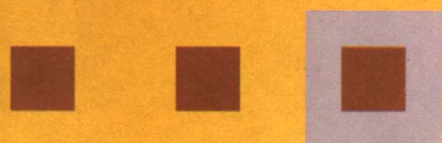




普通高等教育“十一五”规划教材

高职高专计算机软件技术专业系列教材



HIGHER TECHNICAL
AND
VOCATIONAL
EDUCATION



动态网站设计 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

JSP 动态网站设计实例教程

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 (JSP 部分),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《JSP 动态网站设计教程》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,使读者能够通过实例进一步学习 JSP 的各种标识、各种内建对象和 Servlet 的应用,提高实际的编程技巧。本书的实例架构采用 B/S (浏览端/服务器) 架构,分离请求控制、数据处理、页面显示代码。讲解的知识包括 Tomcat 服务器的应用、3 种指示 (directive) 标识的应用、4 种脚本 (scriptlet) 标识的应用、6 种动作 (action) 标识的应用、8 种 HTML 标识和 JavaScript 的应用、9 种内建对象的应用、JavaBean 的应用、Servlet 的应用、数据库各种操作的应用、文件的上传和下载应用。

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 JSP 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

JSP 动态网站设计实例教程 / 吴其庆编著. —北京:
冶金工业出版社, 2006.7
ISBN 7-5024-4039-9

I. J... II. 吴... III. JAVA 语言—主页制作—程序
设计—高等学校—教材 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2006) 第 063566 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

本书是根据普通高等教育“十一五”国家级规划教材的指导精神而编写的。

目前,全国各地高职高专院校普遍扩招,高职高专学生人数迅速增长,这给他们的就业带来了巨大的压力。而当前高职高专学生的就业情况不容乐观,究其原因,所用教材与实际应用脱轨是一主要因素。针对现有教材质量较差、品种单一、版本陈旧、实用性和可操作性不强等原因,肩负着应用型人才培养的高职高专院校急需一系列符合当前教学改革需要的教材。

2006年JSP的最新技术版本是2.1,Servlet的最新技术版本是2.5。相比于JSP和Servlet的1.0版本,新版本的JSP和Servlet引入更多的标识库,如JSTL和EL。但JSP和Servlet的核心应用并不改变,如JSP的3种指示(directive)标识、4种脚本(scriptlet)标识、6种动作(action)标识和9种内建对象;JSP和Servlet的核心应用语言同样是Java。

面对不断出现新的技术标准,读者应该如何去适应?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。所以本书将结合JSP、JavaBean和Servlet最核心的技术介绍各种实例的编码、测试和应用。

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、本书的结构

本书共10章,具体安排如下:

第1章:用户管理实例。

第2章:用户登录实例。

第3章:飞机票录入实例。

第4章:飞机票查询实例。

第5章:飞机票定价管理实例。

第6章:员工录入实例。

第7章:员工查询实例。

第8章:员工工资管理实例。

第9章:用户管理(多媒体版)实例。

第10章:安装程序制作与发布。

三、本书的特点

本书应用 B/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) 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。

(2) 作为 JSP 的初学者, 不懂得 Java 基础语法, 是否适合阅读本书?

答: 建议先阅读《JSP 动态网站设计教程》, 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。

(3) 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。

(4) 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, 应用其他数据库也可以运行本书程序。

(5) 如果要详细了解 Java 的基础知识, 应该参考哪些书籍?

答: 《Java 程序设计经典教程》、《JBuilder 程序设计经典教程》和《Java 编程思想与实践》。

五、本书适用对象

本书系统、全面地研究和借鉴了国外相关教材先进的教学方法, 结合国内院校教学实际和先进的教学成果, 根据教育部“十一五”国家级规划教材应用型高职高专教育的指导思想编写, 具有实用性和可操作性, 与时俱进, 与当前就业市场结合得更加紧密。

