# 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

1.1.1 区分物理层和逻辑层 2

1.1.2 软件层的特征 3

1.1.3 软件分层的优点 4

1.1.4 软件分层的缺点 4

1.1.5 java应用的持久化层 5

1.2 软件的模型 6

1.2.1 概念模型 7

1.2.2 关系数据模型 8

1.2.3 域模型 10

1.2.4 域对象 10

1.2.5 域对象之间的关系 11

1.2.6 域对象的持久化概念 17

1.3 小结 19

1.4 思考题 19

# 第2章 java对象持久化技术概述 21

业务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

2.2 orm简介 27

2.2.1 对象-关系映射的概念29

2.2.2 orm中间件的基本使用方法 31

2.2.3 常用的orm中间件 33

2.3 实体域对象的其他持久化模式 34

2.3.1 主动域对象模式 35

2.3.2 jdo模式 37

2.3.3 cmp模式 37

2.4 hibernate api简介 38

2.4.1 hibernate的核心接口 39

2.4.2 事件处理接口 41

2.4.3 hibernate映射类型接口 41

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

3.2 创建持久化类 48

3.3 创建数据库schema 51

3.4 创建对象-关系映射文件 52

3.4.1 映射文件的文档类型定义（dtd） 52

3.4.2 把customer持久化类映射到customers表 54

3.5 通过hibernate api操纵数据库 58

3.5.1 hibernate的初始化 61

3.5.2 访问hibernate的session接口 63

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 hibernate的内置标识符生成器的用法 128

6.4.1 increment标识符生成器 131

6.4.2 identity标识符生成器 133

6.4.3 sequence标识符生成器 134

6.4.4 hilo标识符生成器 135

6.4.5 native标识符生成器 137

6.5 映射自然主键 138

6.5.1 映射单个自然主键 138

6.5.2 映射复合自然主键 140

6.6 小结 143

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

7.1.1 [many-to-one]元素的not-null属性 153

7.1.2 级联保存和更新 155

7.2 映射一对多双向关联关系 156

7.2.1 [set]元素的inverse属性 161

7.2.2 级联删除 163

7.2.3 父子关系 164

7.3 映射一对多双向自身关联关系 165

7.4 改进持久化类 171

7.5 小结 175

7.6 思考题 176

# 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

8.2 理解session的缓存 181

8.2.1 session的缓存的作用 182

8.2.2 脏检查及清理缓存的机制 184

8.3 java对象在hibernate持久化层的状态 187

8.3.1 临时对象的特征 188

8.3.2 持久化对象的特征 189

8.3.3 被删除对象的特征 190

8.3.4 游离对象的特征 191

8.4 session接口的详细用法 191

8.4.1 session的save()和persist()方法 191

8.4.2 session的load()和get()方法 194

8.4.3 session的update()方法 195

8.4.4 session的saveorupdate()方法 197

8.4.5 session的merge()方法 198

8.4.6 session的delete()方法 200

8.4.7 session的replicate()方法 201

8.5 级联操纵对象图 202

8.5.1 级联保存临时对象 206

8.5.2 更新持久化对象 207

8.5.3 持久化临时对象 207

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

9.2 利用拦截器（interceptor）生成审计日志 217

9.3 hibernate的事件处理机制 224

9.4 批量处理数据 226

9.4.1 通过session来进行批量操作 227

9.4.2 通过statelesssession来进行批量操作 229

9.4.3 通过hql来进行批量操作 230

9.4.4 直接通过jdbc api来进行批量操作 232

9.5 使用元数据 233

9.6 通过hibernate调用存储过程 234

9.7 小结 234

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

10.2 建立粗粒度关系数据模型 239

10.3 映射组成关系 240

