



普通高等教育“十一五”国家级规划教材
高职高专计算机系列规划教材

基于工作任务的 Java Web应用教程

覃国蓉 毛树生 叶建锋 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

普通高等教育“十一五”国家级规划教材
高职高专计算机系列规划教材

基于工作任务的 Java Web 应用教程

覃国蓉 毛树生 叶建锋 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

PetStore 是 SUN 公司推出的一个宠物商店的系统,是学习 J2EE 技术的一个绝好例子。本教材围绕 PetStore 的实现介绍 Java Web 应用开发技术:HTML、CSS、JSP、JSTL、EL、JavaBean、Struts、Spring 和 Hibernate。PetStore 的实现采用基于原型迭代的实现思路,将整个过程分为 9 个工作任务,各章节以完成工作任务为目标讲解相关技术。从最简单的用 HTML+CSS 实现宠物分类展现模块的静态版本开始,每一章中讲解的新技术和方法,都是建立在前一章的基础之上的,从而使能够循序渐进地进行学习,到最后能够编写出 Java 高手编写的代码。

贯穿本教材的项目是对经典开源项目进行教学适用化(逆向工程、本地化改造和中间版本的开发)改造的成果,具有技术典型性、代码规范性和教学适用性。本教材适合作为高等职业院校的计算机应用与软件技术专业教材,也可作为自学和急需了解 Java Web 应用相关开发技术和知识、对 Java 开源项目感兴趣的技术人员的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

基于工作任务的 Java Web 应用教程 / 覃国蓉, 毛树生, 叶建锋编著. —北京: 电子工业出版社, 2009.12
(普通高等教育“十一五”国家级规划教材. 高职高专计算机系列规划教材)

ISBN 978-7-121-09909-0

I. 基… II. ①覃… ②毛… ③叶 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(2009)第 211149 号

策划编辑: 吕 迈

责任编辑: 徐云鹏

印 刷: 北京市李史山胶印厂

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 18.25 字数: 464 千字

印 次: 2009 年 12 月第 1 次印刷

印 数: 3 000 册 定价: 28.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

PetStore 是 SUN 公司推出的一个宠物商店的系统,是学习 J2EE 技术的一个绝好例子。后来很多开源社区又推出了它的不同轻量级版本,如 Spring 开发包中的示例程序 JPetStore,Java 开发者社区中的 Hibernate JPetStore 等,该系统的不同版本是世界各地优秀程序员智慧的结晶,所以本教材选择该系统作为案例,让学生在实现该系统的过程中掌握 Java Web 应用系统开发的技术,获得丰富的软件开发经验,最终能够成为一个合格的 Java Web 应用开发工程师,具备解决企业中最实际的问题、做出实际的软件产品和项目的最实用的技能。

本教材具有以下特色:

1. 基于工作任务,体现了任务驱动的行为导向教学法。每章的开始都有一个小工作任务描述,接着是学习要点说明,然后是相关知识介绍,最后是案例项目的实现。

2. 选择经典的、开源的技术示范项目 PetStore 作为贯穿本教材的案例项目,真正实现了“够用为度”。Java 技术一直在不断发展,Java Web 应用开发技术要讲解到什么程度,无疑其示范项目最有发言权。

3. 采用基于原型迭代的软件开发方法的教学法,符合高职学生和课程的特点。与 Java 技术相关的共有 9 个工作任务,从最简单的用 HTML+CSS 实现宠物分类展现模块的静态版本开始,每一章中讲解的新技术和方法,都是建立在前一章的基础之上的,从而使学生能够循序渐进地进行学习,到最后能够编写出 Java 高手编写的代码。

4. “陈述性知识”和“过程性知识”并重。示范项目由软件厂商或一线软件开发人员来完成,包含了完成示范项目的众多优秀程序员的最佳实践经验。本教材代码是对开源项目的代码进行教学实用化改造而来的,所以本教材不只是介绍相关技术(陈述性知识),还可以让学生学到很多经验性的知识(过程性知识):编码规范,设计技巧和编程模式等。

为了便于教学,本教材为教师提供配套的 PPT 课件、课后习题答案和工作任务需要完成的程序源代码。欢迎登录华信教育资源网(<http://www.hxedu.com.cn>),免费获取本书的教学资源。