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 JSP 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, 可以来信进行探讨。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 用户管理实例	1	2.3.1 用例图	46
1.1 用户管理的实例构思	1	2.3.2 活动图	47
1.2 用户管理的 JSP 知识	1	2.3.3 类图	47
1.3 用户管理的建模分析	1	2.4 用户登录的数据	47
1.3.1 用例图	1	2.5 B/S 结构的用户登录的文件结构	47
1.3.2 活动图	2	2.6 用户登录的 B/S 结构实例	48
1.3.3 类图	2	2.6.1 实例编码	48
1.4 用户管理的数据	2	2.6.2 与 JavaScript 结合的实例编码	51
1.4.1 数据字典	2	2.6.3 JUnit 测试程序编码和运行	53
1.4.2 数据表创建的 SQL 代码	3	2.6.4 Cactus 测试程序编码和运行	53
1.4.3 数据添加的 SQL 代码	3	2.6.5 程序运行说明	54
1.4.4 数据表创建的 JSP 文件	3	2.6.6 与 JavaScript 结合的程序运行说明	56
1.4.5 数据添加的 JSP 文件	6	2.6.7 互联网访问压力测试	56
1.5 B/S 结构的用户管理的文件结构	7	2.7 实例的剖析与升华	59
1.6 用户管理的 B/S 结构实例	7	2.8 小结	59
1.6.1 实例编码	7	2.9 问与答	59
1.6.2 与 JavaScript 结合的实例编码	15	综合练习二	59
1.6.3 JUnit 测试程序编码和运行	19	一、填空题	59
1.6.4 Cactus 测试程序编码和运行	22	二、判断题	59
1.6.5 程序运行说明	27	三、问答题	60
1.6.6 与 JavaScript 结合的程序运行说明	29	四、上机实习题	60
1.6.7 数据库访问压力测试	30	第3章 飞机票录入实例	61
1.6.8 互联网访问压力测试	32	3.1 飞机票录入的实例构思	61
1.7 实例的剖析与升华	43	3.2 飞机票录入的 JSP 知识	61
1.8 小结	44	3.3 机票录入的建模分析	61
1.9 问与答	44	3.3.1 用例图	61
综合练习一	44	3.3.2 活动图	62
一、填空题	44	3.3.3 类图	62
二、判断题	45	3.4 飞机票录入的数据	62
三、问答题	45	3.4.1 数据字典	62
四、上机实习题	45	3.4.2 数据表创建的 SQL 代码	63
第2章 用户登录实例	46	3.4.3 数据添加的 SQL 代码	63
2.1 用户登录的实例构思	46	3.4.4 数据表创建的 JSP 文件	63
2.2 用户登录的 JSP 知识	46	3.4.5 数据添加的 JSP 文件	65
2.3 用户登录的建模分析	46	3.5 飞机票录入的文件结构	66

3.6 飞机票录入的 B/S 结构实例	66	综合练习四	115
3.6.1 实例编码	66	一、填空题	115
3.6.2 与 JavaScript 结合的实例编码	75	二、判断题	115
3.6.3 JUnit 测试程序编码和运行	79	三、问答题	115
3.6.4 Cactus 测试程序编码和运行	81	四、上机实习题	115
3.6.5 程序运行说明	83	第 5 章 飞机票定价管理实例	116
3.6.6 与 JavaScript 结合的程序运行 说明	85	5.1 飞机票定价管理的实例构思	116
3.6.7 数据库的压力测试	86	5.2 飞机票定价管理的 JSP 知识	116
3.6.8 互联网访问压力测试	88	5.3 飞机票定价管理的建模分析	116
3.7 实例的剖析与升华	91	5.3.1 用例图	117
3.8 小结	91	5.3.2 活动图	117
3.9 问与答	91	5.3.3 类图	117
综合练习三	91	5.4 飞机票定价管理的数据	117
一、填空题	91	5.5 飞机票定价管理的文件结构	117
二、判断题	91	5.6 飞机票定价管理的 B/S 结构实例	118
三、问答题	91	5.6.1 实例编码	118
四、上机实习题	92	5.6.2 与 JavaScript 结合的实例编码	124
第 4 章 飞机票查询实例	93	5.6.3 JUnit 测试程序编码和运行	126
4.1 飞机票查询的实例构思	93	5.6.4 Cactus 测试程序编码和运行	128
4.2 飞机票查询的 JSP 知识	93	5.6.5 程序运行说明	129
4.3 飞机票查询的建模分析	94	5.6.6 与 JavaScript 结合的程序 运行说明	131
4.3.1 用例图	94	5.6.7 互联网访问压力测试	132
4.3.2 活动图	94	5.7 飞机票管理的互联网程序集成	134
4.3.3 类图	94	5.7.1 飞机票管理的互联网程序 集成编码	134
4.4 飞机票查询的数据	94	5.7.2 飞机票管理的互联网程序 运行说明	138
4.5 飞机票查询的文件结构	96	5.8 小结	140
4.6 飞机票查询的 B/S 结构实例	97	5.9 问与答	140
4.6.1 实例编码	97	综合练习五	140
4.6.2 与 JavaScript 结合的实例编码	103	一、填空题	140
4.6.3 JUnit 测试程序编码和运行	106	二、判断题	141
4.6.4 Cactus 测试程序编码和运行	108	三、问答题	141
4.6.5 程序运行说明	109	四、上机实习题	141
4.6.6 与 JavaScript 结合的程序 运行说明	111	第 6 章 员工录入实例	142
4.6.7 互联网访问压力测试	112	6.1 员工录入的实例构思	142
4.7 实例的剖析与升华	114	6.2 员工录入的 JSP 知识	142
4.8 小结	114	6.3 员工录入的建模分析	142
4.9 问与答	115		

