



普通高等教育“十一五”规划教材
高职高专计算机技术专业系列教材



HIGHER TECHNICAL
AND
VOCATIONAL
EDUCATION



程序设计 实例教程

吴其庆 编著

冶金工业出版社

普通高等教育“十一五”规划教材
高职高专计算机软件技术专业系列教材

J2EE 程序设计实例教程

吴其庆 编著

北 京

冶金工业出版社

内 容 简 介

本书根据普通高等教育“十一五”国家级规划教材的指导精神,根据 Sun 公司 J2EE (Java EE 5) 的技术规格书和白皮书的标准,紧密结合 Sun 公司的网络控件开发者认证 (SCWCD)、商业控件开发者认证 (SCBCD)、网络服务开发者认证 (SCDJWS)、企业构造设计师认证 (SCEA) 的考试大纲编写本书。本书将着力于培养 21 世纪的需求导向、市场导向的实用性的 Java 计算机人才。

本书将在《J2EE 程序设计经典教程》一书的基础上,抽取实际项目的多个应用实例,将其写成通俗易懂的教学实例,使读者通过实例进一步学习 J2EE 的 3 种 EJB 的语法和应用,提高实际的编程技巧。本书的实例架构采用 C/M/S (客户端/中间层服务器/数据库服务器) 架构。讲解的知识包括 WebLogic 服务器的应用、非会话状态 EJB 的应用、会话状态 EJB 的应用、数据库缓冲池的应用、Bean 持续性全局 EJB 的应用、容器持续性全局 EJB 的应用、信息驱动 EJB 的应用、JSP 的应用、EJB 的测试、容器管理事务应用。

本书适用于 Java 语言的学习者,本科、专科、成教、职业技术培训的学生。本书目的是使读者扎实地掌握 J2EE (Java EE 5) 的各种基础知识,具备项目开发的能力。

图书在版编目 (CIP) 数据

J2EE 程序设计实例教程 / 吴其庆编著. —北京:
冶金工业出版社, 2006.7
ISBN 7-5024-4037-2

I. J... II. 吴... III. JAVA 语言—程序设计—高等
学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 063568 号

出版人 曹胜利 (北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 程志宏

广州锦昌印务有限公司印刷; 冶金工业出版社发行; 各地新华书店经销

2006 年 7 月第 1 版第 1 次印刷

787mm × 1092mm 1/16; 20 印张; 462 千字; 312 页

30.00 元

冶金工业出版社发行部 电话: (010) 64044283 传真: (010) 64027893

冶金书店 地址: 北京东四西大街 46 号 (100711) 电话: (010) 65289081

(本社图书如有印装质量问题, 本社发行部负责退换)

前 言

一、关于本书

2006 年 Sun 公司推出其企业级平台的新技术标准 Java EE 5。Java EE 5 也就是 J2EE 1.5 版本，英文全称是 Java Platform Enterprise Edition 5。如 J2SE 技术，J2EE 也不沿用 1.x 的编号，直接采用 5.0 编号，并且省略了原来的数字 2，所以称为 Java EE 5。但新版本的企业级平台的核心技术并不改变，同样包括状态 EJB、全局 EJB 和信息驱动 EJB 的应用，并且 EJB 的编写语法和步骤并不改变。应用 Java EE 5 技术标准的服务器将使 EJB 的访问效率更高和有更多的数据库连接支持。

面对不断出现的新的技术标准，读者应该如何适应？答案很简单，应用导向（Application-Oriented），市场导向（Market-Oriented）。因为无论哪一种新技术，最终都要为需求服务，都要显示友好界面给用户，都要提供方便的编码和维护环境给程序员。如果一种新技术不能达到上述要求，那将很快被淘汰。其实，Sun 公司对 Java EE 5 企业级平台的最大动作是在 2005 年 6 月成立了 GlassFish 团队，该团队的目标是设计免费、开放性代码的 Java EE 5 企业级应用的服务器。

本书将结合 J2EE 的核心知识（状态 EJB、全局 EJB 和信息驱动 EJB）和目前最为广泛应用的 WebLogic 服务器介绍各种实例的编码、部署、测试和应用；本书以 EJB 为数据和逻辑控件，结合可视化界面（Swing GUI）、互联网页面讲述各种客户端与中间层服务器的结合应用，从而使读者在不断更新的技术浪潮中成为实用性、需求导向和市场导向的技术人员。

在 2005 年至 2006 年，本人在冶金工业出版社出版了《Java 程序设计经典教程》、《JBuilder 程序设计经典教程》、《JSP 动态网站设计教程》和《J2EE 程序设计经典教程》等一系列针对 Java 基础技术丛书，受到读者的广泛欢迎。本人将在这些丛书的基础上编写一系列实例教程，作为基础技术系列丛书的补充教程。

