

编著
监制

[illegible]畅销书
升级

精通Hibernate:

Java 对象持久化技术详解

(第2版)

Mastering Hibernate

Mastering Hibernate

Let Java objects

hibernate in the relational database.



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

CD-ROM
包含配套Hibernate教学讲座



孙卫琴

飞思科技产品研发中心

编著
监制

IBER. ALEMA. TERII
NATEMASTERING F
JEM**STERING HI
ERNATE**TERING
MASTERING HIB RNATEM AST. RING HI
RNATE MASTERING HIBEI NATEMA TE. NG HIBE
ERNATE MASTER' .G HIBERNATEMASTER' NG HIBER
NG HIBERNATE MAS' ERING HIBE NATEMASTERING HIB
G HIBERNATE MAST' RING HIBERN. TEMASTERING HIBI
NG HIBERNATE MASTE JING HIBERNATE. MASTERING HIBI
RING HIBERNATE MAST RING HIBERNATEMASTERING HI
ERING HIF RNATE MAS' ERING HIBERNATEMASTERING F
ERING HIBI NATE MASTI RING HIBERNATEMASTERING H
STERING HIB RNATE MASTERING HIBERNATEMASTERING
TERING HIBE NATE MASTERING HIBERNATEMASTERING B.
TERING HIBEI NATE MASTERING HIBERNATEMASTERING IBI
MASTERING HIBE NATE MASTERING HIBERNATEMASTER' G HIB
STERING HIBERNATE MASTERING HIBERNATEMASTER' G HIBE
STERING HIBERNATE MASTERI G HIBERNATEMASTER' NG HIBE
STERING HIBERNAT MASTERIN G HIBERNATEMASTER' JING HIBE
TERING HIBERNA' MA MASTERING HIBERNATEMA' ERING HIBE
STERING HIBER' MASTER' G HIBERNATEI TERING HIBE
MASTERING HIB' MASTERI G HIBERNAT TERING HIBI
MASTERING I E MASTE G HIBERNA TERING HIB
TERING P MASTER' IBERNATI NG HIBER
TERING I MASTERI ERNATEM. G HIBERNA
MASTERI MASTER. BERNATEM. NG HIBERNA
IASTI RIN G H MASTER. BERNATI MA G HIBER' AT

精通 Hibernate:

Java 对象持久化技术详解
(第2版)

Mastering Hibernate

Let Java objects
hibernate in the relational database.

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

· 内 容 简 介 ·

Hibernate 是非常流行的对象-关系映射工具。本书详细介绍运用目前最成熟的 Hibernate 3.3 版本进行 Java 对象持久化的技术。Hibernate 是连接 Java 对象模型和关系数据模型的桥梁,通过本书,读者不仅能掌握用 Hibernate 工具对这两种模型进行映射的技术,还能获得设计与开发 Java 对象模型和关系数据模型的先进经验。本书将帮助读者编写出具有合理的软件架构,以及好的运行性能和并发性能的实用 Hibernate 应用。书中内容注重理论与实践相结合,列举大量具有典型性和实用价值的 Hibernate 应用实例,并提供详细的开发和部署步骤。

随书配套光盘内容为本书所有范例源程序、本书涉及软件的最新版本的安装程序,以及由作者亲自制作的与本书配套的语音讲座。

本书无论对于 Java 开发的新手还是行家来说,都是精通 Java 对象持久化技术的必备实用手册。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

精通 Hibernate: Java 对象持久化技术详解/孙卫琴编著. 2 版. —北京: 电子工业出版社, 2010.2
(Java 开发专家)

ISBN 978-7-121-09373-9

I. 精… II. 孙… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 133391 号

责任编辑: 郭 晶 田 蕾

印 刷: 北京东光印刷厂

装 订: 三河市皇庄路通装订厂

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

开 本: 787×1092 1/16 印张: 44.75 字数: 1148.8 千字 彩插: 1

印 次: 2010 年 2 月第 1 次印刷

印 数: 5 000 册 定价: 75.00 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,联系及邮购电话:(010) 88254888。

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

服务热线:(010) 88258888。

“开发专家之 Sun ONE”全新提升为“Java 开发专家”系列
——源自精品 成就理想

出版说明

★ 从“开发专家之 Sun ONE”到“Java 开发专家”

“开发专家之 Sun ONE”系列丛书从诞生之日至今，已经八岁了。在这个系列里面，我们一直努力体现着这么一个理念：用一种较为敏锐的视角来跟踪 IT 技术的发展轨迹，并把可能为广大程序员所希望获得的知识，用图书出版的方式奉献给大家。

在这个系列中，我们陆续出版了约 30 种图书，有《Java 与模式》、《JSP 应用开发详解（第二版）》、《精通 EJB（第三版）》、《Tomcat 与 Java Web 应用开发详解》、《精通 Struts：基于 MVC 的 Java Web 设计与开发》、《JBoss 管理与开发核心技术（第三版）》、《精通 Spring》、《精通 Hibernate：Java 对象持久化技术详解》等一大批读者朋友耳熟能详的作品。很多作品都是在国内没有同类图书的情况下出版的。在这几年的出版工作中，我们时刻感受着市场的风险，也时刻收获着无数读者给我们的认可。