6.3.1 用例图	143	7.6.2 与 JavaScript 结合的实例编码	203
6.3.2 活动图	143	7.6.3 JUnit 测试程序编码和运行	207
6.3.3 类图	143	7.6.4 Cactus 测试程序编码和运行	209
6.4 员工录入的数据	143	7.6.5 程序运行说明	211
6.4.1 数据字典	143	7.6.6 与 JavaScript 结合的程序运行 说明	214
6.4.2 数据表创建的 SQL 代码	144	7.6.7 互联网访问压力测试	215
6.4.3 数据添加的 SQL 代码	145	7.7 实例的剖析与升华	217
6.4.4 数据表创建的 JSP 文件	146	7.8 小结	217
6.4.5 数据添加的 JSP 文件	147	7.9 问与答	217
6.5 员工录入的文件结构	149	综合练习七	217
6.6 员工录入的 B/S 结构实例	149	一、填空题	217
6.6.1 实例编码	150	二、判断题	218
6.6.2 与 JavaScript 结合的实例编码	166	三、问答题	218
6.6.3 JUnit 测试程序编码和运行	174	四、上机实习题	218
6.6.4 Cactus 测试程序编码和运行	176	第 8 章 员工工资管理实例	219
6.6.5 程序运行说明	180	8.1 员工工资管理的实例构思	219
6.6.6 与 JavaScript 结合的程序运行 说明	183	8.2 员工工资管理的 JSP 知识	219
6.6.7 数据库访问压力测试	184	8.3 员工工资管理的建模分析	220
6.6.8 互联网访问压力测试	186	8.3.1 用例图	220
6.7 实例的剖析与升华	189	8.3.2 活动图	220
6.8 小结	189	8.3.3 类图	220
6.9 问与答	189	8.4 员工工资管理的数据	220
综合练习六	189	8.4.1 数据字典	221
一、填空题	189	8.4.2 数据表创建的 SQL 代码	221
二、判断题	189	8.4.3 数据添加的 SQL 代码	221
三、问答题	190	8.4.4 数据表创建的 JSP 文件	221
四、上机实习题	190	8.4.5 数据添加的 JSP 文件	223
第 7 章 员工查询实例	191	8.5 员工工资管理的文件结构	224
7.1 员工查询的实例构思	191	8.6 员工工资管理的 B/S 结构实例	225
7.2 员工查询的 JSP 知识	191	8.6.1 实例编码	225
7.3 员工查询的建模分析	192	8.6.2 与 JavaScript 结合的实例编码	236
7.3.1 用例图	192	8.6.3 JUnit 测试程序编码和运行	240
7.3.2 活动图	192	8.6.4 Cactus 测试程序编码和运行	242
7.3.3 类图	192	8.6.5 程序运行说明	245
7.4 员工查询的数据	193	8.6.6 与 JavaScript 结合的程序运行 说明	248
7.5 员工查询的文件结构	195	8.6.7 数据库访问压力测试	249
7.6 员工查询的 B/S 结构实例	195	8.6.8 互联网访问压力测试	251
7.6.1 实例编码	196		

