否存在, 那可能就须要在开发项目前进行调研了。比如做一个只用一次的投票表单, 用一个 JSP 搞定更干净利落, 这时重用性并不重要。

♣ 主持人:

网友提问, 其公司的业务一般都使用最简单的 JSP+Beand 技术。这样, 接触 Struts 和 Spring 的机会就很少了, 可否请林老师提些建议, 如何在这种环境下学好 Spring。

♥ 林信良:

公司如果不使用 Spring, 并不是什么坏事, 我看过因为用了 Spring/Hibernate 而使得学习成本大幅提高的情况。如果你们公司业务完全不用 Spring 或 Sturts 等技术, 那您就需要个人学习 Spring 了, 没必要为了用这些东西在公司大力倡导。您可以自学, 将来有机会自然就会用上, 到时您所学的知识就可派上用场了。

♣ 主持人:

网友提问, 请问夏老师对 EJB 3.0 怎么看, EJB 3.0 从开源框架里汲取了不少灵感, 甚至有一些开源大师也参与其中, 那 EJB 3.0 会和 Spring、Hibernate 形成竞争关系吗?

♣ 夏昕:

EJB 3.0 实际上包括了 Hibernate, 所以它们不存在设计思想上的竞争。至于 Spring, 竞争可能会比较激烈一点, 呵呵。

♣ 主持人:

网友提问, Spring 框架是否是更适用于 B/S 项目?

♥ 林信良:

Spring 框架提供各种子框架, 并不局限于 B/S 项目, 您也不用将 Spring 中的子框架全部用上, 您可以根据项目需要, 采用其中适当的子框架。不必为了用 Java 而把 Java SE 的所有 API 全部都用上, 不是吗?

♣ 主持人:

网友提问, 是否可以通俗地解释一下“依赖注入”, 谁依赖谁了?

♥ 林信良:

依赖注入指的是, 当您在 A 对象内部需要 B 对象的某个功能时, A 就依赖于 B, 您可以直接在 A 内部实例化 B, 但日后要修改就麻烦些。如果您不直接在 A 内部实例化 B, 而是通过 IoC 将一个 B 实例从外部设置给 A, 那就比较有弹性。举个通俗例子, 如果将 CPU 固定在主板上, 那主板就依赖于 CPU, 若保留 CPU 的插槽规格, 让 CPU 可以自由更换, 就有点 IoC 的味道了。

♣ 主持人:

网友提问, Spring 的 MVC framework 不算是整个 Spring 的核心部件, 但它的威力却超出了很多人的想象。很多人认为 Spring 的 MVC framework 是非常优秀的, 甚至比 Webwork 2 更优秀。两位老师是如何评价的呢?

♥ 林信良:

就我个人而言, 使用 Spring MVC 最大的好处是, 可以直接整合并善用其 IoC 容器功能, Spring MVC 与 Struts 的设计其实是蛮像的, 至于其他部分, 我并不觉得 Spring MVC 有什么特别突出之处。

♣ 夏昕:

呃, 我觉得 Spring MVC 是以干练取胜, 它能以极其简炼的办法实现 Control 逻辑。喜欢简洁的人会很喜欢 Spring MVC 的实现方式。但是它在某些方面还是存在一些缺陷, 比如 wizard 的 backspace 问题一直没能很好地解决。另外, 在实际应用过程中跟 Spring Webflow 的配合也总是比较尴尬。我们公司目前选择的是 Webwork+Spring+Hibernate 的组合。

♣ 主持人:

网友提问, Spring 和轻量级、轻量级和重量级的概念是如何划分的? Spring 对 JTA、EJB3、JMS、WebService 都支持, 而且也封装得很好。

♣ 夏昕:

现在轻量级和重量级的确很难区分了, 标准越来越多, 也越来越难以衡量。之前有人说, 有 classloader 的算是重量级; 也有人说, 超过 3MB 的算是重量级, 真的很难说了。EJB 有轻量化的倾向, Spring 也有重量化的趋势。简单说吧, Spring core 算是轻量级的, 但是加上其余的部件, 估计就得列入重量级的范畴了, 呵呵。

♣ 主持人:

网友提问, 怎么设计系统架构才能成功实现轻量级开发? 是不是用了 Spring 就是轻量级开发了呢?