本教材在完成过程中得到了很多人的帮助,在此表示感谢:

- 首先感谢 Pprun,他是 Hibernate JPetStore 的实现者,本教材的案例项目就是在对 Hibernate JPetStore 改造的基础上完成的,他还在开源技术的使用方面给予了我们很多宝贵经验,正是像 Pprun 这些开源技术爱好者的无私奉献,Java 技术才得以快速发展和广泛使用。
- 其次感谢以电子工业出版社的吕迈老师、徐云鹏老师为代表的专业出版团队以及他们的分社社长赵云锋先生,他们的认真负责和专业技术是本书能够出版的保证。
- 最后感谢成都电子机械高等专科学校的刘甫迎教授,河北农业大学信息学院杨学全

副教授，北京信息职业技术学院常鹏老师等同行以及同事李华忠博士，李亮老师，袁晖老师，张建辉老师和魏勇老师等给予的宝贵建议。

由于时间仓促，加之作者水平有限，不当及错误之处在所难免，恳请读者批评指正。作者的 E-mail 地址是 qingr@szit.com.cn。

编 者

2009 年 10 月

目 录

第 1 章 背景知识	1
任务 1 开发一个简单的个人网站	1
1.1 C/S 架构与 B/S 架构	4
1.1.1 C/S 架构	4
1.1.2 B/S 架构	5
1.2 Web 相关概念	5
1.3 静态网页和动态网页的处理流程	6
1.3.1 静态网页的处理流程	6
1.3.2 动态网页的处理流程	7
1.4 Web 应用开发技术	8
1.4.1 静态网页开发技术	8
1.4.2 动态网页开发技术	9
1.4.3 企业应用解决方案: J2EE 与 .NET	10
1.4.4 Java 轻量级企业应用解决方案: Struts+Spring+Hibernate	10
1.4.5 更佳的用户体验: Ajax	10
1.5 PetStore 项目简介	11
1.6 任务介绍及后面章节内容组织	11
作 业	11
第 2 章 用 HTML+CSS 实现宠物分类展现静态网页版本	13
任务 2 使用 HTML+CSS 实现宠物分类展现模块的静态网页版本	13
2.1 HTML	15
2.1.1 HTML 的基本知识	15
2.1.2 HTML 的文档结构	15
2.1.3 HTML 标记的公共属性	16
2.1.4 HTML 常用标记	17
2.1.5 HTML 表单	18
2.2 CSS	19
2.2.1 CSS 的分类	20
2.2.2 CSS 的语法	20
2.2.3 CSS 的选择器	21
2.2.4 CSS 的伪类	22
2.2.5 CSS 的盒子模式	22
2.2.6 CSS 的常用属性	23
2.3 宠物分类展现的界面及 Web 应用开发步骤	24
2.3.1 宠物分类展现的界面	24

2.3.2	Web 应用的开发步骤	26
2.4	实现思路	29
2.4.1	宠物分类展现模块的术语表	29
2.4.2	基于原型迭代的实现思路	30
2.4.3	宠物分类展现模块的发布和迭代计划	30
2.5	实现主页面主体部分 Main_body.html	31
2.5.1	使用图像、超链接、换行和空格标记实现主页面的左边导航条部分	31
2.5.2	<map>、<area>标记实现主页面的图片导航	32
2.5.3	通过层 div 标记进行布局, 组装成主页面的主体部分	34
2.5.4	通过 CSS 设置效果	34
2.6	实现品种列表页面、商品列表页面和宠物信息页面主体部分 Category_body.html, Product_body.html, Item_body.html	36
2.6.1	实现品种列表页面主体部分 Category_body.html	36
2.6.2	实现商品列表页面主体部分 Product_body.html	39
2.6.3	实现宠物详细信息页面主体部分 Item_body.html	39
2.7	实现网页的公共部分 IncludeTop.html, IncludeBottom.html	39
2.7.1	实现网页的公共底部 IncludeBottom.html	39
2.7.2	实现网页的公共顶部 IncludeTop.html	41
2.8	组装调试	44
作 业		45
第 3 章	宠物分类展现模块数据库准备及 JDBC	46
任务 3	为宠物分类展现准备数据并完成各表对应的 DAO 类	46
3.1	宠物分类展现模块数据准备	49
3.1.1	安装 MySQL 及其界面管理工具 Navicat	49
3.1.2	在 MySQL 中创建一个数据库 petstore	50
3.1.3	分类表 category 的创建	53
3.1.4	品种表 product 的创建	54
3.1.5	供应商表 supplier 的创建	55
3.1.6	系列表 item 的创建	56
3.1.7	库存表 inventory 的创建	58
3.1.8	插入测试数据	59
3.1.9	为宠物商城系统创建一个访问数据库 petstore 的用户	60
3.2	JDBC 数据库编程	60
3.2.1	安装 MySQL 的驱动程序	61
3.2.2	JDBC 应用程序的模板代码	61
3.2.3	编写 JDBC 应用程序修改数据库	63
3.2.4	编写封装创建数据库连接的类	66
3.3	POJO+DAO 的访问数据库编程模式	68
3.3.1	编写表结构对应的 POJO 类	68
3.3.2	设计访问各表的 DAO 类	69