8.7 员工管理的互联网程序集成	253	9.7 实例的剖析与升华	297
8.7.1 员工管理的互联网程序 集成编码	253	9.8 小结	297
8.7.2 员工管理的互联网程序 运行说明	258	9.9 问与答	297
8.8 小结	260	综合练习九	297
8.9 问与答	260	一、填空题	297
综合练习八	261	二、判断题	298
一、填空题	261	三、问答题	298
二、判断题	261	四、上机实习题	298
三、问答题	261	第 10 章 安装程序制作与发布	299
四、上机实习题	261	10.1 互联网程序默认文件的设置与 运行	299
第 9 章 用户管理 (多媒体版) 实例	262	10.1.1 互联网程序默认文件的设置	299
9.1 用户管理 (多媒体版) 的实例构思	262	10.1.2 互联网程序默认文件的运行	300
9.2 用户管理 (多媒体版) 的 JSP 知识	262	10.2 Ant 工具	300
9.3 用户管理 (多媒体版) 的建模分析	262	10.2.1 Ant 工具的下载与安装	300
9.3.1 用例图	263	10.2.2 员工管理的 Ant 安装程序制作	301
9.3.2 活动图	263	10.2.3 员工管理的 Ant 安装程序运行	303
9.3.3 类图	263	10.3 员工管理互联网程序在 WebLogic 7、 8 的部署与运行	304
9.4 用户管理 (多媒体版) 的数据	263	10.3.1 员工管理互联网程序在 WebLogic 7、8 的部署	304
9.4.1 数据字典	263	10.3.2 员工管理互联网程序在 WebLogic 7、8 的运行	306
9.4.2 数据表创建的 SQL 代码	264	10.4 员工管理互联网程序在 WebLogic 9 的部署与运行	306
9.4.3 数据添加的 SQL 代码	264	10.4.1 员工管理互联网程序在 WebLogic 9 的部署	306
9.4.4 数据表创建的 JSP 文件	264	10.4.2 员工管理互联网程序在 WebLogic 9 的运行	310
9.4.5 数据添加的 JSP 文件	266	10.5 小结	310
9.4.6 图片的写入 JSP 文件	267	10.6 问与答	310
9.4.7 图片的读出 JSP 文件	269	综合练习十	310
9.5 用户管理 (多媒体版) 的文件结构	270	一、填空题	310
9.6 用户管理 (多媒体版) 的 B/S 结 构实例	270	二、判断题	311
9.6.1 实例编码	271	三、问答题	311
9.6.2 与 JavaScript 结合的实例编码	284	四、上机实习题	311
9.6.3 JUnit 测试程序编码和运行	287	参考文献	312
9.6.4 Cactus 测试程序编码和运行	289		
9.6.5 程序运行说明	289		
9.6.6 与 JavaScript 结合的程序 运行说明	292		
9.6.7 数据库的压力测试	293		
9.6.8 互联网访问压力测试	295		

第 1 章 用户管理实例

本章教学目标:

- (1) 了解用户管理的实例构思、JSP 知识和建模分析。
- (2) 掌握用户管理的数据字典、数据表创建和数据添加的 SQL 代码。
- (3) 了解用户管理的文件结构。
- (4) 掌握用户管理的 B/S 结构的程序编码、与 JavaScript 结合的 B/S 结构的程序编码。
- (5) 掌握用户管理的测试程序、B/S 结构程序、压力测试程序的运行。

1.1 用户管理的实例构思

用户管理实例是互联网应用程序，实现的功能如下所示：

- (1) 可以添加新用户。
- (2) 可以删除旧用户。
- (3) 可以修改用户名、密码和用户权限。

1.2 用户管理的 JSP 知识

用户管理包括的 JSP 知识如下所示：

- (1) 互联网程序在 Tomcat 服务器的部署和运行。
- (2) page 标识、include 标识的应用。
- (3) 声明标识、表达式标识、帮助标识和程序标识的应用。
- (4) jsp:useBean 标识的应用。
- (5) 表单标识、编辑框标识、按钮标识、复选框标识、列表标识、隐藏域标识的应用。
- (6) JavaScript 的应用。
- (7) 变量、操作符、控制语句、字符串类、数字类、变量的互相转换、数组、异常处理的应用。
- (8) out 对象、request 对象、response 对象、session 对象、application 对象、config 对象的应用。
- (9) JavaBean 的应用。
- (10) Servlet 的应用。
- (11) 数据库的应用。

1.3 用户管理的建模分析

本节讲解用户管理实例的用例图、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，如果要了解 UML 建模语言和该工具的应用，请参考《Java 程序设计 35 讲》的第 30 讲。