♥ 林信良:

我觉得这个问题提得非常好! 常有人问, 这个功能用 Spring 怎么实现? 那个功能用 Spring 如何实现? 事实上, 我心里想的就是, 要实现某个功能, 自己写个组件会比直接用 Spring 来得好! 用了 Spring 并不等于轻量级开发。重点是您有没有采纳其精神, 用了 IoC 容器就轻量级了吗? 不见得。什么都来个 DI, 我看也是很沉重的。如果您在设计时善用 IoC, 即使不用 Spring 也是轻量级。

♣ 主持人:

网友提问, 现在 Spring 添加了许多的功能, 这样下去岂不是也变得不轻量了么?

♥ 林信良:

您什么都要用 Spring 来做, 那就是重量级了。还是那句话, 不要为了 Spring 而 Spring! 如果 Spring 开始让您觉得沉重, 那它就不是个适用的框架, 谨记! 如果在采用 Spring 前, 您的 team 就开始叫苦连天, 而且还要先花一堆时间去学, 但又无法入门, 那就不要用。

♣ 主持人:

网友提问, Spring 和 Struts 在运用方面各有什么特点?

♥ 林信良:

Spring 与 Struts 基本上是两个目的不同的框架 您要比的应该是 Spring MVC 与 Struts 的特点! Struts 的特点是成熟, 它已经是 Model 2 的代表了, 它的可用组件多、Tag 丰富。而 Spring 的 View 层可以适当地抽换, 使用 JSTL 可以整合其 IoC 功能等, 这就是其特点。

♣ 主持人:

网友提问, 如果现在要开发一个新的 B/S 项目, 两位推荐的 MVC 组合是什么? (? + Spring + ?)

♥ 林信良:

您指的已经不只有 MVC, 而是层与层的设计了, Web 层现在有很多选择了, Struts/Webwork 2 都是不错的, 至于 Persistence 层, 现在主流是 Hibernate, 而学习 Hibernate, 有助于了解未来的 EJB 3.0, 这点或许可以列入考虑中。

♣ 主持人:

网友提问, 请问夏老师☺, 怎样修炼才能成为系统架构师?

♣ 夏昕:

这里就不说具体的方面了。我觉得, 首先是要有好奇心和钻研精神。大家今天能来这里, 相信都不缺乏这些, 另外很重要的一点就是总结和分析的能力。之前我发布的那些 open .doc, 都是在研究某个框架时随手记录的一些经验总结, 类似每天一篇 blog 的形式, 最后汇总起来, 也可以成为一本书籍。《深入浅出 Hibernate》就是这么诞生的。有什么收获, 写下来, 与大家分享, 可以使自己进步更快。这一点是我最大的体会。

♣ 主持人:

网友提问, 请问夏昕老师是哪里人呀? 请您说说您的 Java 学习之路。

♣ 夏昕:

呵呵, 扬州人。在学习 Java 之前, 做过几年 VC++ 开发, 后来转到 Java 也是很奇怪的事情。好像是 Java 项目组里的人不够了, 刚好这时候自己的事已做完, 于是也就凑个热闹写了两个 Java Socket 程序。后来 Java 项目组里的人总是不够, 于是……呵呵。基本算是在实战中学习的吧。

♣ 主持人:

非常感谢两位老师的精彩回答与奉献精神, 敬礼!!!

序

对我来说，写书是一种想法的实现过程。书稿与我在网络上所放置的笔记文件不同，笔记文件只是学习过程的一个记录，而在写作成书时，需要考虑以什么方式呈现书的内容、呈现哪些内容，以及传达什么样的信息。

在写作《Spring 技术手册》时，我的想法是总结从实践中学习的成果，当 Spring 2.0 接近发布之时，我又在思考：驱动《Spring 技术手册》改版的想法是什么。

《Spring 技术手册》问世之后，收到了很多建议与批评，也有幸受邀参与了 Spring 相关课程的交流讨论。从这些回馈与交流当中，我逐步明确了将《Spring 技术手册》改版为《Spring 2.0 技术手册》的思路与轮廓。

最开始的想法是在改版时，要将 Spring 2.0 的所有特性详加说明，然而一来个人能力有限，二来 Spring 的参考文档中已将自己想到的很多东西论述得很清楚了，若再要置喙，只是沦于抄写翻译之工而已。所以最后决定重点介绍 Spring 2.0 的新特性。另一个想法是，在几次 Spring 相关课程的交流中，我有了一些教学经验、观念与实践方式的积累，想将这些东西加入书中。

《Spring 2.0 技术手册》仍然定位于初学者，它是引导初学者入门的敲门砖，希望初学者通过这块敲门砖，得以进入 Spring 的大门，进而探索门后的宝库。《Spring 2.0 技术手册》之目的，就是让您在阅读完之后，能够自行查阅 Spring 参考文档，看懂更进阶的书籍，以及参与网络上关于 Spring 2.0 的深入讨论，真正踏上 Java 开发之路。