10.3.1 区分值（value）类型和实体（entity）类型 242

10.3.2 在应用程序中访问具有组成关系的持久化类 244

10.4 映射复合组成关系 247

10.5 小结 248

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

11.1.1 java基本类型的hibernate映射类型 251

11.1.2 java时间和日期类型的hibernate映射类型 252

11.1.3 java大对象类型的hibernate映射类型 253

11.1.4 jdk自带的个别java类的hibernate映射类型 253

11.1.5 使用hibernate内置映射类型 254

11.2 客户化映射类型 256

11.2.1 用客户化映射类型取代hibernate组件 260

11.2.2 用usertype映射枚举类型 263

11.2.3 实现compositeusertype接口 266

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

12.1.1 创建映射文件 289

12.1.2 操纵持久化对象 290

12.2 继承关系树的根类对应一个表 293

12.2.1 创建映射文件 294

12.2.2 操纵持久化对象 296

12.3 继承关系树的每个类对应一个表 297

12.3.1 创建映射文件 298

12.3.2 操纵持久化对象 300

12.4 选择继承关系的映射方式 302

12.5 映射多对一多态关联 305

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

13.1.1 set的一般用法 312

13.1.2 hashset类 313

13.1.3 treeset类 315

13.1.4 向set中加入持久化类的对象 319

13.2 list（列表） 320

13.3 map（映射） 321

13.4 小结 325

13.5 思考题 326

# 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

14.2 映射bag（包） 331

14.3 映射list（列表） 334

14.4 映射map 337

14.5 对集合排序 339

14.5.1 在数据库中对集合排序 340

14.5.2 在内存中对集合排序 341

14.6 映射组件类型集合 345

14.7 小结 350

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

17.1.3 本地sql检索方式 419

17.1.4 关于本章范例程序 420

17.1.5 使用别名 421

17.1.6 多态查询 421

17.1.7 对查询结果排序 422

17.1.8 分页查询 423

17.1.9 检索单个对象（uniqueresult()方法） 424

17.1.10 按主键逐个处理查询结果（iterate()方法） 425

17.1.11 可滚动的结果集 426

17.1.12 在hql查询语句中绑定参数 428

17.1.13 设置查询附属事项 433

17.1.14 在映射文件中定义命名查询语句 436

17.1.15 在hql查询语句中调用函数 438

17.2 设定查询条件 439

17.2.1 比较运算 440

17.2.2 范围运算 441

17.2.3 字符串模式匹配 442

17.2.4 逻辑运算 443

17.2.5 集合运算 444

17.3 小结 445

17.4 思考题 446

# 第18章 hibernate的检索方式（下） 449

本章和第17章详细介绍hibernate提供的hql、qbc及本地sql检索方式的用法，总结比较hql与qbc的优缺点。

18.1 连接查询 449

18.1.1 默认情况下关联级别的运行时检索策略 450

18.1.2 迫切左外连接 451

18.1.3 左外连接 453

18.1.4 内连接 457

18.1.5 迫切内连接 461

18.1.6 隐式内连接 463

18.1.7 右外连接 464

18.1.8 使用sql风格的交叉连接和隐式内连接 466

18.1.9 关联级别运行时的检索策略 467

18.2 投影查询 468

18.3 报表查询 471

18.3.1 使用聚集函数 471

18.3.2 分组查询 472

18.3.3 优化报表查询的性能 475

18.4 高级查询技巧 475

18.4.1 动态查询 476

18.4.2 集合过滤 478

18.4.3 子查询 480

18.4.4 本地sql查询 482

18.4.5 查询结果转换器 484

18.5 查询性能优化 486

18.5.1 iterate()方法 486

18.5.2 查询缓存 487

18.6 小结 488

18.7 思考题 489

# 第19章 hibernate高级配置 491

本章介绍hibernate的高级配置选项。在任何运行环境中，都可由hibernate本身负责管理数据库连接池和jdbc事务；在受管理环境中，hibernate可以使用容器提供的数据源，使用jta事务，并且把sessionfactory发布为jndi资源。