在这个系列中，凝聚了大量资深技术专家的心血。有大家都熟知的阎宏、刘晓华、孙卫琴、罗时飞等，还有一些正在不断腾跃的开发高手。这些非常优秀的国内原创作者们一直都在支持着“开发专家之 Sun ONE”系列的出版工作，在这里，我们要向他们说声：谢谢。

桃李不言，下自成蹊。由于这些年“开发专家之 Sun ONE”在“两个效益”中的杰出表现，电子工业出版社授予这个系列“最佳品牌奖”。

时代不断前进，技术不断变革。为了顺应 Java 领域的技术发展态势，为了赋予这个经典的图书系列更强的生命力，我们将“开发专家之 Sun ONE”升级为“Java 开发专家”。我们将继承原有的出版理念，紧密跟踪技术热点和发展趋势，会聚更多优秀作者，全力奉献更经典的作品。

★ 规划你的 Java 开发之路

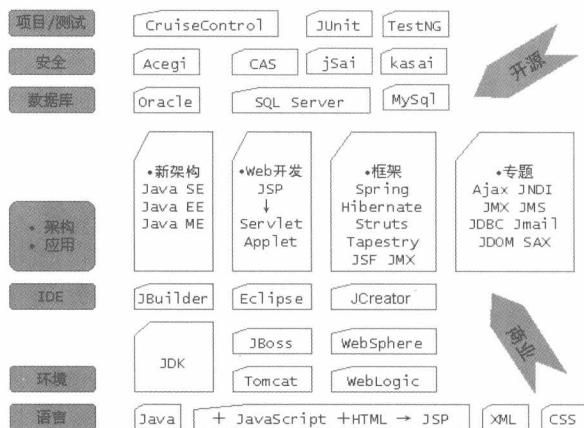
喜马拉雅山脉的最高峰不断地在温室效应中降低，而 Java 世界的颠峰永远都在技术人员的追求中不断升高。每个人都有不同的路，每个人都有不同的行路方式，不过，往往“到了山顶才发现，错误的路和正确的路就差那么几步！”

身处 Java 洪流中的程序员最累（不过大家都说 Java 程序员薪水最高，呵呵），我们简单整理了一下 Java 领域的相关技术、工具、架构，如下图所示。这个框图中的每一个英文单词（或缩写）都可以写成一本书。Java 领域还有一个特点，那就是商业产品和开源产品层出不穷，潮流不断。相比于其他领域，如.NET，Java 开发更是体现了这句谚语：条条大路通罗马。

罗马只有一个，大路却有多条。看上去，似乎到罗马很容易，反正路多嘛。不过，路多却容易迷失方向。当你在 Java 领域中摸爬滚打几年后，发现自己在无数条道路上走了很久，却不知道罗马何日才能到达，甚至连罗马的方向都不知道，这时你肯定会很失落。

很遗憾，在这个简短的出版说明文章里面，我们无法告诉你每一条连贯的、不费周折的通往罗马的道路该如何走。或许，通过“Java 开发专家”系列中的某本书，你可以找到属于你的正确道路。在一般情况下，我们不会就某一项很窄的话题来单独写一本书，我们还是希望通过我们的一些专业和智慧，尽力把一些相关技术整合起来，用较为简明的方式表达出来，最后由你来选择。

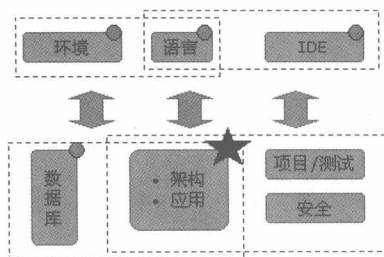
这里有句话与大家共勉：少走弯路，就是捷径！



★ “Java 开发专家”的奉献

犹如在上面那个框图中展现的那样，我们希望在各个层面、各个方向上都能给读者奉献出优秀的图书作品，全面体现技术与应用的结合。从宏观上看，我们会从语言、IDE、环境、数据库、架构与应用、安全、项目与测试等方面进行选择，选出一些读者迫切需要的技术来先行规划。

“Java 开发专家”虽然新蓓初绽，但因其源自盛放的“开发专家之 Sun ONE”系列而根基稳健，两个系列会有一段很长的并行时间，我们会用一种优化的方式来保证读者的顺利选择。无论哪一个系列，必定都有大家喜欢的图书。



在技术上，有着持久化的方法，在学习上，也需要有持久化的精神。从“开发专家之 Sun ONE”到“Java 开发专家”，希望可以带给你持久化的动力。

联系方式

咨询电话：(010) 68134545 88254160
 电子邮件：support@fecit.com.cn
 服务网址：http://www.fecit.com.cn http://www.fecit.net
 通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

在如今的企业级应用开发环境中，面向对象的开发方法已成为主流。众所周知，对象只能存在于内存中，而内存不能永久保存数据。如果要永久保存对象的状态，需要进行对象的持久化，即把对象存储到专门的数据存储库中。目前，关系数据库仍然是使用最广泛的数据存储库。关系数据库中存放的是关系数据，它是非面向对象的。

对象和关系数据其实是业务实体的两种表现形式。业务实体在内存中表现为对象，在数据库中表现为关系数据。内存中的对象之间存在关联和继承关系，而在数据库中，关系数据无法直接表达多对多关联和继承关系。因此，把对象持久化到关系数据库中，需要进行对象-关系的映射（Object/Relation Mapping，简称 ORM），这是一项烦琐耗时的工作。

