前言

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），这是一项繁琐耗时的工作。

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。

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、Hibernate 和 MyBatis。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。

为了统一各式各样的持久化中间件，Oracle 公司为持久化层制定了统一的 JPA（Java Persistence API）。包括 Hibernate 在内的一些持久化中间件实现了 JPA。应用程序通过 JPA 来访问数据库，可以提高应用程序的独立性和灵活性。应用程序可以灵活地改变所使用的持久化中间件，而这种改变不会对程序代码带来很大影响，通常只需要改动一些配置文件即可。

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，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，而且还可以在 JavaEE 架构中取代 CMP（Container-managered 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- managered 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。

本书结合大量典性的实例，详细介绍了运用 JPA 以及目前最成熟的 Hibernate5 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 JPA 和 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．JPA 和 Hibernate 入门（第 1 、2、3 和 4 章）

第 1 至 4 章为入门篇。第 1 章和第 2 章概要介绍了和 Java 对象持久化相关的各种技术，详细阐述了中间件、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 章以一个应用实例——helloapp 应用为例，引导读者把握设计、开发和部署基于 Hibernate 的应用程序的整体流程，理解 Hibernate 在分层的软件结构中所处的位置。

第 4 章通过具体的范例，介绍了 JPA 与 Hibernate 进行整合的各种方式，介绍了 JPA 的基本的注解、以及类与接口的用法。

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 章和第 4 章，以便快速获得开发基于 Hibernate 和 JPA 的应用程序的实际经验。

2．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 JPA 和 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：介绍了 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。

3．通过 JPA 和 Hibernate API 操纵对象（第 8、9 和 23 章）

第 8 章介绍了运用 JPA 和 Hibernate API 来保存、更新、删除、加载或查询 Java 对象的方法，并介绍了 Java 对象在持久化层的四种状态：临时状态、持久化状态、游离状态和删除状态。深入理解 Java 对象的四种状态及状态转化机制，是编写健壮的 JPA 应用程序的必要条件。

第 9 章介绍了 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，还介绍了 JPA 和 Hibernate 提供的批量处理数据的各种方法。

第 23 章介绍了 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 JPA 应用的程序代码，并且为应用设计合理的软件架构。

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

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

5．数据库事务、并发、缓存与性能优化（第 20、21 和 22 章）

第 20 章先介绍了数据库事务的概念，接着介绍了运用 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，然后介绍了采用悲观锁，以及版本控制功能来避免并发问题的方法。

第 22 章介绍了 Hibernate 的二级缓存机制，并介绍了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。

6．Hibernate 高级配置（第 19 章）

第 19 章主要介绍了应用程序的两种运行环境：受管理环境与不受管理环境，然后介绍了在这两种环境中配置数据库连接池、SessionFactory 实例以及事务的方法。

7．Spring、JPA 与 Hibernate 的整合（第 24 章）

第 24 章通过一个非常精简的范例程序，介绍了 Spring、JPA 和 Hibernate 的整合过程。让读者熟悉并掌握如何在 Spring 的现成框架的基础上开发应用程序，并且把管理一些公共资源（如 EntityManagerFactory、数据源、事务管理器等）的任务交給 Spring 来处理。

8．附录

本书的附录介绍了标准 SQL 语言的主要用法和 Java 的反射机制。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了与 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。本书的附录 C 还提供了本书所有思考题的答案。

本书的范例程序

为了使读者不但能掌握用 JPA 和 Hibernate 来持久化 Java 对象的理论，并且能迅速获得开发具有实用价值的软件应用的实际经验，彻底掌握并会灵活运用 JPA 和 Hibernate 技术，本书几乎为每一章都提供了完整的应用范例，在本书配套光盘中包含了所有范例源文件。

为了方便初学者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。读者只要把它们拷贝到本地机器上，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。

本书以购物网站的模型为例，介绍了软件的 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，几乎是所有企业级 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。JPA 和 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。

如果您是学习 JPA 和 Hibernate 的新手，建议按照本书的先后顺序来学习。您可以先从简单的应用实例下手，把握开发基于 JPA 和 Hibernat 的应用（简称 JPA 应用）的大致流程，然后逐步深入地了解把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 JPA 应用方面有着丰富的经验，则可以把本书作为实用的 JPA 和

Hibernate 技术参考资料。本书深入探讨了把复杂的对象模型映射到关系数据模型的各种映射方案，详细介绍了 JPQL 查询语言的用法，并且介绍了优化 JPA 应用性能的各种手段，如选择恰当的检索策略和事务隔离级别，以及运用版本控制和 Hibernate 的第二级缓存等。灵活运用本书介绍的 JPA 和 Hibernate 最新技术，将使您开发 JPA 应用更加得心应手。

实践是掌握 Java 持久化技术的好方法。为了让读者彻底掌握并学会灵活运用 JPA 和 Hibernate，本书为每一章都提供了典型的范例，在本书配套光盘上提供了完整的源代码，以及软件安装程序。建议读者在学习本书介绍的各种持久化技术的过程中，善于将理论与实践相结合，达到事半功倍的效果。

此外，本书的大部分范例主要是通过 JPA API 来编写，所以本书的内容不仅适合使用 Hibernate 技术的读者，还适合使用 MyBatis 等其他对象-映射工具的读者。

光盘使用说明

本书配套光盘包含以下目录。

1．software 目录

在该目录下包含了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 安装软件（Hibernate 5）。
（2）MySQL 服务器的安装软件（MySQL 5）。
（3）MySQL 的 JDBC 驱动程序（Mysql-Connector-Java）。
（4）ANT 的安装软件（Ant 1.10）。
（5）Log4J 的安装软件（Log4J 1.2）
（6）SLF4J 的安装软件（SLF4J 1.7）。
（7）Tomcat 的安装软件（Tomcat 9）。
（8）Spring 的安装软件（Spring 5）。

2．sourcecode 目录

在该目录下提供了本书所有的源程序。

3．book_resource 目录

在该目录下提供了与本书有关的拓展知识文档以及相应的源程序。

4．lesson 目录

在该目录下提供了由本书作者亲自制作的配套视频课程。

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。本书大部分范例创建的 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。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 JPA 和 Hibernate 技术的 Java 应用简称为 JPA 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 JavaEE 服务器。

本书在编写过程中得到了 Hibernate 软件组织和 Oracle 公司在技术上的大力支持。此外本书第一版和第二版的读者以及 JavaThinker.net 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、软件安装程序和视频课程、PPT 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.net/hibernate.jsp