二、本书结构

本书共 7 章，具体安排如下：

第 1 章 用户登录实例。

第 2 章 指令录入实例。

第 3 章 指令查询实例。

第 4 章 工作团队录入实例。

第 5 章 工作团队查询实例。

第 6 章 指令录入（信息驱动版）实例。

第 7 章 系统的安装程序制作与发布。

三、本书的特点

本书应用 C/M/S 结构详细剖析各种实例编写，本书的实例容易阅读、容易理解、容易

上机实习。本书的讲解风格使读者先明白是什么，怎样做，再了解为什么。本书模块实例的覆盖面广，讲解深入，以软件工程的标准与 UML 运行流程图讲解实例项目，使读者的编程理念面向需求，面向市场。

四、本书读者的问题解答

(1) 应用 WebLogic 7 是否可以部署本书的 EJB?

答：本书的 EJB 可以在 WebLogic 7 部署，但需要注意 WebLogic 7 服务器的 JDK 版本是 1.3，某些 JDK 1.4 的方法在 JDK 1.3 不能运行。

(2) 在 SJS AS 9 服务器部署 EJB 与 SJS AS 8 服务器部署 EJB 有什么区别?

答：以新技术标准设计的 SJS AS 9 服务器的 EJB 部署步骤与 SJS AS 8 服务器相同，具体步骤参考《J2EE 程序设计经典教程》。

(3) 本书应用什么版本的 JBuilder，如果应用其他版本的 JBuilder 是否可以运行本书的程序?

答：本书应用 JBuilder 2006。可以应用 JBuilder 9、X、2005、2006 运行本书的程序，只要将 J2SE 的版本设为 1.4.x 以上。

(4) 作为 J2EE 的初学者，不懂得 Java 基础语法和服务器的部署，是否适合阅读本书?

答：建议先阅读《Java 程序设计经典教程》、《Java 编程思想与实践》和《J2EE 程序设计经典教程》，因为本书不会详细讲解各个语法点和具体的操作步骤，主要以实例讲解为主。

(5) 不安装数据库，是否可以运行本书的程序?

答：本书的所有实例都和数据库结合运行和应用标准的 SQL 语句，如果不安装数据库，将不能运行本书的程序。

(6) 本书应用什么数据库，如果应用其他数据库是否可以运行本书的程序?

答：本书应用 SQL Server 2005 Express，应用其他数据也可以运行本书程序。

五、本书适用对象

本书适用于 Java 语言的学习者，本科、专科、成教、职业技术培训的学生。本书目的是使读者扎实地掌握 J2EE (Java EE 5) 的各种基础知识，具备项目开发的能力。