在实际应用中，除了需要把内存中的对象持久化到数据库外，还需要把数据库中的关系数据再重新加载到内存中，以满足用户查询业务数据的需求。频繁地访问数据库，会对应用的性能造成很大影响。为了降低访问数据库的频率，可以把需要经常被访问的业务数据存放在缓存中，并且通过特定的机制来保证缓存中的数据与数据库中的数据同步。

在 Java 领域，可以直接通过 JDBC 编程来访问数据库。JDBC 可以说是访问关系数据库的最原始、最直接的方法。这种方式的优点是运行效率高，缺点是在 Java 程序代码中嵌入大量 SQL 语句，使得项目难以维护。在开发企业级应用时，可以通过 JDBC 编程来开发单独的持久化层，把数据库访问操作封装起来，提供简洁的 API，供业务层统一调用。但是，如果关系数据模型非常复杂，那么直接通过 JDBC 编程来实现持久化层就需要有专业的知识。对于企业应用的开发人员，花费大量时间从头开发自己的持久化层不是很可行的。

幸运的是，目前在持久化层已经有好多种现成的持久化中间件可供选用，有些是商业性的，如 TopLink；有些是非商业性的，如 JDO 和 Hibernate。Hibernate 是一个基于 Java 的开放源代码的持久化中间件，它对 JDBC 做了轻量级封装，不仅提供 ORM 映射服务，还提供数据查询和数据缓存功能，Java 开发人员可以方便地通过 Hibernate API 来操纵数据库。

现在，越来越多的 Java 开发人员把 Hibernate 作为企业应用和关系数据库之间的中间件，以节省和对象持久化有关的 30% 的 JDBC 编程工作量。2005 年，Hibernate 作为优秀的类库和组件，荣获了第 15 届 Jolt 大奖。Hibernate 之所以能够流行，归功于它的以下优势。