林信良

2006 年 11 月 9 日

目录

导 读	1
第 1 章 认识 Spring	1
1.1 术语介绍	2
1.2 控制反转 (Inversion of Control)	4
1.3 依赖注入 (Dependency Injection)	9
1.4 接下来的主题	11
第 2 章 Spring 入门	13
2.1 第一个 Spring 程序	14
2.1.1 下载、设置 Spring	14
2.1.2 准备 Spring 设计环境 (使用 Eclipse)	16
2.1.3 编写第一个 Spring 程序	20
2.1.4 编写第二个 Spring 程序	23
2.1.5 Spring 2.0 的 XML 设置	26
2.2 安装、使用 Spring IDE	27
2.2.1 安装 Spring IDE	27
2.2.2 使用 Spring IDE	29
2.3 接下来的主题	32
第 3 章 Bean、消息、事件	35
3.1 Bean 基本管理	36
3.1.1 BeanFactory、ApplicationContext	36
3.1.2 Bean 的识别名称与别名	39
3.1.3 Bean 的实例化	40
3.1.4 Bean 的 scope	42
3.1.5 Bean 的生命周期	43

3.1.6 Bean 定义的继承.....	45
3.2 Bean 的依赖设置	48
3.2.1 Type 2 IoC、Type 3 IoC.....	48
3.2.2 依赖的值设置与参考	52
3.2.3 自动绑定	55
3.2.4 集合对象	60
3.2.5 Spring 2.0 的<util>标签	68
3.2.6 Lookup Method Injection.....	71
3.3 Bean 高级管理	73
3.3.1 非 XML 定义文件的配置方式	74
3.3.2 Aware 相关接口	76
3.3.3 BeanPostProcessor	78
3.3.4 BeanFactoryPostProcessor	81
3.3.5 PropertyPlaceholderConfigurer.....	81
3.3.6 PropertyOverrideConfigurer	83
3.3.7 CustomEditorConfigurer	84
3.4 资源、消息、事件	88
3.4.1 资源的取得.....	88
3.4.2 解析文字消息	89
3.4.3 监听事件	92
3.4.4 事件传播	93
3.5 接下来的主题	95
第 4 章 Spring AOP.....	97
4.1 AOP 入门	98
4.1.1 从代理机制初探 AOP	98
4.1.2 AOP 观念与术语	105
4.1.3 Spring AOP	108
4.2 Advices	109
4.2.1 Before Advice.....	109
4.2.2 After Advice	113
4.2.3 Around Advice	115
4.2.4 Throw Advice.....	117
4.3 Pointcut、Advisor	121

4.3.1	NameMatchMethodPointcutAdvisor.....	121
4.3.2	RegExpMethodPointcutAdvisor	124
4.3.3	ControlFlowPointcut	126
4.3.4	Pointcut 接口	129
4.3.5	Pointcut 交集、并集操作.....	131
4.4	Introduction.....	132
4.4.1	IntroductionInterceptor.....	133
4.4.2	DelegatingIntroductionInterceptor	137
4.5	Autoproxing	141
4.5.1	BeanNameAutoProxyCreator.....	141
4.5.2	DefaultAdvisorAutoProxyCreator.....	142
4.6	Spring 2.0 的 AOP 支持.....	144
4.6.1	Before Advice: 基于 XML Schema	144
4.6.2	Before Advice: 基于 Annotation.....	147
4.6.3	Spring 2.0 的 Pointcut 定义	149
4.6.4	After Returning Advice: 基于 XML Schema	151
4.6.5	After Returning Advice: 基于 Annotation.....	153
4.6.6	After Throwing Advice: 基于 XML Schema.....	155
4.6.7	After Throwing Advice: 基于 Annotation	158
4.6.8	Around Advice: 基于 XML Schema.....	159
4.6.9	Around Advice: 基于 Annotation	161
4.7	接下来的主题	162
第 5 章	JDBC、事务支持.....	163
5.1	Spring 持久层入门.....	164
5.1.1	Spring 的 DAO 支持.....	164
5.1.2	DataSource 注入	169
5.1.3	DataSource 置换	175
5.2	JDBC 支持.....	177
5.2.1	使用 JdbcTemplate.....	177
5.2.2	执行与更新 JdbcTemplate.....	179
5.2.3	JdbcTemplate 查询.....	183
5.2.4	JdbcTemplate 的 Lob 支持	185
5.2.5	以对象方式进行操作	187