



普通高等教育“十一五”国家级规划教材
高等职业院校计算机教育规划教材
Gaodeng Zhiye Yuanxiao Jisuanji Jiaoyu Guihua Jiaocai

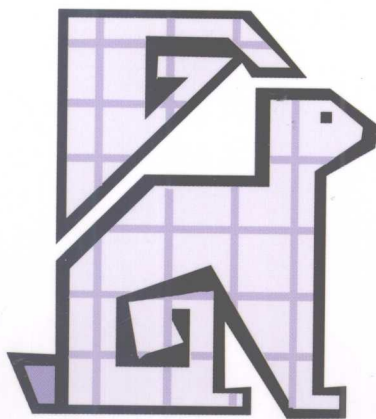
Java网络编程基础

(第2版)

Java WANGLUO BIANCHENG JICHU

朱喜福 戴舒樽 王晓勇 编

- 强调网络编程理论基础
- 突出网络编程技术实践
- 综合实例的针对性较强



人民邮电出版社
POSTS & TELECOM PRESS



精品系列



普通高等教育“十一五”国家级规划教材

高等职业院校计算机教育规划教材

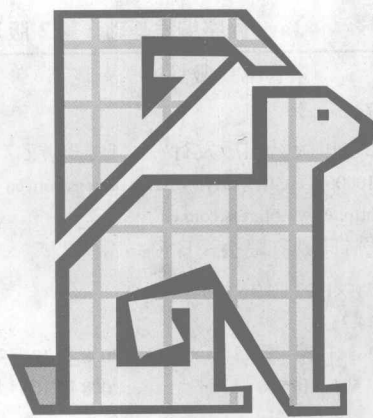
Gaodeng Zhiye Yuanxiao Jisuanji Jiaoyu Guihua Jiaocai

Java网络编程基础

(第2版)

Java WANGLUO BIANCHENG JICHU

朱喜福 戴舒樽 王晓勇 编



人民邮电出版社

北京



精品系列

图书在版编目 (CIP) 数据

Java 网络编程基础 (第 2 版) / 朱喜福, 戴舒樽, 王晓勇编. —2 版.—北京: 人民邮电出版社, 2008.10
高等职业院校计算机教育规划教材
ISBN 978-7-115-18503-7

I. J… II. ①朱…②戴…③王… III. JAVA 语言—程序设计—高等学校: 技术学校—教材 IV.TP312

中国版本图书馆 CIP 数据核字 (2008) 第 102978 号

内 容 提 要

本书是为 Java 学习者在了解 Java 编程的基础知识之后, 进一步学习 Java 网络编程而编写的教材。全书共 4 章, 第 1 章通过使用 MySQL 讲述 JDBC 数据库应用编程, 内容紧密围绕“综合实例——Java 考试管理系统”来展开; 第 2 章讲述 C/S 网络编程及应用, 通过“综合实例——C/S 考试系统”着重讲述 C/S 模式下的 Java 网络应用, 这个实例与第 1 章的“Java 考试管理系统”相结合, 实现了一个完整的实用项目; 第 3 章讲述 JSP 及 JavaBean 网站编程, 并通过“综合实例——电子商务网站框架”讲述基于 Web 的电子商务网站的设计与实现; 第 4 章讲述如何应用 J2ME 开发简单的手机应用程序, 并通过一个综合实例完整地讲述了 J2ME 手机游戏设计与编码实现。

本书内容实用, 通俗易懂, 循序渐进, 实例丰富, 并注重培养学生解决实际问题的能力。本书每章都安排了大量有针对性的上机编程练习题, 便于教师教学和检验学生的学习效果, 帮助学生巩固和加强所学内容。本书可作为高等职业院校、应用型本科院校“Java 网络编程”课程的教材; 对于具有一定的 Java 编程基础, 并希望进一步熟悉 Java 网络编程、数据库编程、商务网站开发和手机编程的读者, 本书也是一本理想的参考书。

普通高等教育“十一五”国家级规划教材

高等职业院校计算机教育规划教材

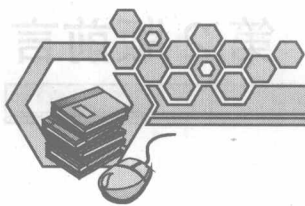
Java 网络编程基础 (第 2 版)

- ◆ 编 朱喜福 戴舒樽 王晓勇
责任编辑 李 凯
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 14.25
字数: 365 千字
印数: 7 501—10 500 册
- 2008 年 10 月第 2 版
2008 年 10 月北京第 1 次印刷

ISBN 978-7-115-18503-7/TP

定价: 24.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154



丛书出版前言

目前, 高职高专教育已经成为我国普通高等教育的重要组成部分。在高职高专教育如火如荼的发展形势下, 高职高专教材也百花齐放。根据教育部发布的《关于全面提高高等职业教育教学质量的若干意见》(简称 16 号文) 的文件精神, 本着为进一步提高高等教育的教学质量和服务的根本目的, 同时针对高职高专院校计算机教学的思路和方法的不断改革和创新, 人民邮电出版社精心策划了这套高质量、实用型的教材——“高等职业院校计算机教育规划教材”。