（1）它是开放源代码的，允许开发人员在需要的时候研究源代码，改写源代码，定制客户化功能。

（2）具有详细的参考文档。

（3）对 JDBC 仅做了轻量级封装，若有必要的话，用户还可以绕过 Hibernate，直接访问 JDBC API。

（4）具有可扩展性。

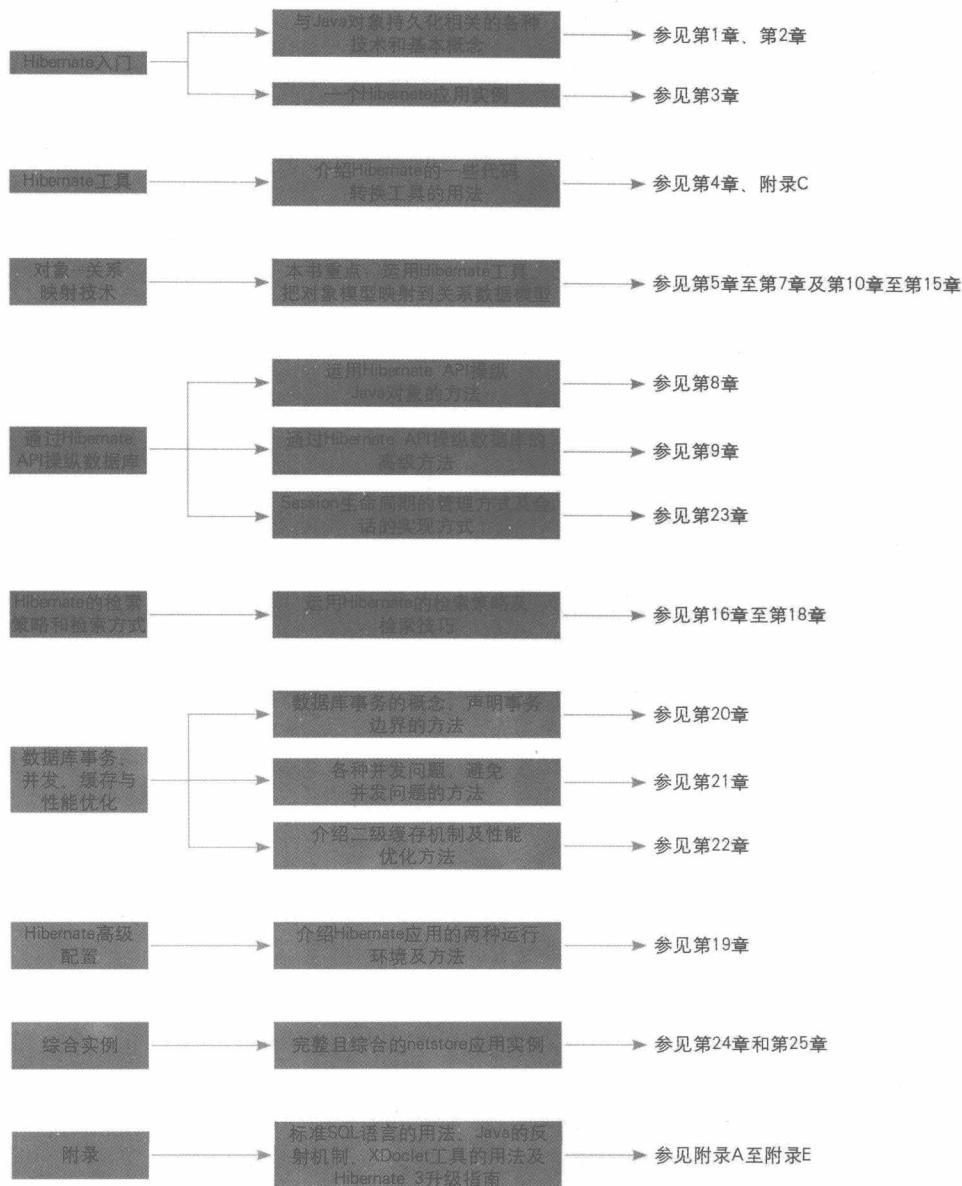
（5）使用方便，容易上手。

（6）Hibernate 既适用于独立的 Java 程序，也适用于 Java Web 应用，而且还可在 J2EE 架构中取代 CMP（Container-managed Persistence，由容器管理持久化），完成对象持久化的重任，Hibernate 能集成到会话 EJB 和基于 BMP 的实体 EJB 中，BMP（Bean-managed Persistence）是指由实体 EJB 本身管理持久化。

(7) Hibernate 可以和多种 Web 服务器、应用服务器良好集成, 并且支持几乎所有流行的数据库服务器。

本书结合大量典型的实例, 详细介绍运用目前最成熟的 Hibernate 3.3 版本进行 Java 对象持久化的技术。Hibernate 是连接 Java 对象模型和关系数据模型的桥梁, 通过本书, 读者不仅能掌握用 Hibernate 工具对这两种模型进行映射的技术, 还能获得设计与开发 Java 对象模型和关系数据模型的先进经验。

本书知识体系概览



本书的组织结构和主要内容

本书按照由浅入深、前后照应的顺序来安排内容，主要包含以下内容。

1. Hibernate 入门（第 1 章、第 2 章和第 3 章）

第 1 章、第 2 章和第 3 章为入门篇。第 1 章和第 2 章概要介绍和 Java 对象持久化相关的各种技术，详细阐述中间件、Java 对象的持久化、持久化层、数据访问细节、ORM、域模型和关系数据模型等概念。

第 3 章以一个 Hibernate 应用实例—helloapp 应用为例，引导读者把握设计、开发和部署 Hibernate 应用的整体流程，理解 Hibernate 在分层的软件结构中所处的位置。

对于已经在 Java 对象持久化领域有一定工作经验的开发人员，可以从第 1 章和第 2 章入手，高屋建瓴地把握持久化领域的各种理论，对于新手，不妨先阅读第 3 章，以便快速获得开发 Hibernate 应用的实际经验。

2. Hibernate 工具（第 4 章和附录 C）

第 4 章和附录 C 介绍 Hibernate 的一些代码转换工具的用法，例如，hbm2java 工具能根据映射文件自动生成 Java 源文件，hbm2ddl 功能能根据映射文件自动生成数据库 Schema。

3. 对象-关系映射技术（第 5、6、7、10、11、12、13、14 和 15 章）

本书重点介绍的内容之一就是如何运用 Hibernate 工具，把对象模型映射到关系数据模型，相关章节包括：

第 5 章：介绍对象-关系映射的基础知识。

第 6 章：介绍对象标识符的映射方法。

第 7 章：介绍一对多关联关系的映射方法。

第 10 章：介绍组成关系的映射方法。

第 11 章：介绍 Java 类型、SQL 类型和 Hibernate 映射类型之间的对应关系。

第 12 章：介绍继承关系的映射方法。

第 13 章：介绍 Java 集合类的用法，这一章主要是为第 14 章做铺垫的。

第 14 章：介绍 Java 集合的映射方法。

第 15 章：介绍一对一和多对多关联关系的映射方法。

4. 通过 Hibernate API 操纵数据库（第 8、9 和 23 章）

第 8 章介绍运用 Hibernate API 来保存、更新、删除、加载或查询 Java 对象的方法，并介绍 Java 对象在持久化层的 4 种状态：临时状态、持久化状态、游离状态和删除状态。深入理解 Java 对象的 4 种状态及状态转化机制，是编写健壮的 Hibernate 应用程序的必要条件。

第 9 章介绍 Hibernate 与触发器协同工作的技巧、拦截器（Interceptor）的用法，以及扩展 Hibernate 的事件监听器的方法。此外，还介绍 Hibernate 提供的批量处理数据的各种方法。

第 23 章介绍 Session 的生命周期的管理方式,以及会话的实现方式。这一章的内容将帮助读者简化 Hibernate 应用的程序代码,并且为应用设计合理的软件架构。

5. Hibernate 的检索策略和检索方式 (第 16、17 和 18 章)

第 16 章介绍 Hibernate 的各种检索策略,对每一种检索策略,都介绍它的适用场合。第 17 章和第 18 章介绍 HQL 查询语句的语法,以及 QBC API 的使用方法。合理运用 Hibernate 的检索策略及检索技巧,是提高 Hibernate 应用性能的重要手段。

6. 数据库事务、并发、缓存与性能优化 (第 20、21 和 22 章)