由于时间仓促，水平有限，书中错漏之外在所难免，恳请读者批评指正。读者如果有好的意见、建议或者在学习的过程中遇到不解的地方，可以发邮件进行探讨。

联系方式如下：

电子邮箱：qibrother1116@263.net

网址：www.cnbook.net

本书电子教案、源代码及习题参考答案可在该网站免费下载。此外，该网站还有一些其他相关书籍的介绍，可以方便读者选购参考。

编 者
2006 年 5 月

目 录

第 1 章 用户登录实例	1	1.9 实例的剖析与升华.....	51
1.1 用户登录的实例构思.....	1	1.10 小结.....	51
1.2 用户登录的 J2EE 知识	1	1.11 问与答	51
1.3 用户登录的建模分析.....	1	综合练习一	52
1.3.1 用例图	1	一、填空题.....	52
1.3.2 活动图	2	二、判断题.....	52
1.3.3 类图	2	三、问答题.....	52
1.4 用户登录的数据	2	四、上机实习题	52
1.4.1 数据字典.....	2	第 2 章 指令录入实例.....	53
1.4.2 数据表创建的 SQL 代码	3	2.1 指令录入的实例构思.....	53
1.4.3 数据添加的 SQL 代码	3	2.2 指令录入的 J2EE 知识	53
1.4.4 数据表创建的类文件.....	4	2.3 指令录入的建模分析.....	53
1.4.5 数据添加的类文件	5	2.3.1 用例图.....	53
1.4.6 数据库连接缓冲池在 WebLogic		2.3.2 活动图.....	54
服务器的设置.....	6	2.3.3 类图	54
1.5 C/M/S 结构的用户登录的文件结构	9	2.4 指令录入的数据	54
1.6 用户登录的 C/M/S 结构实例的编码 ..	10	2.4.1 数据字典.....	54
1.6.1 全局 EJB 编码	10	2.4.2 数据表创建的 SQL 代码.....	55
1.6.2 状态 EJB 编码	15	2.4.3 数据添加的 SQL 代码	55
1.6.3 数据类的编码	17	2.4.4 数据表创建的类文件.....	56
1.6.4 窗口实例编码	19	2.4.5 数据添加的类文件.....	57
1.6.5 JSP 实例编码	21	2.4.6 数据库连接缓冲池在 WebLogic	
1.6.6 EJB 应用程序的创建	23	服务器的设置	58
1.6.7 EJB 应用程序在 WebLogic		2.5 指令录入的文件结构	58
服务的部署.....	25	2.6 指令录入的 C/M/S 结构实例的编码 ..	59
1.7 用户登录的 C/M/S 结构实例的测试 ..	28	2.6.1 全局 EJB 编码.....	59
1.7.1 全局 EJB 的 Cactus 测试程序	28	2.6.2 状态 EJB 编码.....	64
1.7.2 状态 EJB 的 JUnit 测试程序.....	31	2.6.3 数据类的编码	67
1.7.3 数据类的 JUnit 测试程序	33	2.6.4 窗口实例编码	69
1.7.4 Servlet 的 Cactus 测试程序.....	36	2.6.5 JSP 实例编码	75
1.7.5 数据库压力测试	38	2.6.6 EJB 应用程序的创建	81
1.7.6 互联网访问压力测试.....	40	2.6.7 EJB 应用程序在 WebLogic	
1.8 用户登录的 C/M/S 结构实例的运行 ..	48	服务器的部署	81
1.8.1 窗口程序的运行说明.....	49	2.7 指令录入的 C/M/S 结构实例的测试 ..	81
1.8.2 JSP 程序的运行说明	49	2.7.1 全局 EJB 的 Cactus 测试程序	81

2.7.2 状态 EJB 的 JUnit 测试程序	83	3.7.3 数据类的 JUnit 测试程序	128
2.7.3 数据类的 JUnit 测试程序	85	3.7.4 Servlet 的 Cactus 测试程序	130
2.7.4 Servlet 的 Cactus 测试程序	86	3.7.5 互联网访问压力测试	131
2.7.5 数据库压力测试	88	3.8 指令查询的 C/M/S 结构	
2.7.6 互联网访问压力测试	90	实例的运行	133
2.8 指令录入的 C/M/S 结构实例的运行 ..	92	3.8.1 窗口程序的运行说明	133
2.8.1 窗口程序的运行说明	92	3.8.2 JSP 程序的运行说明	135
2.8.2 JSP 程序的运行说明	94	3.9 指令管理的窗口程序集成	137
2.9 实例的剖析与升华	97	3.9.1 指令管理的窗口程序	
2.10 小结	97	集成编码	137
2.11 问与答	97	3.9.2 指令管理的窗口程序	
综合练习二	97	运行说明	142
一、填空题	97	3.10 指令管理的互联网程序集成	143
二、判断题	98	3.10.1 指令管理的互联网	
三、问答题	98	程序集成编码	143
四、上机实习题	98	3.10.2 指令管理的互联网程序	
第 3 章 指令查询实例	99	运行说明	145
3.1 指令查询的实例构思	99	3.11 实例的剖析与升华	147
3.2 指令查询的 J2EE 知识	99	3.12 小结	147
3.3 指令查询的建模分析	99	3.13 问与答	147
3.3.1 用例图	99	综合练习三	148
3.3.2 活动图	100	一、填空题	148
3.3.3 类图	100	二、判断题	148
3.4 指令查询的数据	100	三、问答题	148
3.5 指令查询的文件结构	103	四、上机实习题	148
3.6 指令查询的 C/M/S 结构		第 4 章 工作团队录入实例	149
实例的编码	103	4.1 工作团队录入的实例构思	149
3.6.1 全局 EJB 编码	103	4.2 工作团队录入的 J2EE 知识	149
3.6.2 状态 EJB 编码	108	4.3 工作团队录入的建模分析	149
3.6.3 数据类的编码	111	4.3.1 用例图	149
3.6.4 窗口实例编码	113	4.3.2 活动图	150
3.6.5 JSP 实例编码	118	4.3.3 类图	150
3.6.6 EJB 应用程序的创建	122	4.4 工作团队录入的数据	151
3.6.7 EJB 应用程序在 WebLogic		4.4.1 数据字典	151
服务器的部署	122	4.4.2 数据表创建的 SQL 代码	152
3.7 指令查询的 C/M/S 结构		4.4.3 数据添加的 SQL 代码	152
实例的测试	122	4.4.4 数据表创建的类文件	153
3.7.1 全局 EJB 的 Cactus 测试程序 ..	122	4.4.5 数据添加的类文件	155
3.7.2 状态 EJB 的 JUnit 测试程序	125	4.4.6 数据库连接缓冲池在 WebLogic	