3.3.3	编写访问各表的 DAO 类	71
3.3.4	DAO 类的使用	74
作 业		75
第 4 章	用 Hibernate 优化宠物分类展现 DAO 类	77
任务 4	使用 Hibernate 优化宠物分类展现 DAO 类	77
4.1	Hibernate 简介	78
4.2	使用 Hibernate 的准备工作	79
4.2.1	Hibernate 开发环境的安装配置	79
4.2.2	在项目中添加 Hibernate 配置文件	79
4.2.3	准备好 POJO 类和映射文件	80
4.3	用 Hibernate 访问数据库	81
4.3.1	Hibernate 的编程模式	81
4.3.2	使用 Hibernate 实现数据的插入	82
4.3.3	使用 Hibernate 实现数据的删除和修改	83
4.3.4	使用 Hibernate 实现数据的加载	84
4.3.5	使用 Hibernate 实现数据的查询	85
4.4	使用 Hibernate 重写 DAO 类	86
4.4.1	使用 Hibernate 重写 BaseDAO 类	87
4.4.2	BaseDAO 类的使用	89
4.4.3	基于 BaseDAO 改写 CategoryDAO 类	90
4.4.4	基于 BaseDAO 改写 ProductDAO 类	90
作 业		91
第 5 章	宠物分类展现模块的第一个动态版本的实现	92
任务 5	使用 JSP+POJO+DAO+DB 实现宠物分类展现的动态网页版本	92
5.1	JSP 简介	93
5.1.1	什么是 JSP	93
5.1.2	JSP 的页面组成	93
5.1.3	JSP 指令	95
5.1.4	JSP 动作	96
5.2	JSP 常用内部对象	96
5.2.1	request 与请求参数	97
5.2.2	out	98
5.2.3	session	98
5.3	宠物分类展现网页动态版本开发准备	99
5.3.1	实现思路	99
5.3.2	在 web.xml 中设置欢迎页面	99
5.4	用 JSP 实现主页面 Main.jsp	100
5.4.1	由 IncludeTop.html 得到网页顶部文件的动态版本 IncludeTop.jsp	100
5.4.2	由 IncludeBottom.html 得到网页底部文件的动态版本 IncludeBottom.jsp	101
5.4.3	由 Main_body.html 得到主页面 Main.jsp	101