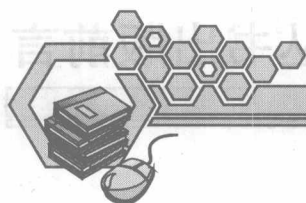
本套教材中的绝大多数品种是我社多年来高职计算机精品教材的积淀, 都经过了广泛的市场检验, 赢得了广大师生的认可。为了适应新的教学要求, 紧跟新的技术发展, 我社再一次组织了广泛深入的调研, 组织了上百名教师、专家对原有教材做认真的分析和研讨, 在此基础上重新修订出版。

本套教材中虽然还有一部分品种是首次出版, 但其原稿也经过实际教学的检验并不断完善。因此, 本套教材集中反映了高职院校近几年来的教学改革成果, 是教师们多年来教学经验的总结。本套教材中的每一部作品都特色鲜明, 集高质量与实用性为一体。

本套教材的作者都具有丰富的教学经验和写作经验, 思路清晰, 文笔流畅。教材编写充分体现高职高专教学的特点, 深入浅出, 言简意赅。理论知识以“够用”为度, 突出工作过程导向, 突出实际技能的培养。

为方便教师授课, 本套教材将提供完善的教学服务体系。教师可通过访问人民邮电出版社网站 <http://www.ptpress.com.cn/download> 下载相关资料。

欢迎广大教师对本套教材的不足之处提出批评和建议!



第 2 版前言

本书的第 1 版《Java 网络应用编程入门》自 2004 年出版以来,受到了全国许多高等职业院校师生的欢迎,并被指定为“北京市高等教育自学考试网络应用技术专业”的教材。该教材作为《Java 程序设计(第二版)》(ISBN 978-7-115-15764-5/TP,朱喜福 编)的后续教材或实训教材,为学生比较完整地学习 Java 编程提供了很大的帮助,为学生掌握 Java 网络编程、毕业后从事软件开发打下了很好的基础。

编者结合近几年的 Java 课程教学改革实践和广大读者的反馈意见,在保留原书特色的基础上,对教材进行了全面的修订。这次修订将书名更改为《Java 网络编程基础(第 2 版)》,并以项目教学为理念,精心设计了每章的综合实例,修改或替换了书中的源代码,程序设计思路清晰、注释清楚、讲解详细,方便教师教学和学生自学。编者还修改了每章的编程练习题,使之更加贴近教学。本书的具体修订内容如下。

- 对第 1 章 JDBC 数据库应用编程(原书第 2 章)的内容进行了较大的修改:大部分内容的讲述都围绕本章“综合实例——Java 考试管理系统”展开,设计了比较符合实际应用的 JDBC 数据库程序和方便易用的图形用户界面。程序设计思路充分体现面向对象的设计思想。

- 第 2 章(原书第 1 章)中的“综合实例——C/S 考试系统”结合了第 1 章“Java 考试管理系统”,使 JDBC 数据库编程和 C/S 网络编程相结合实现了一个完整的应用项目。为学生较好地掌握相关知识,并将其应用于实际项目开发、毕业设计提供了一个完整的学习实例。

- 第 3 章重点讲述“综合实例——电子商务网站框架”,编者对其内容进行了增加和完善:增加了“用户注册、登录模块及库管理”模块,还增加了“用户浏览商品及购物车”模块;对“管理员模块”进行了完善。这个框架作为一个电子商务网站的基础,已经可以基本满足用户浏览、购物和后台管理人员管理的需要。

- 第 4 章增加了综合实例:通过一个简单的手机游戏讲述手机游戏的设计和编码实现。

- 增加了 3 个附录:详细说明 Eclipse 开发 Java 项目、MyEclipse 下开发 Web 项目和 Eclipse 下开发 J2ME 手机应用程序等,便于教师在教学中配置开发环境。

修订后,教材的内容更加符合课程的培养目标,教学实施更加方便、有效,重点更加突出。在修订过程中,编者注意增强学生学习兴趣,使内容符合学生学习特点,在内容编排和难度控制上更加贴近实际教学情况。

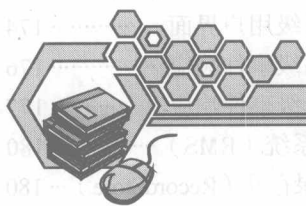
本书力求内容新颖、结构合理、概念清晰、通俗易懂并具有较强的实用性。例题的选择既考虑加深学生对知识的理解和掌握,又考虑到学生的学习兴趣 and 编程应用,每道例题都有详细的讲解和分析。每章综合实例都围绕一个中、小型项目的设计和实现展开,对所学知识进行综合、拓展,具有较强的针对性,有助于提高学生程序设计和编写代码的能力。