1.3.1 用例图

用例图表示用户与程序模块的关系，用户管理实例的注册用户都可以应用用户管理互

联网程序，如图 1-1 所示。

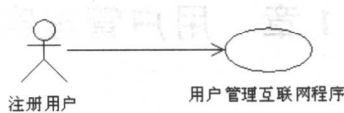


图 1-1

1.3.2 活动图

本节讲解用户管理程序的活动图。用户管理程序的运行流程是注册用户打开用户管理页面，进行重新显示用户、添加用户、修改用户和删除用户操作。活动图如图 1-2 所示。

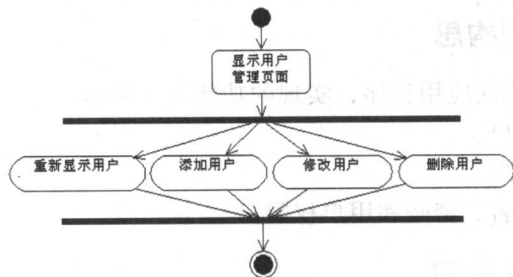


图 1-2

1.3.3 类图

本节介绍用户管理的 B/S 结构的类图，如图 1-3 所示。

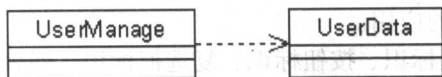


图 1-3

UserManage 类是 Servlet 程序，应用 UserData 类的方法进行各种数据处理。

1.4 用户管理的数据

本节介绍用户管理的数据字典、数据表创建和数据添加的 SQL 代码。

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

续表 1-1

	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
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《JSP 动态网站设计教程》的 12.1 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 JSPBaseExample 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《JSP 动态网站设计教程》的 12.2 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)
```

1.4.4 数据表创建的 JSP 文件

本节应用的项目是 UserBSPro, 保存在附带程序包的 C1 目录。数据表创建的 JSP 文件的创建步骤如下所示:

(1) 创建项目 UserCMSPro, 选择 Tools 菜单→Configure→Libraries 选项 (JBuilder X

选择 Tools 菜单→Configure Libraries 选项)添加 SQL Server 2005 类包 (sqljdbc.jar 文件), 然后将类包加入 UserBSPPro 项目, 如图 1-4 所示。如果不熟悉类包添加具体的步骤, 可以参考《Java 程序设计经典教程》的 12.3.2 节或者《JBuilder 程序设计经典教程》的 3.5 节。

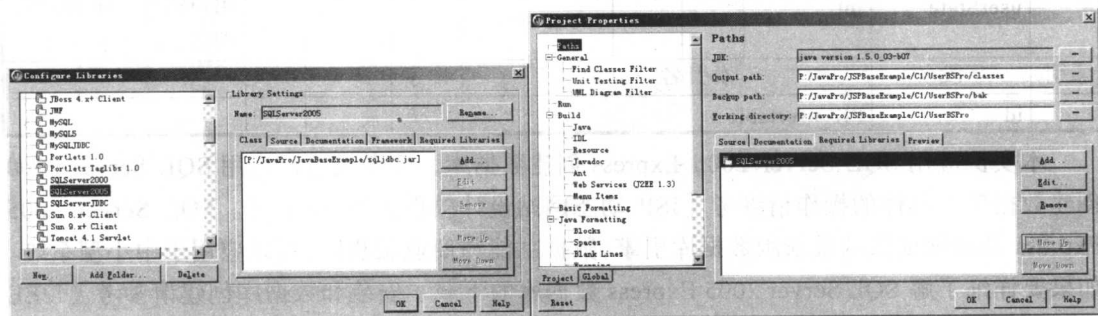


图 1-4

注意: 如果要在 JBuilder 中应用 SQL Server JDBC 类包, 如在设计面板显示数据, 需要将 sqljdbc.jar 复制到 (JBuilder 的安装目录) \lib, 再重新启动 JBuilder, 便可以正常在 JBuilder 的设计面板显示数据。

(2) 创建互联网模块, 名字和运行路径是 usermanage, JSP 的版本设为 1.2, Servlet 的版本设为 2.3, 如图 1-5 所示, 互联网模块的创建参考《JSP 动态网站设计教程》的 2.3.5 节。读者也可以将 JSP 的版本设为 2.0, Servlet 的版本设为 2.4 版本, 但 2.0 版本的 JSP 只能在 Tomcat 5.x 上运行。

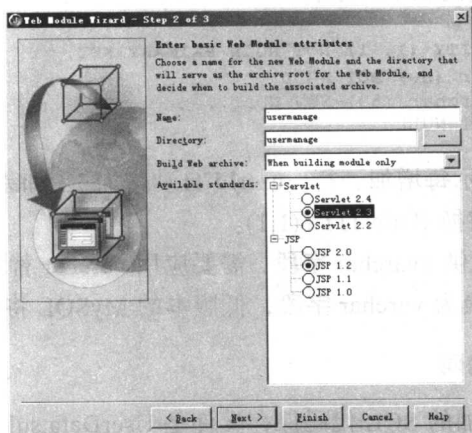


图 1-5

(3) 创建数据表创建的 JSP 文件, 名字是 usercreate.jsp, 代码如下所示 (JSP 文件的创建参考《JSP 动态网站设计教程》的 2.3.6 节)。