5.5	用 JSP 实现品种列表 Category.jsp	102
作 业		104
第 6 章	用 JavaBean、EL 和 JSTL 优化宠物分类展现页面	107
任务 6	使用 JSTL、EL 表达式和 JavaBean 优化宠物分类展现的页面代码	107
6.1	JavaBean	108
6.1.1	JavaBean 简介	108
6.1.2	在 JSP 中使用 JavaBean	108
6.2	EL 表达式	109
6.2.1	EL 表达式简介	109
6.2.2	JSP EL 中的有效表达式	110
6.2.3	EL 隐含对象	110
6.3	JSTL 标签	111
6.3.1	JSTL 标签简介	111
6.3.2	常用 JSTL 标签	112
6.4	优化宠物分类展现页面	117
6.4.1	使用<jsp:useBean>去掉宠物分类展现页面中的 new 语句	117
6.4.2	用 EL 表达式和 JSTL 标签简化宠物分类展现页面代码	117
6.4.3	通过迭代使用 EL 表达式点符号简化对象属性的输出	120
6.5	使用统一的业务接口	122
6.5.1	设计一个系统共享的业务接口 PetStoreFacade	122
6.5.2	设计接口 PetStoreFacade 的实现类 PetStoreImpl	123
6.5.3	用 PetStoreImpl 实现宠物分类展现各页面	124
作 业		125
第 7 章	用 Struts 实现宠物商城的登录功能	127
任务 7	使用 Struts 实现宠物商城的用户登录功能	127
7.1	Struts 工作原理	128
7.1.1	MVC 设计模式	128
7.1.2	Struts 简介	128
7.1.3	Struts 的运行流程	129
7.1.4	Struts 开发环境配置	130
7.2	用户登录界面和 MVC 模块划分	131
7.2.1	用户登录的界面及流程	131
7.2.2	用 Struts 实现用户登录的要点	132
7.2.3	用户登录的实现思路	133
7.3	用户登录 Model 层的实现	134
7.3.1	在数据库中创建表 account	134
7.3.2	创建表 account 对应的 POJO 类 Account	136
7.3.3	创建表 account 和 Account 类的 Hibernate 映射文件	137
7.3.4	创建表 account 对应的数据库访问 DAO 类	138
7.3.5	修改宠物商城业务逻辑接口 PetStoreFacade 和实现类 PetStoreImpl	139

7.3.6 创建封装用户录入数据的 AccountForm	142
7.4 用户登录 View 层的实现	143
7.4.1 用户登录界面	143
7.4.2 用户登录成功的界面	145
7.4.3 用户登录失败的界面	147
7.5 用户登录 Controller 层的实现	148
7.5.1 处理用户登录的 SignonAction 的实现	148
7.5.2 处理用户退出的 SignoffAction 的实现	150
7.6 Struts 相关的配置文件	151
7.6.1 在 web.xml 中配置	151
7.6.2 创建 Struts 核心配置文件 struts-config.xml	154
7.7 用 Struts 实现主页显示	156
7.7.1 在 struts-config.xml 中为主页显示配置一个简单的 forward action	156
7.7.2 修改欢迎页面 index.jsp 的代码	157
作 业	158
第 8 章 用 Struts 完善宠物商城的 catalog 模块	160
任务 8 使用 Struts 完善宠物分类展现和宠物查找模块	160
8.1 用 Struts 优化完善宠物分类展现页面	163
8.1.1 定义为品种列表页面准备数据的 Action	163
8.1.2 定义相关的 ActionForm	165
8.1.3 配置 ActionForm 和 Action	165
8.1.4 修改项目中对 Category.jsp 的页面请求为对 Action 的请求	166
8.1.5 相应修改 Category.jsp 的代码	167
8.2 用 Struts 实现宠物查找功能	168
8.2.1 宠物查找界面及实现思路	168
8.2.2 定义封装查找关键字的 ActionForm	170
8.2.3 定义处理查找功能的 Action 并进行配置	171
8.2.4 在 PetStoreFacade 和 PetStoreImpl 中增加支持查找的方法	172
8.2.5 在 ProductDAO 中增加支持查找的方法	173
8.2.6 创建查找结果页面 SearchProducts.jsp	174
8.2.7 创建查找失败页面 noPet.jsp	174
8.3 Struts 深入	175
8.3.1 使用 Struts 标签优化保留查询条件	175
8.3.2 在 ActionForm 中进行验证	176
作 业	177
第 9 章 用 Spring 优化宠物商城的 Catalog 和用户登录模块	179
任务 9 使用 Spring 改写 Catalog 和用户登录模块	179
9.1 Spring 简介	180
9.1.1 Spring 简介	180
9.1.2 Spring 开发环境的安装配置	180