本教材由北京联合大学师范学院朱喜福编写,戴舒搏、王晓勇参与了第 4 章部分内容的修订,刘江委参与了第 1 章部分内容的修订。

由于编者水平有限,书中难免存在缺点和错误,恳请广大读者批评指正。

编 者

2008 年 7 月



目 录

第 1 章 JDBC 数据库应用编程.....1	第 2 章 C/S 网络编程及应用..... 53
1.1 编程环境与关系型数据库管理系统..... 1	2.1 C/S 网络编程概述.....53
1.1.1 编程环境.....1	2.1.1 InetAddress 类..... 53
1.1.2 关系型数据库管理系统.....2	2.1.2 服务器和客户机..... 54
1.2 SQL 基础.....3	2.1.3 端口号..... 55
1.2.1 SQL 数据类型.....3	2.2 Socket 通信.....55
1.2.2 数据定义言语 (DDL).....4	2.2.1 Socket 通信简介..... 55
1.2.3 数据处理语言 (DML).....6	2.2.2 Socket 类..... 56
1.3 JDBC 基础.....8	2.2.3 ServerSocket 类..... 56
1.3.1 JDBC 驱动.....8	2.2.4 Socket 通信实现简单聊天室..... 59
1.3.2 JDBC 的功能.....8	2.3 UDP 通信.....69
1.3.3 连接池.....9	2.3.1 UDP 通信简介..... 69
1.3.4 事务操作.....9	2.3.2 简单的 UDP 客户机/服务器通信..... 69
1.4 JDBC 编程实现 Admin 表的管理.....9	2.4 综合实例——C/S 考试系统.....72
1.4.1 数据库连接类 MySqlConnection..... 10	2.4.1 C/S 考试系统的客户端程序..... 74
1.4.2 数据库表的对应类 Admin..... 12	2.4.2 C/S 考试系统服务器端程序..... 80
1.4.3 数据库表的操作类 DBAdmin..... 13	2.4.3 C/S 考试系统的运行..... 86
1.4.4 JDBC 操作流程.....19	习题.....88
1.5 综合实例——Java 考试管理系统.....20	第 3 章 JSP 及 JavaBean 网站编程.....90
1.5.1 数据表.....20	3.1 编程环境与 JSP 网络的目录结构.....90
1.5.2 数据表的对应类 (examch1.mybean 包).....21	3.1.1 编程环境..... 90
1.5.3 数据库表的操作类 (examch1.db 包).....23	3.1.2 JSP 网站的目录结构..... 92
1.5.4 考试管理系统图形用户界面类 (examch1.mygui 包).....29	3.2 JSP (JavaServerPages) 技术.....93
1.5.5 考试管理系统主界面 TestFrame 类 (examch1 包).....49	3.2.1 网络请求与响应..... 93
习题.....52	3.2.2 什么是 JSP..... 94
	3.2.3 JSP 的优势..... 94
	3.2.4 Servlet 的编写及在 Web.xml 文档中配置 Servlet..... 95
	3.2.5 JSP 的执行过程..... 98
	3.3 JSP 语法对象.....99

3.3.1	页面指令	100	4.2.3	低级用户界面	174
3.3.2	include 指令	101	4.2.4	高级事件响应	176
3.3.3	声明	101	4.2.5	低级事件响应	177
3.3.4	表达式	101	4.3	记录管理系统 (RMS)	180
3.3.5	Scriptlet	103	4.3.1	记录存储 (RecordStore)	180
3.3.6	JSP 注释	104	4.3.2	记录的添、删、改、查	180
3.3.7	<jsp:useBean>操作	104	4.4	计时器 (Timer) 和计时器任务 (Timer Task)	182
3.3.8	<jsp:include>操作	106	4.4.1	计时器 (Timer)	182
3.3.9	<jsp:param>操作	107	4.4.2	计时器任务 (Timer Task)	183
3.4	JSP 隐含对象及其主要 API	108	4.5	综合实例——简单的射击游戏	185
3.4.1	request	108	4.5.1	游戏规则与玩法	185
3.4.2	response	108	4.5.2	游戏的流程与类	185
3.4.3	out	109	4.5.3	游戏类的实现	186
3.4.4	session	110	4.5.4	游戏的运行及结果	201
3.4.5	application	111	4.5.5	游戏制作的总结	203
3.5	综合实例——电子商务网站框架	111	习题		203
3.5.1	电子商务网站“shop”的总体说明	111	附录：在 Eclipse+MyEclipse 环境下创建 Java 项目和 Web 项目		204
3.5.2	用户注册、登录模块中的类	112	附录 1 在 Eclipse 环境下创建 Java 项目		204
3.5.3	商品类别表和商品表对应的数据库操作类	116	附录 1.1 Eclipse 3.2+MyEclipse 5.1 的安装		204
3.5.4	用户注册、登录模块的 JSP 页面	129	附录 1.2 在 Eclipse 3.2 环境下创建 Java 项目		205
3.5.5	用户浏览商品类别、浏览商品的 JSP 页面	134	附录 1.3 创建 JavaBean 和代码重构		211
3.5.6	用户购物车模块的 JSP 页面	138	附录 2 在 MyEclipse 环境下创建 Web 项目		213
3.5.7	管理员后台管理模块的 JSP 页面	143	附录 2.1 创建 Web 项目		213
习题		157	附录 2.2 配置 JDK 和 Tomcat		215
第 4 章	J2ME 编程及应用	159	附录 2.3 启动 Tomcat 并运行 Web 项目		218
4.1	J2ME 开发环境	159	附录 3 在 Eclipse 环境下开发 J2ME 手机应用程序		219
4.1.1	J2ME 概述	159	附录 3.1 EclipseMe 插件的获取和安装		219
4.1.2	J2ME 开发环境配置	160	附录 3.2 在 EclipseMe 环境下创建 J2ME 项目并运行程序		220
4.1.3	J2ME 应用程序编译及运行	161			
4.2	高、低级用户界面编程和事件响应	164			
4.2.1	概述	164			
4.2.2	高级用户界面	165			