```
<%@ page contentType="text/html; charset=GBK" %>
<%@ page import="java.sql.*" %>
<html>
<head><title>数据表创建页面</title></head>
<body>
<%
```

//定义数据库连接的驱动程序

```
1: String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
```

```
//定义 SQL Server 数据库的连接地址, 连接实例是 SQL2005EXPRESS
2: String url = "jdbc:sqlserver://localhost\\SQL2005EXPRESS;databaseName=UserC1";
//如果是默认实例, 不用定义 SQL2005EXPRESS
//String url = "jdbc:sqlserver://localhost;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
3: conn = DriverManager.getConnection(url, "sa", "test");
    if(conn != null){
        out.println("<h3>成功连接 JDBC 数据源.</h3>");
    }
    Statement stmt = conn.createStatement();
    try{
        ResultSet rs = stmt.executeQuery("select * from usertable");
        if(rs != null){
            out.println("<h3>数据库 UserC1 已经存在 usertable 数据表.</h3>");
        }
    }catch(Exception ex){
        String sql = "CREATE TABLE usertable ("
            + "userId int IDENTITY (1, 1) NOT NULL PRIMARY KEY,"
            + "userName nvarchar (50) NOT NULL ,"
            + "userPassword nvarchar (50) NOT NULL ,"
            + "userState int NOT NULL "
            + ")";
        //创建数据表 usertable
        stmt.executeUpdate(sql);
        out.println("<h3>在数据库 UserC1 成功创建 usertable 数据表.</h3>");
    }
    }catch(Exception ex){
        ex.printStackTrace();
    }
    try{
        //关闭数据库连接类
        conn.close();
    }catch(Exception ex){
        ex.printStackTrace();
    }
}
%>
</body>
</html>
```

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 驱动程序名字是 com.microsoft.sqlserver.jdbc.SQLServerDriver, 该驱动程序名字可以连接 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。

(4) 在 JBuilder 启动 Tomcat 服务器, 在浏览器输入 http://localhost:8080/usermanage/usercreate.jsp 运行 usercreate.jsp 文件, 结果如图 1-6 所示 (JSP 文件的运行参考《JSP 动态

网站设计教程》的 2.3.6 节)。



图 1-6

1.4.5 数据添加的 JSP 文件

创建数据添加的 JSP 文件, 名字是 userdataadd.jsp, 保存目录是 usermanage, 代码如下所示:

```
<%@ page contentType="text/html; charset=GBK" %>
<%@ page import="java.sql.*" %>
<html>
<head><title>数据添加页面</title></head>
<body>
<%
//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){
    out.println("<h3>成功连接 JDBC 数据源.</h3>");
}
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from usertable");
if (rs.next()) {
    out.println("<h3>数据表 usertable 已经存在数据.</h3>");
}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();
    out.println("<h3>在数据表 usertable 成功添加 2 条记录.</h3>");
}
}catch(Exception ex){
    ex.printStackTrace();
}
try{
//关闭数据库连接类
conn.close();
}catch(Exception ex){
    ex.printStackTrace();
}
```