第 20 章先介绍数据库事务的概念,接着介绍运用 Hibernate API 和 JTA API 来声明事务边界的方法。

第 21 章介绍在并发环境中出现的各种并发问题,然后介绍采用 Hibernate 的悲观锁,以及版本控制功能来避免并发问题的方法。

第 22 章介绍 Hibernate 的二级缓存机制,并介绍如何根据实际需要来配置 Hibernate 的第二级缓存,以提高应用的性能。

7. Hibernate 高级配置 (第 19 章)

第 19 章主要介绍 Hibernate 应用的两种运行环境:受管理环境与不受管理环境,然后介绍在这两种环境中配置数据库连接池、SessionFactory 实例及事务的方法。

8. 综合实例 (第 24、25 章)

第 24 章和第 25 章介绍一个名为 netstore 应用的电子商务网站的实例,netstore 应用是利用 Struts 作为 Java Web 框架,用 Hibernate 来完成对象持久化的任务,并且分别用普通的 JavaBean 及 EJB 组件来实现业务逻辑。

9. 附录

本书的附录介绍标准 SQL 语言的主要用法、Java 的反射机制、XDoclet 工具的用法,以及把 Hibernate 2 应用升级到 Hibernate 3 应用的方法。在介绍标准 SQL 语言和 Java 反射机制时,都不是泛泛而谈,而是有针对性地介绍与 Hibernate 紧密相关的知识,如 SQL 连接查询,以及运用 Java 反射机制来实现持久化中间件的基本原理。

本书的范例程序

为了使读者不但能掌握用 Hibernate 来持久化 Java 对象的理论,并且能迅速获得开发 Hibernate 应用的实际经验,彻底掌握并会灵活运用 Hibernate 技术,本书几乎为每一章都提供完整的 Hibernate 应用范例,本书配套光盘中包含所有范例源文件。

为了方便初学者能顺利地运行本书的范例,光盘上提供的所有范例程序都是可运行的。读者只要把它们复制到本地机器上,就能够运行,不需要再做额外的配置。此外,在每个范例的根目录下还提供了 ANT 工具的工程文件 build.xml,它用于编译和运行范例程序。

本书最后还提供了一个完整的 netstore 应用例子,它实现了一个购物网站,更加贴近实际应用。本书以 netstore 应用为例,介绍软件的 MVC 框架,控制层与模型层之间通过游

离对象来传输数据的方式，以及模型层采用合理的检索策略来控制检索出来的对象图的深度，从而优化应用的性能的技巧。

这本书是否适合您

把 Java 对象持久化到关系数据库，几乎是所有企业 Java 应用必不可少的重要环节，因此本书适用于所有从事开发 Java 应用的读者。Hibernate 是 Java 应用和关系数据库之间的桥梁，阅读本书，要求读者具备 Java 语言和关系数据库的基础知识。

如果您是开发 Hibernate 应用的新手，建议按照本书的先后顺序来学习。可以先从简单的 Hibernate 应用实例下手，把握开发 Hibernate 应用的大致流程，然后逐步深入地了解对象模型映射到关系数据模型的各种细节。

如果您已经在开发 Hibernate 应用方面有着丰富的经验，则可以把本书作为实用的 Hibernate 技术参考资料。本书深入探讨把复杂的对象模型映射到关系数据模型的各种映射方案，详细介绍通过 Hibernate API 来操纵数据库的方法。灵活运用本书介绍的 Hibernate 最新技术，将使您开发 Hibernate 应用更加得心应手。

实践是掌握 Hibernate 的好方法。为了让读者彻底掌握并学会灵活运用 Hibernate，本书为每一章都提供了典型范例，本书配套光盘中提供了完整的源代码，以及软件安装程序。建议读者在学习 Hibernate 技术的过程中，善于将理论与实践相结合，以达到事半功倍的效果。

光盘使用说明

本书配套光盘包含以下目录。

1. software 目录

在该目录下包含本书内容涉及的所有软件的最新版本的安装程序，包括：

- (1) Hibernate 核心软件包 (Hibernate 3.3)。
- (2) Hibernate 工具软件包 (HibernateTools 3.2)。
- (3) MySQL 服务器的安装软件 (MySQL 5)。
- (4) MySQL 的 JDBC 驱动程序 (Mysql-Connector-Java-3.1.7)
- (5) ANT 的安装软件 (Ant 1.7)。
- (6) Tomcat 的安装软件 (Tomcat 6)。
- (7) JBoss 的安装软件 (JBoss 4)。

2. sourcecode 目录

在该目录下提供本书所有的源程序。

3. ppt 目录

在该目录下提供由本书作者亲自制作的配套语音讲座。

写作规范

为了节省文章的篇幅，在本书中显示范例的源代码时，有时做了一些省略。对于 Java 类，省略显示 `package` 语句和 `import` 语句。除了 `netstore` 应用外，本书其他范例创建的 Java 类都位于 `mypack` 包下。对于持久化类，还省略显示了属性的 `getXXX()` 和 `setXXX()` 方法。对于对象-关系映射文件，省略显示开头的 `<?xml>` 和 `<!DOCTYPE>` 元素。但在配套光盘中可获得完整的源代码。

在本书提供的 SQL 语句中，表名和字段名都采用大写形式，而 SQL 关键字，如 `select`、`from`、`insert`、`update` 和 `delete` 等，都采用小写形式。