服务器的设置	156	5.4 工作团队查询的数据	222
4.5 工作团队录入的文件结构	157	5.5 工作团队查询的文件结构	224
4.6 工作团队录入的 C/M/S 结构		5.6 工作团队查询的 C/M/S 结构	
实例的编码	157	实例的编码	225
4.6.1 全局 EJB 编码	158	5.6.1 全局 EJB 编码	225
4.6.2 状态 EJB 编码	165	5.6.2 状态 EJB 编码	232
4.6.3 数据类的编码	170	5.6.3 数据类的编码	235
4.6.4 窗口实例编码	172	5.6.4 窗口实例编码	236
4.6.5 JSP 实例编码	182	5.6.5 JSP 实例编码	243
4.6.6 EJB 应用程序的创建	194	5.6.6 EJB 应用程序的创建	248
4.6.7 EJB 应用程序在 WebLogic		5.6.7 PEJB 应用程序在 WebLogic	
服务器的部署	194	服务器的部署	249
4.7 工作团队录入的 C/M/S 结构		5.7 工作团队查询的 C/M/S 结构	
实例的测试	195	实例的测试	249
4.7.1 全局 EJB 的 Cactus 测试程序	195	5.7.1 全局 EJB 的 Cactus 测试程序	249
4.7.2 状态 EJB 的 JUnit 测试程序	198	5.7.2 状态 EJB 的 JUnit 测试程序	252
4.7.3 数据类的 JUnit 测试程序	200	5.7.3 数据类的 JUnit 测试程序	254
4.7.4 Servlet 的 Cactus 测试程序	202	5.7.4 Servlet 的 Cactus 测试程序	255
4.7.5 数据库压力测试	206	5.7.5 互联网访问压力测试	257
4.7.6 互联网访问压力测试	208	5.8 工作团队查询的 C/M/S 结构	
4.8 工作团队录入的 C/M/S 结构		实例的运行	259
实例的运行	209	5.8.1 窗口程序的运行说明	259
4.8.1 窗口程序的运行说明	210	5.8.2 JSP 程序的运行说明	261
4.8.2 JSP 程序的运行说明	213	5.9 工作团队管理的窗口程序集成	263
4.9 实例的剖析与升华	217	5.9.1 工作团队管理的窗口	
4.10 小结	218	程序集成编码	263
4.11 问与答	218	5.9.2 工作团队管理的窗口	
综合练习四	218	程序运行说明	268
一、填空题	218	5.10 工作团队管理的互联网程序集成	269
二、判断题	219	5.10.1 工作团队管理的互联网	
三、问答题	219	程序集成编码	269
四、上机实习题	219	5.10.2 工作团队管理的互联网	
第 5 章 工作团队查询实例	220	程序运行说明	273
5.1 工作团队查询的实例构思	220	5.11 实例的剖析与升华	274
5.2 工作团队查询的 J2EE 知识	220	5.12 小结	274
5.3 工作团队查询的建模分析	220	5.13 问与答	274
5.3.1 用例图	221	综合练习五	274
5.3.2 活动图	221	一、填空题	274
5.3.3 类图	221	二、判断题	275

三、问答题	275	7.1 Ant 工具	293
四、上机实习题	275	7.1.1 Ant 工具的下载与安装	293
第 6 章 指令录入 (信息驱动版) 实例	276	7.1.2 工作团队管理窗口程序的 Ant 安装程序制作	293
6.1 指令录入 (信息驱动版) 的实例构思	276	7.1.3 工作团队管理窗口程序的 Ant 安装程序运行	294
6.2 指令录入 (信息驱动版) 的 J2EE 知识	276	7.1.4 工作团队管理互联网程序的 Ant 安装程序制作	295
6.3 指令录入 (信息驱动版) 的建模分析	276	7.1.5 工作团队管理互联网程序的 Ant 安装程序运行	295
6.3.1 用例图	276	7.2 可执行文件	296
6.3.2 活动图	276	7.2.1 可执行文件的制作	296
6.3.3 类图	277	7.2.2 可执行文件的运行	299
6.4 指令录入 (信息驱动版) 的数据	277	7.3 互联网程序默认文件的 设置与运行	299
6.5 C/M/S 结构的指令录入 (信息驱动版) 的文件结构	277	7.3.1 互联网程序默认文件的设置	300
6.6 指令录入 (信息驱动版) 的 C/M/S 结构实例的编码	278	7.3.2 互联网程序默认文件的运行	302
6.6.1 主题信息域在 WebLogic 服务器的部署	278	7.4 InstallAnywhere 工具	302
6.6.2 信息驱动 EJB 编码	280	7.4.1 InstallAnywhere 工具的 下载与安装	303
6.6.3 指令发送窗口实例编码	282	7.4.2 工作团队管理的 InstallAnywhere 安装程序制作	303
6.6.4 指令接收窗口实例编码	287	7.4.3 工作团队管理的 InstallAnywhere 安装程序运行	306
6.6.5 信息驱动 EJB 在 WebLogic 服务器的部署	289	7.4.4 工作团队管理的互联网 安装程序运行	308
6.7 指令录入 (信息驱动版) 的 C/M/S 结构实例的运行	290	7.4.5 工作团队管理的中文 安装程序运行	308
6.8 实例的剖析与升华	291	7.5 小结	310
6.9 小结	291	7.6 问与答	310
6.10 问与答	292	综合练习七	310
综合练习六	292	一、填空题	310
一、填空题	292	二、判断题	311
二、判断题	292	三、问答题	311
三、问答题	292	四、上机实习题	311
四、上机实习题	292	参考文献	312
第 7 章 系统的安装程序制作与发布	293		