```
%>  
</body>  
</html>
```

运行结果如图 1-7 所示。

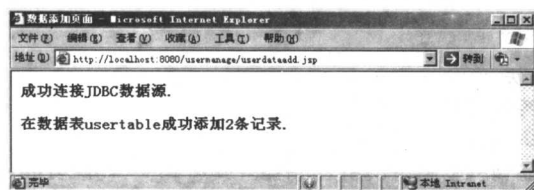


图 1-7

1.5 B/S 结构的用户管理的文件结构

本例的项目保存在附带程序包的 C1\UserBSPPro 目录，文件结构如图 1-8 所示。

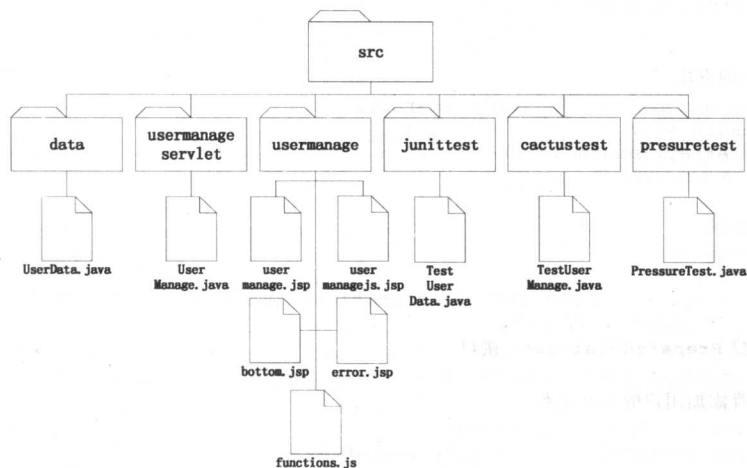


图 1-8

data 目录保存数据处理类文件，usermanageservlet 目录保存用户管理 Servlet 类文件，usermanage 目录保存用户管理的 JSP 文件，junittest 目录保存 JUnit 测试类文件，cactustest 目录保存 Cactus 测试类文件，presuretest 目录保存数据库压力测试文件。

1.6 用户管理的 B/S 结构实例

本节讲解用户管理 B/S 结构的实例编码，与 JavaScript 结合的实例编码，JUnit 测试程序编码和运行，Cactus 测试程序编码和运行，实例的运行，数据库访问压力测试和互联网访问压力测试。

1.6.1 实例编码

(1) 本节应用的项目是 UserBSPPro。为项目加入 SQL Server 类包文件，保存在附带程序包的 JSPBaseExample 目录，文件名是 sqljdbc.jar。

(2) 本节应用的互联网模块是 usermanage，名字和运行路径是 usermanage，JSP 的版本设为 1.2，Servlet 的版本设为 2.3，互联网模块的创建参考《JSP 动态网站设计教程》的

2.3.5 节。

(3) 创建 UserData 类文件，保存目录是 data，写入数据处理方法，代码如下所示：

```
package data;
import java.sql.*;
public class UserData {
    Connection conn = null;
    //取得数据库连接的方法
    public void setupConn(){
        try{
            if(conn == null){
                Class.forName("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";
                //使用 DriverManager 类的 getConnection() 方法建立连接
                conn = DriverManager.getConnection(url, "sa", "test");
            }
        }catch(Exception e){
            e.printStackTrace();
        }
    }
    //创建用户的方法
    public int createUser(String[] userArray) {
        int result = 0;
        if(userArray.length != 3){
            return result;
        }
        //建立连接
        setupConn();
        try {
            String sql = "INSERT INTO usertable(userName, userPassword, userState) "
                + "VALUES(?, ?, ?)";
            //创建 PreparedStatement 接口
            PreparedStatement insertUser = conn.prepareStatement(sql);
            //设置添加用户的 3 个参数
            for(int i = 0; i < 3; i++){
                insertUser.setString(i + 1, userArray[i]);
            }
            //成功添加记录，返回 1
            result = insertUser.executeUpdate();
        }catch (Exception ex) {
            ex.printStackTrace();
        }
        return result;
    }
    //更新用户的方法
    public int updateUser(String[] userArray) {
        int result = 0;
        if(userArray.length != 4){
            return result;
        }
        //建立连接
        setupConn();
        try {
            //第 1 个 SCROLL_SENSITIVE 参数表示反映其他用户的更新操作
            //第 2 个 CONCUR_UPDATABLE 参数表示数据集可更新
            Statement stmt = conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                ResultSet.CONCUR_UPDATABLE);
            String sql = "select * from usertable where userId = '" +
                userArray[0] + "'";
            //取得要修改的记录
            ResultSet rs = stmt.executeQuery(sql);
```