19.1 配置数据库连接池 491

19.1.1 使用默认的数据库连接池 494

19.1.2 使用配置文件指定的数据库连接池 495

19.1.3 从容器中获得数据源 495

19.1.4 由java应用本身提供数据库连接 497

19.2 配置事务类型 497

19.3 把sessionfactory与jndi绑定 500

19.4 配置日志 502

19.5 使用xml格式的配置文件 505

19.6 小结 507

19.7 思考题 508

# 第20章 声明数据库事务 511

本章重点介绍通过hibernate api及jta api来声明事务边界的方法，本章还简要介绍通过mysql.exe客户程序，以及在程序中通过jdbc api来声明事务边界的过程。

20.1 数据库事务的概念 511

20.2 声明事务边界的方式 513

20.3 在mysql.exe程序中声明事务 515

20.4 java应用通过jdbc api声明jdbc事务 517

20.5 java应用通过hibernate api声明jdbc事务 518

20.5.1 处理异常 520

20.5.2 session与事务的关系 522

20.5.3 设定事务超时 525

20.6 java应用通过hibernate api声明jta事务 525

20.7 java应用通过jta api声明jta事务 526

20.8 小结 529

20.9 思考题 530

# 第21章 处理并发问题 533

本章介绍事务隔离级别、悲观锁和乐观锁等概念，并且介绍在应用程序中设置事务隔离级别及运用悲观锁和乐观锁的方法。

21.1 多个事务并发运行时的并发问题 533

21.1.1 第一类丢失更新 534

21.1.2 脏读 535

21.1.3 虚读 535

21.1.4 不可重复读 536

21.1.5 第二类丢失更新 536

21.2 数据库系统的锁的基本原理 537

21.2.1 锁的多粒度性及自动锁升级 538

21.2.2 锁的类型和兼容性 538

21.2.3 死锁及其防止办法 540

21.3 数据库的事务隔离级别 541

21.3.1 在mysql.exe程序中设置隔离级别 543

21.3.2 在应用程序中设置隔离级别 543

21.4 在应用程序中采用悲观锁 544

21.4.1 利用数据库系统的独占锁来实现悲观锁 544

21.4.2 由应用程序实现悲观锁 550

21.5 利用hibernate的版本控制来实现乐观锁 550

21.5.1 使用[version]元素 551

21.5.2 使用[timestamp]元素 556

21.5.3 对游离对象进行版本检查 558

21.5.4 强制更新版本 559

21.6 实现乐观锁的其他方法 559

21.7 小结 560

21.8 思考题 561

# 第22章 管理hibernate的缓存 563

本章先介绍缓存的基本原理，然后介绍hibernate的二级缓存结构，接下来介绍第一级缓存和第二级缓存的管理和配置，重点介绍第二级缓存的配置方法。

22.1 缓存的基本原理 563

22.1.1 持久化层的缓存的范围 564

22.1.2 持久化层的缓存的并发访问策略 566

22.2 hibernate的二级缓存结构 568

22.3 管理hibernate的第一级缓存 569

22.4 管理hibernate的第二级缓存 570

22.4.1 配置进程范围内的第二级缓存 571

22.4.2 配置集群范围内的第二级缓存 575

22.4.3 在应用程序中管理第二级缓存 578

22.4.4 session与第二级缓存的交互模式 578

22.5 运行本章的范例程序 579

22.6 小结 583

22.7 思考题 584

# 第23章 管理session和实现对话 587

本章提供实现对话的几种方式，并且分析每一种实现方式的运行性能。这些实现方式的主要区别在于：session对象的生命周期、事务的生命周期及清理缓存的方式不一样。

23.1 管理session对象的生命周期 587

23.1.1 session对象的生命周期与本地线程绑定 590

23.1.2 session对象的生命周期与jta事务绑定 592

23.2 实现对话 593

23.2.1 使用游离对象 595

23.2.2 使用手工清理缓存模式下的session 597

23.3 小结 603

23.4 思考题 605

# 第24章 hibernate与struts框架 607

本章以一个名为netstore的购物网站应用为例，介绍模型层、持久化层与数据层的设计与开发，介绍把hibernate集成到struts框架中的方法。

24.1 实现业务数据 609

24.2 实现业务逻辑 612

24.3 netstore应用的订单业务 622

24.4 小结 626

# 第25章 hibernate与ejb组件 629

本章将创建一个基于j2ee 1.4的netstore应用，它包含一个无状态会话ejb组件，名为netstoreejb，这个ejb组件实现了netstore应用的业务逻辑。