第 1 章 用户登录实例

本章教学目标：

- (1) 了解用户登录的实例构思、J2EE 知识和建模分析。
- (2) 掌握用户登录的数据结构。
- (3) 了解用户登录的文件结构。
- (4) 掌握用户登录的 C/M/S 结构的程序、JUnit 测试程序和 Cactus 测试程序的编码。
- (5) 掌握用户登录的测试程序、C/M/S 结构程序、压力测试程序的运行。

1.1 用户登录的实例构思

用户登录实例由 2 种不同的客户端组成，分别是窗口和互联网应用程序，可以接受用户输入的名字和密码，实现的功能如下所示：

- (1) 登录实例可以有效检验用户名和密码，并且区分大小写。
- (2) 登录实例可以识别不同的用户，如管理员、普通用户。

1.2 用户登录的 J2EE 知识

用户登录包括的 J2EE 知识如下所示：

- (1) 数据库连接缓冲池在 WebLogic 服务器的部署。
- (2) 非会话状态 (Session) EJB 的应用。
- (3) 连接缓冲池的语法和应用。
- (4) 容器持续性全局 EJB 的应用。
- (5) JSP 的基础知识。
- (6) JUnit 技术的状态 EJB 和数据类测试、Cactus 技术的全局 EJB 测试。
- (7) 容器管理事务的应用。

1.3 用户登录的建模分析

本节讲解用户登录实例的用例图、活动图和类图。本节的用例图、活动图和类图的作图工具是 Rational Rose。

1.3.1 用例图

用例图表示用户与程序模块的关系，用户登录实例的所有用户都可以应用用户登录程序，如图 1-1 所示。

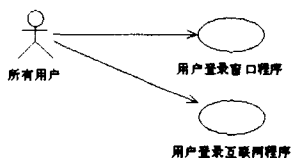


图 1-1

1.3.2 活动图

本节讲解登录程序的活动图。登录程序的运行流程是用户输入用户名、密码和选择用户类型，然后单击“登录”按钮，登录程序确认相应用户名和密码是否正确，然后显示登录信息，单击“退出”按钮结束登录程序的运行。活动图如图 1-2 所示。

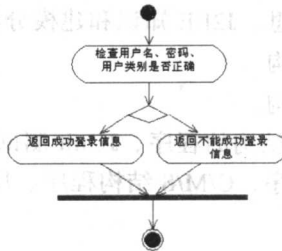


图 1-2

1.3.3 类图

本节介绍用户登录的 C/M/S 结构的类图，如图 1-3 所示。

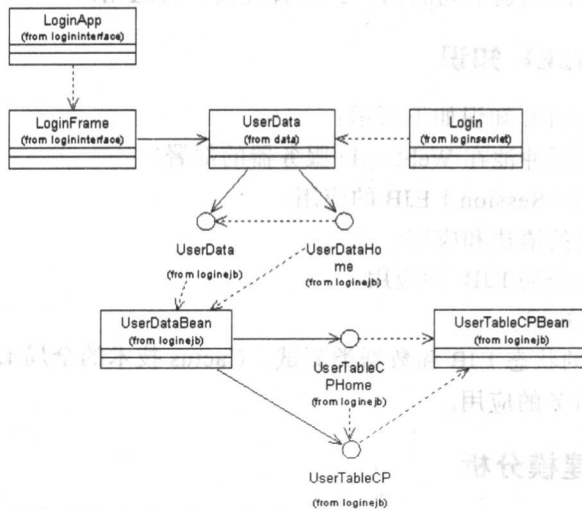


图 1-3

LoginApp 和 LoginFrame 类是窗口类，访问 UserData 类进行数据处理；Login 类是 Servlet，访问 UserData 类的方法进行数据处理；data.UserData 类是客户端数据类，通过访问状态 EJB UserData 和 UserDataHome 接口应用 UserDataBean 主类进行数据处理；UserDataBean 是状态 EJB 的主类，通过访问全局 EJB 的 UserTableCPHome 和 UserTableCP 的本地接口应用 UserTableCPBean 主类进行数据处理。