在本书中，有时把运用了 Hibernate 技术的 Java 应用简称为 Hibernate 应用。此外，对象和实例是相同的概念；覆盖方法、重新定义方法，以及重新实现方法是相同的概念；继承和扩展是相同的概念；表的记录和表的数据行是相同的概念；表的字段和表的数据列是相同的概念；查询与检索是相同的概念；持久化类和 POJO 都是指其实例需要被持久化的基于 `JavaBean` 形式的实体域对象；对象-关系映射文件和映射文件是相同的概念；本书中的应用服务器主要指 J2EE 服务器。

本书在编写过程中得到了 Hibernate 软件组织和 SUN 公司在技术上的大力支持，飞思科技产品研发中心负责监制工作，此外本书第 1 版的读者及 `JavaThinker.org` 网站的网友为本书的编写提供了有益的帮助，在此表示衷心的感谢！尽管我们尽了最大努力，但本书难免会有不妥之处，欢迎各界专家和读者朋友批评指正。以下网址是作者为本书提供的技术支持网址，读者可通过它下载与本书相关的资源（如源代码、软件安装程序和讲义等），还可以与其他读者交流学习心得，以及对本书提出宝贵意见：

http://www.javathinker.org/hibernate_support.jsp



第 1 章 Java 应用分层架构及软件模型 1

本章介绍软件的分层结构、关系数据模型和域模型等概念。Hibernate 位于持久化层，是域模型和关系数据模型之间的桥梁。

1.1 应用程序的分层体系结构 1	1.2.2 关系数据模型 8
1.1.1 区分物理层和逻辑层 2	1.2.3 域模型 10
1.1.2 软件层的特征 3	1.2.4 域对象 10
1.1.3 软件分层的优点 4	1.2.5 域对象之间的关系 11
1.1.4 软件分层的缺点 4	1.2.6 域对象的持久化概念 17
1.1.5 Java 应用的持久化层 5	1.3 小结 19
1.2 软件的模型 6	1.4 思考题 19
1.2.1 概念模型 7	

第 2 章 Java 对象持久化技术概述 21

本章数据在内存中表现为实体域对象形式，而在关系数据库中表现为关系数据形式。数据访问代码负责把实体域对象持久化到关系数据库中。

2.1 直接通过 JDBC API 来持久化 实体域对象 21	2.3.2 JDO 模式 37
2.2 ORM 简介 27	2.3.3 CMP 模式 37
2.2.1 对象-关系映射的概念 29	2.4 Hibernate API 简介 38
2.2.2 ORM 中间件的基本 使用方法 31	2.4.1 Hibernate 的核心接口 39
2.2.3 常用的 ORM 中间件 33	2.4.2 事件处理接口 41
2.3 实体域对象的其他 持久化模式 34	2.4.3 Hibernate 映射类型接口 41
2.3.1 主动域对象模式 35	2.4.4 可供扩展的接口 42
	2.5 小结 43
	2.6 思考题 45

第 3 章 第一个 Hibernate 应用 47

本章通过简单的 helloapp 应用例子，演示如何利用 Hibernate 来持久化 Java 对象。

3.1 创建 Hibernate 的配置文件 47	3.4.2 把 Customer 持久化类映射到 CUSTOMERS 表 54
3.2 创建持久化类 48	3.5 通过 Hibernate API 操纵数据库 58
3.3 创建数据库 Schema 51	3.5.1 Hibernate 的初始化 61
3.4 创建对象-关系映射文件 52	3.5.2 访问 Hibernate 的 Session 接口 63
3.4.1 映射文件的文档类型 定义 (DTD) 52	

3.6 运行 helloapp 应用	67
3.6.1 创建运行本书范例的 系统环境	67
3.6.2 创建 helloapp 应用的 目录结构	72
3.6.3 把 helloapp 应用作为 独立应用程序运行	73

3.6.4 把 helloapp 应用作为 Java Web 应用运行	77
3.7 小结	78
3.8 思考题	80

第 4 章 hbm2java 和 hbm2ddl 工具 83

本章介绍 Hibernate 提供的两个工具 hbm2java 和 hbm2ddl，它们能简化软件开发过程。

4.1 创建对象-关系映射文件	83
4.1.1 定制持久化类	85
4.1.2 定制数据库表	88
4.2 建立项目的目录结构	90
4.3 运行 hbm2java 工具	93

4.4 运行 hbm2ddl 工具	94
4.5 使用 XML 格式的配置文件	96
4.6 小结	97
4.7 思考题	98

第 5 章 对象-关系映射基础 101

本章主要介绍单个持久化类与单个数据库表之间进行映射的方法，尤其是当持久化类的属性不和数据库表的字段一一对应时的映射技巧。

5.1 持久化类的属性及 访问方法	101
5.1.1 基本类型属性和包装 类型属性	102
5.1.2 Hibernate 访问持久化类 属性的策略	104
5.1.3 在持久化类的访问方法中 加入程序逻辑	104
5.1.4 设置派生属性	107

5.1.5 控制 insert 和 update 语句	108
5.2 处理 SQL 引用标识符	109
5.3 创建命名策略	110
5.4 设置数据库 Schema	112
5.5 设置类的包名	113
5.6 运行本章的范例程序	114
5.7 小结	120
5.8 思考题	121