第1章

JDBC 数据库应用编程

本章内容提要

为了方便读者学习 JDBC 数据库应用编程,本章首先对 SQL 的知识进行了简单介绍,并在 MySQL 数据库下演示 SQL 语句的具体使用,包括:数据库、表的建立和删除;表记录的添、删、改、查等。介绍了 JDBC 的主要特点、工作方式和功能,并通过一个具体的数据库表的 JDBC 编程演示了如何加载 JDBC 驱动程序,建立和关闭连接,JavaBean 的编写,重点讲述如何编写 Java 程序实现数据库的添、删、改、查等,通过大量演示程序讲述了如何使用 Statement、PreparedStatement、ResultSet 和 ResultSetMetaData 等接口和类,使读者对 JDBC 操作流程和编程实现有比较清晰的概念和理解。最后通过“综合实例——Java 考试管理系统”对 JDBC 编程从项目实施的角度进行了比较全面的讲述,包括数据库表的设计和对对应类的编写、数据库操作类的编程实现、表现层的编程实现。为了方便不同读者的学习需求,综合实例中的表现层使用了窗口图形界面,类的设计和实现尽可能体现和应用 Java 面向对象编程的特点和相关技术,如抽象、封装、继承和多态等。本程序实例可在任何 Java 编程平台下演示运行,为了方便地编写 Java 程序,本书附录详细介绍了如何安装、配置 Eclipse,并在 Eclipse 创建 Java 项目。

1.1 编程环境与关系型数据库管理系统

1.1.1 编程环境

JDBC 数据库应用编程是指通过 Java 语言来操作数据库。
本章使用的编程环境如下。



1. 数据库

使用 MySQL 5.0（Win32 版本）数据库讲解 JDBC 编程。读者可从相关网站下载该软件，或访问 <http://www.mysql.com/> 获得最新版本的 MySQL 和相关文档资料。

2. Java 开发平台

在 JDK 5.0+Eclipse 3.2 平台下编写、编译和运行 Java 程序。Eclipse 是开源软件，也是很好的 Java 开发平台。Eclipse 的安装和使用将在本书附录中简要说明。本章的所有程序也可以使用其他 Java 编程软件（如 JCreator）编译运行。

1.1.2 关系型数据库管理系统

随着信息化时代的到来，用户对信息存储的数量越来越大，同时对这些海量数据的检索要求也越来越高。使用数据库来管理数据可以减轻存储、检索、管理数据的负担，将这些工作委托给数据库引擎来处理，应用程序设计人员就可以将精力更加集中到应用程序的开发中，同时利用商业化的数据库又大大增强了数据维护的可靠性，改进了程序对持久化数据库操作的效率。

目前存在着各种各样的数据库系统，以满足用户多种多样的需求，比如网状数据库、树状数据库、关系型数据库、面向对象数据库、文本数据库等。在各种数据库系统中，关系型数据库管理系统（Relational Database Management System, RDMS）以其高度的灵活性成为目前发展最成功的数据库系统。

比较著名的商业关系型数据库系统有 Oracle、IBM DB2、Microsoft SQL Server 等，同时还存在为数众多的免费数据库产品，如 MySQL。

在关系型数据库中，每个数据表中都有许多的数据行（Row），称之为记录（Recorder）；每条记录都由一些长度固定的字段（Column）组成，这些字段有不同的类型（Type），比如字符串、整数、小数、日期、二进制数据等；一条一条的记录就组成了数据库中的表（Table）。图 1-1 说明了某个商品表中的记录与字段。

产品ID	产品名称	供应商	类别	单位数量	单价
1	苹果汁	佳佳乐	饮料	每箱24瓶	¥ 18.00
2	牛奶	佳佳乐	饮料	每箱24瓶	¥ 19.00
3	蕃茄酱	佳佳乐	调味品	每箱12瓶	¥ 10.00
4	盐	康富食品	调味品	每箱12瓶	¥ 22.00
5	麻油	康富食品	调味品	每箱12瓶	¥ 21.35
6	酱油	妙生	调味品	每箱12瓶	¥ 25.00
7	海鲜粉	妙生	特制品	每箱30盒	¥ 30.00
8	胡椒粉	妙生	调味品	每箱30盒	¥ 40.00
9	鸡	为全	肉/家禽	每袋500克	¥ 97.00
10	蟹	为全	海鲜	每袋500克	¥ 31.00
11	大众奶酪	日正	日用品	每袋6包	¥ 21.00
12	德国奶酪	日正	日用品	每箱12瓶	¥ 38.00
13	龙虾	德昌	海鲜	每袋500克	¥ 6.00

图 1-1 记录与字段



为了加快在数据表中的查询速度，RDBMS 采用索引（Index）的方式来提高查询的速度。索引就是预先排序好的指针，被作为索引的字段称为关键字（Key）。可以同时使用多个字段来构成关键字。大多数的表都有一个主关键字（Primary Key），也称作“主键”，这个关键字必须是唯一的。也就是说在整个表中这个字段的值都不能重复，必须各不相同。典型的主键就是商品的条形码，对每件商品而言，它必须是唯一的。

RDBMS 大多通过连接（Join）操作来进行工作。进行连接操作的时候，是将两个或者多个表组成一个更大的虚拟表。连接操作允许将多个相对独立，但有逻辑关系的数据表动态地组合到一起，从而完成查询/搜索功能，例如图 1-2 所示的商品类别表和商品表。



图 1-2 主键和关联

为了达到添加、修改、删除数据的目的，RDBMS 采用了标准的结构化查询语言（Structured Query Language，SQL）。SQL 已经是数据库操作的标准，所有商业的数据库都支持并且继承了 SQL 语言，同时部分厂商还附加了自己开发的一些数据库操作。