1.4 用户登录的数据

1.4.1 数据字典

本例应用的数据库是 UserC1，共有 1 个数据表，名字是 usertable，数据字典如表 1-1

所示。

表 1-1 用户数据表 (usertable)

	字段名	数据类型	大小	默认值	允许空值	惟一	主键	自动增加	外键	备注
1	userId	int				*	*	*		序号
2	userName	nvarchar	50							用户名
3	userPassword	nvarchar	50							密码
4	userState	int								用户类型, 0 表示管理员, 1 表示用户
索引	字段名	索引名					排序			
	id	PK_usertable					升序			

本实例应用 SQL Server 2005 Express 创建数据库, 读者也可以应用 SQL Server 2000 创建数据库, 具体的操作请参考《J2EE 程序设计经典教程》的 3.3 节。SQL Server 2005 Express 是微软提供的最新版数据库引擎, 数据功能与商业版相同, 可以应用于中小型系统。如果要详细了解 SQL Server 2005 Express 数据库的下载、安装和数据库创建请参考《J2EE 程序设计经典教程》的第 3 章。SQL Server 2000、2005、2005 Express 都可以成功运行本书的实例, 需要应用 Microsoft JDBC 2005 驱动程序, 保存在附带程序包的 J2EEBaseExample 目录, 文件是 sqljdbc.jar, 应用该类包可以连接 SQL Server 2000、2005 和 2005 Express 数据库。如果应用 MySQL 数据库, 需要修改驱动程序和 URL 代码, 具体的操作参考《J2EE 程序设计经典教程》的 3.3.6 节。

1.4.2 数据表创建的 SQL 代码

用户数据表创建的 SQL 代码的保存文件是 User.sql, 保存在附带程序包的 C1 目录。用户数据表创建的 SQL 代码如下所示:

```
CREATE TABLE usertable (
1:     userId int IDENTITY (1, 1) NOT NULL PRIMARY KEY,
2:     userName nvarchar (50) NOT NULL ,
       userPassword nvarchar (50) NOT NULL ,
       userState int NOT NULL
)
```

1: IDENTITY(1,1)表示每增加一行, userId 自动增加 1。如果应用 MySQL 数据库, 应用 AUTO_INCREMENT 替换 IDENTITY(1,1)。

2: 因为 MySQL 不提供 nvarchar 字段, 需要应用 varchar 替换 nvarchar。如果不替换, MySQL 4.x 和 5.x 默认替换为 varchar 字段, 低版本的 MySQL 将会显示错误信息。

1.4.3 数据添加的 SQL 代码

用户数据表的数据添加的 SQL 代码保存文件是 UserData.sql, 保存在附带程序包的 C1 目录。用户数据表的数据添加的 SQL 代码如下所示:

```
INSERT INTO usertable(userName, userPassword, userState) VALUES('manager', 'user',
0)
INSERT INTO usertable(userName, userPassword, userState) VALUES('user', 'user', 1)
```



```

        + "userPassword nvarchar (50) NOT NULL , "
        + "userState int NOT NULL "
        + ")";
        //创建数据表 usertable
        stmt.executeUpdate(sql);
        System.out.println("在数据库 UserC1 成功创建 usertable 数据表.");
    }
    catch(Exception ex){
        ex.printStackTrace();
    }
    try{
        //关闭数据库连接类
        conn.close();
    }catch(Exception ex){
        ex.printStackTrace();
    }
}
}

```

1: 如果应用 Microsoft JDBC 2000 驱动程序, 驱动程序名是 com.microsoft.jdbc.sqlserver.SQLServerDriver, Microsoft JDBC 2005 驱动程序名是 com.microsoft.sqlserver.jdbc.SQLServerDriver, 该驱动程序名可以连接 SQL Server 2000。

2: 由于本书所应用的计算机安装了多个 SQL Server 数据库引擎, 所以应用实例名字 SQL2005EXPRESS, 如果只安装一个 SQL Server, 不用设置 SQL Server 的实例名字。如果了解在同一台计算机安装多个 SQL Server 实例, 请参考《J2EE 程序设计经典教程》的第3章。

3: 要成功连接 SQL Server 2005, 需要启动 SQL Server 2005 的 TCP/IP 协议, 安装后该协议的默认状态是不启动状态, 具体操作可以参考《J2EE 程序设计经典教程》的 3.3.4 节。如果不启动 TCP/IP 协议, 将显示 Connection Refused 错误。