第 6 章 映射对象标识符 123

本章主要介绍关系数据库中的代理主键（不具有业务含义），接着介绍 Hibernate 提供的几种内置标识符生成器的用法及适用范围。

6.1 关系数据库按主键区分 不同的记录	123
6.1.1 把主键定义为自动增长 标识符类型	123

6.1.2 从序列（Sequence）中获取 自动增长的标识符	124
6.2 Java 语言按内存地址 区分不同的对象	125

6.3	Hibernate 用对象标识符 (OID) 来区分对象	126	6.4.5	native 标识符生成器	137
6.4	Hibernate 的内置标识符生成器的用法	128	6.5	映射自然主键	138
6.4.1	increment 标识符生成器	131	6.5.1	映射单个自然主键	138
6.4.2	identity 标识符生成器	133	6.5.2	映射复合自然主键	140
6.4.3	sequence 标识符生成器	134	6.6	小结	143
6.4.4	hilo 标识符生成器	135	6.7	思考题	144
第 7 章 映射一对多关联关系		147			
本章介绍一对多关联关系的映射方法, 重点介绍 inverse 属性和 cascade 属性的用法。本章还将介绍通过 Hibernate API 来保存、修改和删除具有关联关系的对象的方法。					
7.1	建立多对一的单向关联关系	148	7.2.3	父子关系	164
7.1.1	<many-to-one>元素的 not-null 属性	153	7.3	映射一对多双向自身关联关系	165
7.1.2	级联保存和更新	155	7.4	改进持久化类	171
7.2	映射一对多双向关联关系	156	7.5	小结	175
7.2.1	<set>元素的 inverse 属性	161	7.6	思考题	176
7.2.2	级联删除	163			
第 8 章 通过 Hibernate 操纵对象 (上)		179			
本章站在持久化层的角度, Java 对象在生命周期中可处于临时状态、持久化状态、删除状态和游离状态。处于持久化状态的 Java 对象位于一个 Session 实例的缓存中, Session 能根据这个对象的属性变化来同步更新数据库。					
8.1	Java 对象在 JVM 中的生命周期	179	8.4.2	Session 的 load() 和 get() 方法	194
8.2	理解 Session 的缓存	181	8.4.3	Session 的 update() 方法	195
8.2.1	Session 的缓存的作用	182	8.4.4	Session 的 saveOrUpdate() 方法	197
8.2.2	脏检查及清理缓存的机制	184	8.4.5	Session 的 merge() 方法	198
8.3	Java 对象在 Hibernate 持久化层的状态	187	8.4.6	Session 的 delete() 方法	200
8.3.1	临时对象的特征	188	8.4.7	Session 的 replicate() 方法	201
8.3.2	持久化对象的特征	189	8.5	级联操纵对象图	202
8.3.3	被删除对象的特征	190	8.5.1	级联保存临时对象	206
8.3.4	游离对象的特征	191	8.5.2	更新持久化对象	207
8.4	Session 接口的详细用法	191	8.5.3	持久化临时对象	207
8.4.1	Session 的 save() 和 persist() 方法	191	8.5.4	更新游离对象	209
			8.5.5	遍历对象图	210

8.6 小结	211	8.7 思考题	211
第 9 章 通过 Hibernate 操纵对象 (下)		215	

本章主要内容: 当程序通过 Hibernate 来加载、保存、更新或删除对象时, 会触发 Hibernate 的拦截器及事件监听器而做出相应的处理。Hibernate 提供了 4 种批量处理数据的方式。

9.1 与触发器协同工作	215	9.4.3 通过 HQL 来进行 批量操作	230
9.2 利用拦截器 (Interceptor) 生成审计日志	217	9.4.4 直接通过 JDBC API 来 进行批量操作	232
9.3 Hibernate 的事件处理机制	224	9.5 使用元数据	233
9.4 批量处理数据	226	9.6 通过 Hibernate 调用 存储过程	234
9.4.1 通过 Session 来进行 批量操作	227	9.7 小结	234
9.4.2 通过 StatelessSession 来 进行批量操作	229	9.8 思考题	235

第 10 章 映射组成关系	237
---------------------	-----

本章主要以 Customer 和 Address 类为例, 详细介绍组成关系的映射, Hibernate 用 <component> 元素来映射 Customer 类的 homeAddress 和 comAddress 属性。Address 类作为 Hibernate 的组件, 具有 5 个特征。

10.1 建立精粒度对象模型	238	10.3.2 在应用程序中访问具有 组成关系的持久化类	244
10.2 建立粗粒度关系数据模型	239	10.4 映射复合组成关系	247
10.3 映射组成关系	240	10.5 小结	248
10.3.1 区分值 (Value) 类型和 实体 (Entity) 类型	242	10.6 思考题	249

第 11 章 Hibernate 的映射类型	251
------------------------------	-----

本章详细讲解 Hibernate 映射类型及应用, 它是 Java 类型和 SQL 类型之间的桥梁。Hibernate 映射类型分为两种: 内置映射类型和客户化映射类型。