当查询数量巨大的数据时，比如查询某个大型商场的所有商品，系统不可能一次向查询者返回全部数据，而是只能一部分、一部分地返回结果。在 RDBMS 中使用一种称为“游标”（Cursor）的概念来指向某一条记录，然后向后移动游标。系统一次只返回游标经过的记录，这样就可以一部分、一部分地返回需要的结果集。

1.2 SQL 基础

结构化查询语言（SQL）是由 20 世纪 70 年代 IBM 的结构化英语查询语言（SEQUEL）发展而来的。与其名称不同的是，SQL 远不止是一个简单的查询语言。为了控制整个数据库的操作，SQL 又可以细分成下面的子语言集：

- 数据定义言语（DDL）；
- 数据处理语言（DML）；
- 数据查询语言（DQL）；
- 数据控制语言（DCL）。

与 Java 等大多数面向对象的语言不同，SQL 是描述性的，它不是写一个类来执行某个任务，而是执行一条（段）SQL 语句来处理数据。

1.2.1 SQL 数据类型

SQL 支持表 1-1 中列出的多种数据类型，以及它们对应的 Java 数据类型。不同的数据方式支持不同的数据类型。

表 1-1

标准的 SQL 数据类型及其对应的 Java 数据类型

SQL 类型	Java 类型	说 明
BINARY	byte[]	字节数组，用于二进制大对象
BIT	boolean	布尔值（0/1）
CHAR	String	固定长度字符串，长度为 n 的字符串类型



续表

SQL 类型	Java 类型	说 明
DATETIME	java.sql.Date	yyyy-mm-dd 和 hh:mm:ss 的日期和时间
DECIMAL	java.math.BigDecimal	任意精度的有符号小数
FLOAT	double	浮点数
INTEGER	Int	32 位整型数
LONGVARBINARY	byte[]	变长度字符串, Java 允许作为输入流
LONGVARCHAR	String	变长度字符串, Java 允许作为输入流
NCHAR	String	National Character Unicode 定长字符串
NUMERIC	java.math.BigDecimal	任意精度的有符号十进制数
NTEXT	String	大字符串变量, 用于很长的字符串
NVARCHAR	String	National Character Unicode 变长字符串
REAL	float	浮点数
SMALLINT	short	十六进制整数
TIME	java.sql.Time	时间对应类
TIMESTAMP	java.sql.Timestamp	由 java.sql.Date 和毫秒数组成
VARBINARY	byte[]	字节数组
VARCHAR	String	变长字符串

1.2.2 数据定义言语 (DDL)

数据定义言语的功能是用来完成创建和修改数据库。也就是说, 数据定义言语是与修改数据库结构有关的, 它独立于操作系统和物理存储介质, 已经被规定到 SQL-2 标准中, 但是不同的数据库厂商都在 SQL-2 标准上进行了相应的扩展。

下面通过例子演示如何在 MySQL 的命令行界面创建数据库和数据库表。通过图形用户界面管理 MySQL 数据库 (如 MyManager 软件等) 将更方便, 有兴趣的读者可参考相关书籍。

安装 MySQL 5.0 很简单, 若采用默认安装, 则 MySQL 安装在下面路径:

C:\Program Files\MySQL\MySQL Server 5.0

安装并配置好 MySQL 5.0 后, 可从 Windows 系统的【开始】→【所有程序】进入 MySQL 命令行界面。如果在配置时设置了密码 (例如: 111111), 则进入 MySQL 命令行界面时需要键入密码: 111111。

1. 创建和删除数据库

CREATE DATABASE 数据库名;

DROP DATABASE 数据库名;

例如, 下面的 SQL 语句创建一个名为 “exam” 的数据库:

CREATE DATABASE exam;

注意, 每个 SQL 语句以分号 “;” 结束。

在创建数据库 exam 后, 可在 MySQL 安装目录下的 data 子目录中看到生成了 exam 文件夹。通



过下面的语句选择“exam”数据库后，就可以在这个文件夹中创建、存放多个实现不同目的的数据表：

```
USE exam;
```

指定使用某个数据库后，执行的一系列操作就限定在指定数据库内。

图 1-3 所示是通过 MySQL 创建数据库文件 exam 的过程，并使用“show database”显示了 MySQL 下的所有数据库。其中，“mysql”和“test”两个数据库是 MySQL 默认存在的。使用“DROP”命令可删除数据库。

注意，与编写 java 程序不同，SQL 不区分大小写。

2. 创建和删除数据表

```
CREATE TABLE 表名 (
    列名1 数据类型[(size)] [constraints] [default value],
    列名2 数据类型[(size)] [constraints] [default value],
    ...
);
DROP TABLE 表名;
```

数据表用来存储相互独立的信息。各个数据表相互之间可以有一定关系，也可能没有关系。有关联关系的数据表通过称为“键”的字段相互关联到一起。为了看得清楚，上面创建数据表的 SQL 语句根据列名分行书写，每列用逗号“,”分隔，整个 SQL 语句以分号“;”结束。使用“DROP”命令可删除表。

例如，下面的 SQL 语句指定使用“exam”的数据库，并在库中创建“Admin”表，该表记录管理员姓名和密码：

```
USE exam;
CREATE TABLE Admin (
    AdminId      INTEGER      AUTO_INCREMENT,
    AdminName     VARCHAR(50)  NOT NULL,
    AdminPassWord VARCHAR(50)  NOT NULL,
    PRIMARY KEY (AdminId)
);
```