(3) 类文件的运行结果如图 1-5 所示。

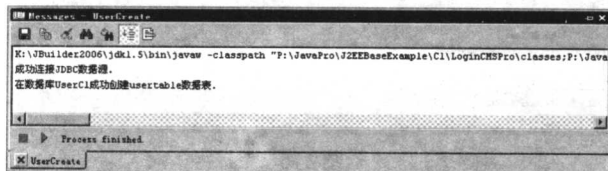


图 1-5

1.4.5 数据添加的类文件

创建数据添加的类文件, 名字是 UserDataAdd.java, 保存目录是 data, 代码如下所示:

```

package data;

import java.sql.*;

public class UserDataAdd {
    public static void main(String[] args) {
        //定义数据库连接的驱动程序
        String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
        //定义 SQL Server 数据库的连接地址, 如果是默认实例, 不用定义\\SQL2005EXPRESS
        String url = "jdbc:sqlserver://localhost\\SQL2005EXPRESS;databaseName=UserC1";
        //声明连接类
        Connection conn = null;
        try{

```

```

//使用 JDBC 技术创建数据库连接
Class.forName(driver);
//使用 DriverManager 类的建立连接, 第一个参数定义用户名, 第二个参数定义密码
conn = DriverManager.getConnection(url, "sa", "test");
if(conn != null){
    System.out.println("成功连接 JDBC 数据源.");
}
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from usertable");
if (rs.next()) {
    System.out.println("数据表 usertable 已经存在数据.");
}else {
    String sql1 = "INSERT INTO usertable(userName, userPassword, userState)"
        + " VALUES('manager', 'user', 0) ";
    String sql2 = "INSERT INTO usertable(userName, userPassword, userState)"
        + " VALUES('user', 'user', 1)";
    //为数据表 usertable 添加 2 条记录
    stmt.addBatch(sql1);
    stmt.addBatch(sql2);
    stmt.executeBatch();
    System.out.println("在数据表 usertable 成功添加 2 条记录.");
}
} catch (Exception ex) {
    ex.printStackTrace();
}
try{
    //关闭数据库连接类
    conn.close();
} catch (Exception ex) {
    ex.printStackTrace();
}
}
}

```

运行结果如图 1-6 所示。

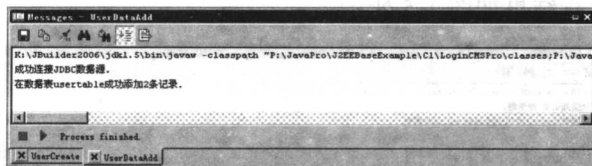


图 1-6

1.4.6 数据库连接缓冲池在 WebLogic 服务器的设置

本节应用 WebLogic 9 服务器创建数据库连接缓冲池（读者也可以应用 WebLogic 8 服务器部署数据库连接缓冲池，具体步骤参考《J2EE 程序设计经典教程》的 3.4.4 节），步骤如下所示：

（1）由于 WebLogic 9 自带的 JDBC 驱动程序（Type 4）不支持对 SQL Server 2005 的连接，所以本节应用 Microsoft 的 SQL Server 2005 JDBC 驱动程序。将 sqljdbc.jar 类包（保存在附带程序包的 J2EEBaseExample 目录）复制到 WebLogic 服务器的安装路径 WebLogic 91\server\lib。用记事本打开 WebLogic 91\samples\domains\wl_server\bin\目录的 startWebLogic.cmd 文件，修改 classpath 类路径参数，作用是在 WebLogic 9 中应用 sqljdbc.jar 类包，如下所示：

```

set CLASSPATH=M:\bea9\WebLogic 91\server\lib\sqljdbc.jar;
%CLASSPATH%;%MEDREC_WEBLOGIC_CLASSPATH%

```


(2) 启动 WebLogic 9 服务器，修改后的类路径如图 1-7 所示。

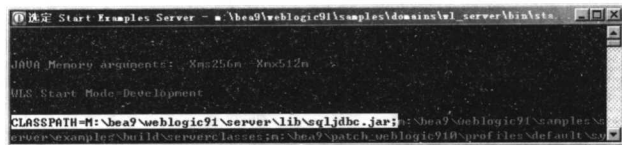


图 1-7

(3) 进入 WebLogic 9 服务器的管理窗口。单击左上角的 Lock & Edit 按钮进入编辑模式，在左边的结构树选择 Services→JDBC→Data Sources，在右边的内容框单击 New 按钮进入缓冲池和数据源创建步骤，第 1 步定义数据源和 JNDI 的名字是 UserC1Pools，定义数据库类型 (Database Type) 是 MS SQL Server，定义数据库驱动程序 (Database Driver) 为 Microsoft's MS SQL Server Driver (Type 4) Versions:2005，如图 1-8 所示。