25.1 创建ejb组件 629

25.1.1 编写remote接口 629

25.1.2 编写home接口 631

25.1.3 编写enterprise java bean类 631

25.2 在业务代理类中访问ejb组件 633

25.3 发布j2ee应用 638

25.3.1 在jboss上部署ejb组件 638

25.3.2 在jboss上部署web应用 639

25.3.3 在jboss上部署j2ee应用 640

25.4 小结 642

# 附录

## 附录a 标准sql语言的用法 643

本附录先介绍数据完整性的概念，接下来从sql运用的角度，以customers表和orders表为例，介绍ddl、dml和dql语言的用法。

a.1 数据完整性 644

a.1.1 实体完整性 644

a.1.2 域完整性 644

a.1.3 参照完整性 644

a.2 ddl数据定义语言 644

a.3 dml数据操纵语言 646

a.4 dql数据查询语言 647

a.4.1 简单查询 648

a.4.2 连接查询 648

a.4.3 子查询 651

a.4.4 联合查询 652

a.4.5 报表查询 652

## 附录b java语言的反射机制 655

hibernate在运行时判断持久化类的对象的实际类型，获得它的各种getxxx()和setxxx()方法，这应该归功于java语言的反射机制。还将介绍如何运用反射机制来持久化任意一个持久化类的实例。

b.1 java reflection api简介 655

b.2 运用反射机制来持久化java对象 658

## 附录c 用xdoclet工具生成映射文件 665

xdoclet是一种通过读取java源文件中的特定标记，然后生成指定格式的文件的工具。本附录介绍如何用xdoclet工具生成hibernate映射文件。

c.1 创建带有@hibernate标记的

java源文件 665

c.2 建立项目的目录结构 669

c.3 运行xdoclet工具 672

## 附录d 发布和运行netstore应用 675

本附录介绍如何发布和运行netstore应用。netstore应用的模型有两种实现方式，与之对应，netstore应用有两种工作模式。

d.1 运行netstore所需的软件 675

d.2 netstore应用的目录结构 676

d.3 安装sampledb数据库 677

d.4 安装和配置jboss服务器 677

d.5 发布netstore应用 679

d.5.1 在工作模式1下发布netstore应用 679

d.5.2 在工作模式2下发布netstore应用 679

d.6 运行netstore应用 680

## 附录e hibernate 3升级指南 685

尽管hibernate 3 与hibernate 2的源代码是不兼容的，但对于现有的基于hibernate 2的java项目，可以很方便地把它升级到hibernate 3。本附录还描述hibernate 3版本的新变化及其变化包括的3个方面。

e.1 hibernate api 变化 685

e.1.1 包名 685

e.1.2 org.hibernate.classic包 685

e.1.3 hibernate所依赖的第三方软件包 686

e.1.4 异常模型 686

e.1.5 session接口 686

e.1.6 createsqlquery() 687

e.1.7 lifecycle和validatable接口 687

e.1.8 interceptor接口 687

e.1.9 usertype和compositeusertype接口 687

e.1.10 fetchmode类 688

e.1.11 persistentenum类 688

e.1.12 对blob 和clob的支持 688

e.1.13 hibernate中供扩展的api的变化 688

e.2 元数据的变化 688

e.2.1 检索策略 688

e.2.2 对象标识符的映射 688

e.2.3 集合映射 689

e.2.4 dtd 689

e.3 查询语句的变化 689

e.3.1 indices()和elements()函数 689

e.4 把hibernate 2应用升级到hibernate 3应用 690

## 附录f 思考题答案 691

参考文献 693