“Admin”表有 3 个字段（列）：

- AdminId：管理员编号，其类型为整数（INTEGER），该字段的数值在添加数据的时候自动增加（AUTO_INCREMENT），是表的“主键”（PRIMARY KEY）；
- AdminName：管理员姓名，其类型为大小不超过 50 个字符的变长字符串（VARCHAR(50)），该字段的值不为空（NOT NULL）；
- AdminPassWord：管理员密码。

图 1-4 所示是通过 MySQL 指定数据库“exam”并创建数据表“Admin”的过程。

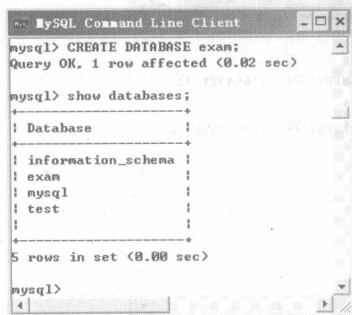


图 1-3 创建数据库“exam”

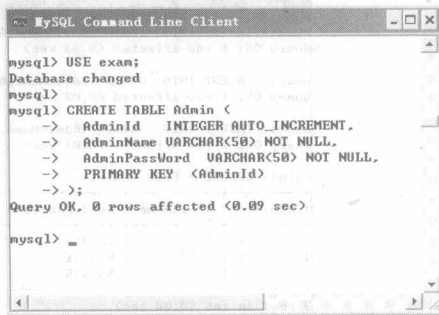


图 1-4 指定数据库并创建数据表“Admin”



1.2.3 数据处理语言（DML）

数据处理语言的功能就是完成添加数据、修改数据、删除数据。这几项功能和后面要介绍的“数据库查询语言”都是数据库应用中最常用的操作命令，而且这两部分也经常协同工作。

下面是常用的 DML 命令：

- INSERT：添加数据；
- UPDATE：修改数据；
- DELETE：删除数据。

1. 添加、查询数据

```
INSERT INTO 表名(列名1, 列名2, ...) VALUES (列值1, 列值2, ...);  
SELECT * FROM 表名;
```

在建了数据表后，使用 INSERT 命令向表中添加新的数据。使用 SELECT 命令查询表中的记录。

上面的列名和列值要一一对应，且类型要匹配。可以简单地一条一条添加记录，也可以结合后面要介绍的命令，将一个数据表中的记录批量地向另一个数据表中添加。注意，INSERT 命令用来向数据表中添加一条新记录，而不是添加某个字段。如果没有指定某个字段的值，数据库会自动添加一个默认值或者 NULL，只要这个字段在创建数据表的时候指定了默认值或者允许为 NULL。

现在使用 INSERT 语句添加数据，在本章的后半部分将使用 Java 语言，通过 JDBC 来添加、维护数据。

例如，向“Admin”表中添加 3 条记录：

```
INSERT INTO Admin (AdminName,AdminPassWord) VALUES('张三','123456');  
INSERT INTO Admin (AdminName,AdminPassWord) VALUES('李四','654321');  
INSERT INTO Admin (AdminName,AdminPassWord) VALUES('张三','555555');
```

注意，字段类型为 VARCHAR 时，其值应放在单引号或双引号中。

图 1-5 所示是通过 INSERT 语句向数据表“Admin”添加 3 条数据的过程。其中使用了 SELECT 语句查询表中的所有记录：

```
SELECT * FROM Admin;
```

```
mysql> INSERT INTO Admin (AdminName,AdminPassWord) VALUES('张三','123456');  
Query OK, 1 row affected (0.03 sec)  
  
mysql> INSERT INTO Admin (AdminName,AdminPassWord) VALUES('李四','654321');  
Query OK, 1 row affected (0.02 sec)  
  
mysql> INSERT INTO Admin (AdminName,AdminPassWord) VALUES('张三','555555');  
Query OK, 1 row affected (0.03 sec)  
  
mysql> SELECT * FROM Admin;  
+-----+-----+-----+  
| AdminId | AdminName | AdminPassWord |  
+-----+-----+-----+  
| 1 | 张三 | 123456 |  
| 2 | 李四 | 654321 |  
| 3 | 张三 | 555555 |  
+-----+-----+-----+  
3 rows in set (0.00 sec)
```

图 1-5 添加数据并查询数据



2. 修改数据

UPDATE 表名 SET 列名1=列值1, 列名2=列值2, ... WHERE 条件子句;

数据的修改是数据库操作中最常见的操作之一。使用 UPDATE 命令可以改变一条记录中某些字段的内容。UPDATE 命令通常与 WHERE 子句一起使用, 该子句的作用是对指定范围中的那些行的数据进行修改。如果在 UPDATE 命令中不使用 WHERE 子句, 那么数据表中的所有记录行都将被修改。根据修改范围的不同, WHERE 子句可以有很多种写法。

例如, 下面的 UPDATE 语句中, WHERE 子句限定 AdminId 和 AdminName 的值修改记录:

```
UPDATE Admin SET AdminName='王五', AdminPassWord='888888' WHERE AdminId=2;
UPDATE Admin SET AdminPassWord='666666' WHERE AdminName='张三';
```