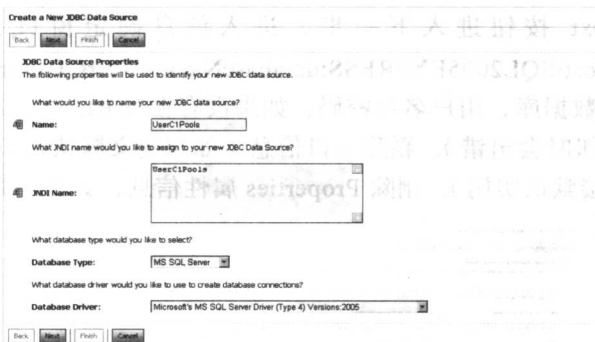


图 1-8

(4) 单击 Next 按钮进入下一步，选择事务处理复选框 (Supports Global Transactions) 和 Emulate Two-Phase Commit (事务的双阶段提交协议) 单选框，如图 1-9 所示。

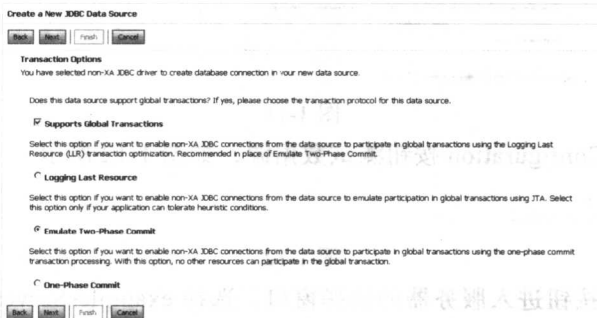


图 1-9

事务的双阶段提交协议是分布式程序常用的协议，它与单阶段提交协议的区别参考《J2EE 程序设计经典教程》的 9.1 节。

(5) 单击 Next 按钮进入下一步，数据库名字定义为 UserC1，服务器名定义为 localhost\SQL2005EXPRESS(注意：使用“\”符号表示实例，\SQL2005EXPRESS 表示应用 SQL Server 2005 Express 数据库的实例，如果只安装一个版本的 SQL Server，直接填写 localhost 便可)，端口定义为 1433，用户名定义为 sa，密码定义为 test (具体的密码根据本地计算机的进

行定义), 如图 1-10 所示。

Figure 1-10 shows the 'Create a New JDBC Data Source' dialog box, Step 1: Connection Properties. The dialog asks for the following information:

- What is the name of database you would like to connect to? Database Name:
- What is the name or IP address of the database server? Host Name:
- What is the port on the database server used to connect to the database? Port:
- What database account user name do you want to use to create database connections? Database User Name:
- What is the database account password to use to create database connections? Password:
- Confirm Password:

图 1-10

(6) 单击 Next 按钮进入下一步, 进入信息确定窗口, 将 URL 修改为 `jdbc:sqlserver://localhost\SQL2005EXPRESS;databaseName=UserC1;user=sa;password=test` (在 URL 地址应用数据库、用户名和密码, 如果没有在 URL 写入用户名和密码, 单击 Activate Changes 按钮时会出错), 删除端口信息 (如果定义默认端口为 1433, Microsoft JDBC 驱动程序会连接默认实例), 删除 Properties 属性信息, 如图 1-11 所示。

Figure 1-11 shows the 'Create a New JDBC Data Source' dialog box, Step 2: Test Database Connection. The dialog asks for the following information:

- What is the full package name of JDBC driver class used to create database connections in the connection pool? (State that the driver class must be in the classpath of any server to which it is deployed.) Driver Class Name:
- What is the URL of the database to connect to? The format of the URL varies by JDBC driver. URL:
- What database account user name do you want to use to create database connections? Database User Name:
- What is the database account password to use to create database connections? Password:
- What are the properties to pass to the JDBC driver when creating database connections? Properties:
- What table name or SQL statement would you like to use to test database connections? Test Table Name:

图 1-11

(7) 单击 Test Configuration 按钮测试数据源, 如图 1-12 所示。

Figure 1-12 shows the Messages window with the message: **Connection test succeeded.**

图 1-12

(8) 单击 Next 按钮进入服务器的选择窗口, 选择 examplesServer 选项, 如图 1-13 所示。

Figure 1-13 shows the 'Create a New JDBC Data Source' dialog box, Step 3: Select Targets. The dialog asks for the following information:

- Select Targets: You can select one or more targets to deploy your new JDBC data source. If you don't select a target, the data source will be created but not deployed. You will need to deploy the data source at a later time.
- Servers: ☒ examplesServer

图 1-13