11.1 Hibernate 的内置映射 类型	251	映射类型	254
11.1.1 Java 基本类型的 Hibernate 映射类型	251	11.2 客户化映射类型	256
11.1.2 Java 时间和日期类型的 Hibernate 映射类型	252	11.2.1 用客户化映射类型取代 Hibernate 组件	260
11.1.3 Java 大对象类型的 Hibernate 映射类型	253	11.2.2 用 UserType 映射 枚举类型	263
11.1.4 JDK 自带的个别 Java 类的 Hibernate 映射类型	253	11.2.3 实现 CompositeUserType 接口	266
11.1.5 使用 Hibernate 内置 映射类型	254	11.2.4 运行本节范例程序	271
		11.3 操纵 Blob 和 Clob 类型数据	279

11.4 小结	284	11.5 思考题	284
第 12 章 映射继承关系	287		

本章主要介绍映射继承关系的 3 种方式：继承关系树的每个具体类对应一个表、继承关系树的根类对应一个表、继承关系树的每个类对应一个表。

12.1 继承关系树的每个具体类 对应一个表	288	12.3 继承关系树的每个类 对应一个表	297
12.1.1 创建映射文件	289	12.3.1 创建映射文件	298
12.1.2 操纵持久化对象	290	12.3.2 操纵持久化对象	300
12.2 继承关系树的根类 对应一个表	293	12.4 选择继承关系的映射方式	302
12.2.1 创建映射文件	294	12.5 映射多对一多态关联	305
12.2.2 操纵持久化对象	296	12.6 小结	308
		12.7 思考题	309

第 13 章 Java 集合类	311
-----------------------	-----

本章主要介绍几种常用 Java 集合类的特性和使用方法。为了保证集合正常工作，有些集合类对存放的对象有特殊的要求，本章内容有详细说明。

13.1 Set（集）	312	13.2 List（列表）	320
13.1.1 Set 的一般用法	312	13.3 Map（映射）	321
13.1.2 HashSet 类	313	13.4 小结	325
13.1.3 TreeSet 类	315	13.5 思考题	326
13.1.4 向 Set 中加入持久化类的 对象	319		

第 14 章 映射值类型集合	327
----------------------	-----

本章介绍值类型集合的映射方法，在这种集合中存放的对象没有 OID，它们的生命周期依赖于集合所属对象的生命周期。Hibernate 采用<set>、<list>和<map>元素来映射 java.util.Set、java.util.List 和 java.util.Map。

14.1 映射 Set（集）	327	14.5.1 在数据库中对集合排序	340
14.2 映射 Bag（包）	331	14.5.2 在内存中对集合排序	341
14.3 映射 List（列表）	334	14.6 映射组件类型集合	345
14.4 映射 Map	337	14.7 小结	350
14.5 对集合排序	339	14.8 思考题	351

第 15 章 映射实体关联关系	353
-----------------------	-----

本章将介绍另外两种关联关系的映射：以 Customer 与 Address 类的关系为例，介绍映射一对一关联的各种方法；以 Category（商品类别）与 Item（商品）类，以及 Order（订单）与 Item（商品）类

的关系为例，介绍映射多对多关联的各种方法。

15.1	映射一对一关联	353
15.1.1	按照外键映射	354
15.1.2	按照主键映射	358
15.2	映射单向多对多关联	361
15.3	映射双向多对多关联关系	365
15.3.1	关联两端使用<set>元素	366
15.3.2	在 inverse 端使用	

	<bag>元素	367
15.3.3	使用组件类集合	371
15.3.4	把多对多关联分解为 两个一对多关联	376
15.4	小结	378
15.5	思考题	379

第 16 章 Hibernate 的检索策略 381

本章介绍 Hibernate 提供的 3 种检索策略，并总结这几种策略的优缺点，以及各自优先考虑使用的场合等。

16.1	Hibernate 的检索策略简介 ...	383
16.2	类级别的检索策略	386
16.2.1	立即检索	387
16.2.2	延迟检索	387
16.3	一对多和多对多关联的 检索策略	390
16.3.1	立即检索 (lazy 属性为 “false”)	392
16.3.2	延迟检索 (lazy 属性为 默认值 “true”)	392
16.3.3	增强延迟检索 (lazy 属性为 “extra”)	393
16.3.4	批量延迟检索和批量立即检索 (使用 batch-size 属性)	393
16.3.5	用带子查询的 select 语句整批量 初始化 orders 集合 (fetch 属性 为 “subselect”)	396
16.3.6	迫切左外连接检索 (fetch 属性为 “join”)	397
16.4	多对一和一对一关联的	

检索策略	398
16.4.1 迫切左外连接检索 (fetch 属性为 “join”)	398
16.4.2 延迟检索 (lazy 属性为默认值 “proxy”)	400
16.4.3 无代理延迟检索 (lazy 属性为 “no-proxy”)	401
16.4.4 立即检索 (lazy 属性为 “false”)	401
16.4.5 批量延迟检索和批量立即检索 (使用 batch-size 属性)	402
16.5 控制迫切左外连接 检索的深度	405
16.6 在应用程序中显式指定迫切 左外连接检索策略	408
16.7 属性级别的检索策略.....	408
16.8 小结	409
16.9 思考题.....	411

第 17 章 Hibernate 的检索方式 (上) 413

本章主要介绍 HQL 检索方式和 QBC 检索方式的基本用法，并对此进行归纳总结。

17.1	Hibernate 的检索 方式简介	413
------	-----------------------------	-----

17.1.1	HQL 检索方式	416
17.1.2	QBC 检索方式	417