图 1-6 所示是修改数据内容的过程和修改后的结果显示, 其中 AdminName 值为“张三”的两条记录的密码都修改为“666666”。数据表的 ID 字段用来唯一标记一条记录。如果只针对某一条数据进行修改, 那么在 WHERE 子句中通过指定 ID 的值是比较通用的做法。

```
mysql> UPDATE Admin SET AdminName='王五', AdminPassWord='888888' WHERE AdminId=2;
Query OK, 0 rows affected (0.01 sec)
Rows matched: 1 Changed: 0 Warnings: 0

mysql> UPDATE Admin SET AdminPassWord='666666' WHERE AdminName='张三';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 2 Changed: 0 Warnings: 0

mysql> SELECT * FROM Admin;
+----+-----+-----+
| AdminId | AdminName | AdminPassWord |
+----+-----+-----+
| 1 | 张三 | 666666 |
| 2 | 王五 | 888888 |
| 3 | 张三 | 666666 |
+----+-----+-----+
3 rows in set (0.00 sec)
```

图 1-6 根据 WHERE 子句的限定条件修改数据

3. 删除数据

DELETE FROM 表名 WHERE 条件子句

使用 DELETE 命令可以将不需要的数据记录删除。与 UPDATE 类似, DELETE 通常与 WHERE 子句一起使用, 来删除特定一条或多条记录。

例如, 下面的 DELETE 语句删除 AdminId 值为 1 的记录:

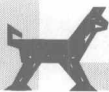
```
DELETE FROM Admin WHERE AdminId=1;
```

图 1-7 所示是删除数据后的结果。

```
mysql> DELETE FROM Admin WHERE AdminId=1;
Query OK, 1 row affected (0.06 sec)

mysql> SELECT * FROM Admin;
+----+-----+-----+
| AdminId | AdminName | AdminPassWord |
+----+-----+-----+
| 2 | 王五 | 888888 |
| 3 | 张三 | 666666 |
+----+-----+-----+
2 rows in set (0.00 sec)
```

图 1-7 根据 WHERE 子句的限定条件删除数据



1.3 JDBC 基础

1.3.1 JDBC 驱动

JDBC 是 Java 数据库连接 (Java Database Connectivity) 的缩写, 它允许用户从 Java 应用程序中访问任何表格化数据源。JDBC 除了提供更广范围的 SQL 数据库的连接外, 也允许用户访问其他表格数据源, 例如电子表格。

JDBC 定义了一个底层的 API, 用来支持独立于任何特定 SQL 实现的基本 SQL 功能。这意味着重要的是执行 SQL 语句, 然后检索它们的结果。JDBC 是以 X/Open SQL Call Level Interface 为基础的。X/Open SQL Call Level Interface 是设计用于访问 SQL 数据库的国际标准, 也是微软 ODBC 接口的基础。

JDBC 的主要特点是与任何关系型数据库协同工作的方式完全相同。JDBC 在许多不同的数据库连接模块的顶端提供了统一的接口。换句话说, 没有必要分别写一个程序访问 Oracle 数据库, 写一个程序访问 Sybase 数据库, 再写一个程序访问 SQL Server 数据库。这样就不用为连接不同的数据库而烦恼了。

要与数据源连接, 就需要所连数据源的驱动程序。

JDBC 驱动程序有 4 种基本类型。

(1) 类型 1: JDBC-ODBC 桥加上 ODBC 驱动程序。JDBC-ODBC 桥产品经过 ODBC (开放数据库连接) 驱动程序提供 JDBC 访问。ODBC 早于 JDBC 出现, 被广泛地应用于 Windows 平台的数据连接当中。

JDBC-ODBC 桥的缺点: ODBC 驱动程序也必须加载到目标计算机上; JDBC 和 ODBC 之间的转换影响效率。

(2) 类型 2: 本地 API 部分 Java 驱动程序。驱动程序使用本地 API 与数据源系统通信。使用 Java 方法调用执行数据操作的 API 函数。

类型 2 驱动程序比类型 1 驱动快很多。但是它必须在目标计算机上存在本地代码, 而且所依赖的 Java 本地接口在不同厂商提供的驱动中不一致。

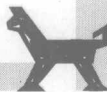
(3) 类型 3: JDBC-Net 纯 Java 驱动程序。类型 3 驱动程序将 JDBC 调用转化为 DBMS 独立网络协议, 然后由服务器转化为 DBMS 协议。这类驱动使用比较复杂。

(4) 类型 4: 本地协议的纯 Java 驱动程序。类型 4 驱动程序是一个本地协议, 是纯粹的 Java 驱动程序。允许从 Java 客户端直接调用到数据库服务器。它全部由 Java 写成, 因此不需要对客户端进行配置, 只需要注册相应的驱动程序名称即可, 也可在 Java 程序中建立连接之前显式地加载 JDBC 驱动程序。JDBC 驱动程序可从 <http://www.mysql.com/> 下载。

1.3.2 JDBC 的功能

JDBC 主要完成以下 3 种功能。

(1) 与数据源建立连接。通过 `java.sql.DriverManager` 类的静态方法 `getConnection()` 建立与数



据源的连接，这个连接将作为一个数据操作的起点，同时也是连接会话事务操作的基础。

(2) 向数据库发送 SQL 命令。通过 `java.sql.Statement` 或者 `java.sql.PreparedStatement` 对象向数据源发送 SQL 命令。在发送了 SQL 命令后，调用类中相应的 `execute` 方法来执行 SQL 命令。

(3) 处理数据源返回的结果。数据库处理了提交的 SQL 命令后，将返回处理结果。对于 DDL 和 DML 操作将返回被修改的记录数量，通过这个数量可以知道对多少条数据进行了操作；对于数据查询等操作将返回 `java.sql.ResultSet` 结果集，通过遍历 `ResultSet` 结果集获得所需要的查询结果。

1.3.3 连接池

进行 JDBC 操作的第 1 步就是建立数据连接。但是建立数据连接的过程是比较消耗资源的。这就好比每次驾车出门都要首先发动汽车，这个启动的过程会消耗掉很多资源。如果每进行一次数据库操作，都要先做一次连接，那么消耗在建立连接上的资源将大大影响程序的效率。

为了解决这一问题，JDBC 使用“连接池”概念。

在“连接池”中，保存了若干个已经建立好的数据连接，每次需要与数据源通信的时候，如果所需的连接在“连接池”里，那么直接使用这些现成的数据连接即可，从而省去了建立连接的巨大消耗，运行的速度将会大大提高。

1.3.4 事务操作

有的时候需要对多个数据表进行操作，只有对这几个表的操作都成功时才能认为整个操作成功，这样的操作称为“事务操作”。如果某一个步骤失败，之前的各种操作都要取消。这种取消的动作称为“回滚 (Rollback)”。

比如用户使用一张优惠券购买商品。这个过程中要对“优惠券表”和“订单表”进行操作。首先要操作“优惠券表”，检验优惠券的有效性。如果可以使用，就标记该优惠券为已经使用，然后再操作“订单表”，为用户下订单，同时将使用的优惠券记录到订单表中。这个过程就是一个“事务操作”。

下面来看一种失败的情况。如果完成第 1 步验证后，已经标记了优惠券被使用，然后生成订单的时候失败，这时就必须认为这个事务操作失败，就必须将之前标记为已经使用的优惠券重新改为“未使用”，也就是“回滚”之前的操作。

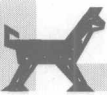
需要注意的是 JDBC 的事务操作是基于同一个数据连接的，各个连接之间互相独立，当数据连接断开后一个事务就结束了。

1.4 JDBC 编程实现 Admin 表的管理

本节将重点讲述 JDBC 编程应用，实现本章综合实例“Java 考试管理系统”中管理员表“Admin”的数据库管理，包括添、删、改、查等操作。为了实现这些操作，本程序将用到 `java.sql` 包中的相关类和接口，具体使用将在 1.4.1 节和 1.4.3 节中讲述。

本节程序的包组织结构如下：

- 数据库连接、数据表操作类在 `examch1.db` 包中。



- 数据库表对应的 JavaBean 在 examch1.mybean 包中。

如果使用 Eclipse 开发 Java 程序，可参考本书附录 1，熟悉如何新建项目、包和类等。

1.4.1 数据库连接类 MySqlConnection

本节通过实例说明如何建立 1.2.2 节中介绍的名为“exam”的数据库的连接。

为了在 Java 程序中建立数据库连接，需要做以下工作。

1. 加载数据库驱动

本章的程序例子中将使用 MySQL 的第 4 类驱动，名为“org.gjt.mm.mysql.Driver”，在 Java 程序中可以使用以下语句加载驱动：

```
Class.forName("org.gjt.mm.mysql.Driver");
```

若类路径下找不到上面方法所加载的驱动程序，则抛出 `ClassNotFoundException` 异常，应进行异常处理。该驱动程序在 `mysql-connector-java-3.0.17-ga-bin.jar` 包中，必须设置开发环境的类路径，以确保驱动程序的正确加载。

2. 建立与数据库的连接

类 `DriverManager` 管理一组 JDBC 驱动程序的基本服务，通过在注册或加载的驱动程序中进行选择，并通过类名 `DriverManager` 调用以下静态方法，建立连接（`Connection`）：

```
public static Connection getConnection(String url, String user, String password)  
throws SQLException
```

方法中各个参数的含义如下。

- `url - jdbc:subprotocol:subname` 形式的数据库 `url`。例如，本节程序例子中的 `url` 为：

```
url = "jdbc:mysql://localhost/exam";
```

- `user` - 数据库用户，连接是为该用户建立的；MySQL 的用户名为“root”。

- `password` - 用户的密码。用户配置 MySQL 时设置，如“111111”。

该方法返回到给定数据库 `URL` 的连接（`Connection`）对象。`DriverManager` 将从已注册的驱动程序集中选择一个适当的驱动程序。如果无法获得数据库连接，该方法抛出异常，返回 `NULL`；如果获得连接，则可以开始进行 JDBC 操作。

具体的连接实现参看【例 1.1】源程序中的第 16~22 行。

在实际使用中，用户名、密码、主机名称等参数要根据具体情况进行调整设定。连接不同数据库的方法会有微小差异，这个差异主要集中在 `URL` 字符串的写法上。

在 JDBC 2.0 API 中新增 `DataSource` 接口，提供了连接到数据源的另一种方法。

3. 关闭与数据库的连接

`Connection` 对应于数据库连接对象，是 JDBC 操作的起点，同时也是一个 JDBC 事务的起点。每个数据库能够同时处理的数据库连接数量是有限的，因此在不使用连接的时候要及时关闭连接。调用 `Connection` 对象的 `close()` 方法可关闭数据库连接。

在实际应用中有可能出现这样或者那样的异常情况。如果仅仅简单地关闭数据库，就有可能造成关闭失败。过多的关闭失败会造成大量的连接不再使用，但却没有关闭，仍然占用着数据库