9.1.3	Spring 的控制反转和依赖注入	181
9.2	用 Spring 管理 PetStoreImpl 和各 DAO 类对象之间的依赖	182
9.2.1	使用配置文件 applicationContext.xml 定义依赖关系	182
9.2.2	使用 Spring 的依赖注入获取相关信息	184
9.3	使用 Spring 的依赖注入重写 catalog 模块	185
9.3.1	在 web.xml 中设置 applicationContext.xml 和 Listener	185
9.3.2	生成 BaseAction, 传递 petstore 对象	186
9.3.3	重写已经完成的 Action 并修改 struts-config.xml 文件	187
9.4	使用 Spring 简化 Hibernate 编程	189
9.4.1	继承 HibernateDaoSupport 实现 BaseDAO 类	189
9.4.2	在 Spring 配置文件中注入 sessionFactory	190
9.4.3	使用 import 简化配置文件	191
9.5	增加分页显示功能	192
9.5.1	分页显示的实现思路	192
9.5.2	使用 Spring 的 PagedListHolder 进行分页	194
9.5.3	修改相关的 Action	195
9.5.4	修改相关的 JSP 页面	197
作 业		198
第 10 章	完成宠物商城的购物车模块	200
任务 10	完成宠物商城的购物车功能	200
10.1	购物车的页面及流程	201
10.2	购物车实现思路	203
10.2.1	购物车的 MVC 划分	203
10.2.2	“添加到购物车”的程序处理流程	204
10.2.3	“从购物车中删除”的程序处理流程	204
10.2.4	“更新购物车”的程序处理流程	204
10.2.5	结账页面的“上一页”、“下一页”链接的程序处理流程	204
10.3	“添加到购物车”功能的实现	204
10.3.1	定义 CartItem 类	204
10.3.2	定义 Cart 类	205
10.3.3	定义 CartActionForm	208
10.3.4	在 struts-cart.xml 中配置 CartActionForm	209
10.3.5	创建 AddItemToCartAction 并配置	210
10.3.6	修改 PetStoreFacade 接口, PetStoreImpl 类和 InventoryDAO 类	211
10.3.7	购物车页面/cart/Cart.jsp 的实现	212
10.4	购物车页面和结账页面分页的实现	214
10.5	“从购物车中删除”与“更新购物车”的实现	215
10.5.1	创建 RemoveItemFromCartAction 并配置	215
10.5.2	创建 UpdateCartQuantitiesAction 并配置	216
10.6	数据库连接池	216

10.6.1	什么是数据库连接池.....	217
10.6.2	在 Tomcat 中配置 DBCP 数据库连接池.....	217
10.6.3	测试数据库连接池.....	219
10.6.4	在 Spring 中使用数据库连接池.....	221
作 业		223
附录 A	Java 开发包 JDK 的下载与安装配置.....	225
附录 B	Tomcat 下载与安装配置及 Web 应用程序的部署.....	228
附录 C	MySQL 数据库下载、安装配置.....	234
附录 D	MySQL GUI Tools 下载、安装配置和使用.....	237
附录 E	MySQL 的 JDBC 驱动程序下载、安装、使用.....	253
附录 F	Eclipse+Lomboz+MySQL 与 JDBC 下载及使用.....	255
附录 G	插入数据的 SQL 语句.....	274

第1章 背景知识

任务1 开发一个简单的个人网站

一、任务说明

通过开发一个简单的静态网站并在此基础上增加动态的内容,让学生深刻理解 B/S 架构,体验 Web 应用程序特殊的开发流程,获取一些经验。具体包括:

1. 开发一个简单的个人主页,了解 Web 应用程序的部署过程。
2. 为个人主页增加动态内容,了解动态网页的实现方式。

二、开发环境准备

1. Java 开发包 JDK,推荐使用 JDK-1_5_0_12,可从 http://java.sun.com/javase/downloads/index_jdk5.jsp 免费下载。安装配置见本书附录。

2. Web 服务器 Tomcat,推荐使用 Tomcat 5.5.26,可从 <http://tomcat.apache.org/download-55.cgi> 免费下载,其中 apache-tomcat-5.5.26.zip 为解压版的,apache-tomcat-5.5.26.exe 为安装版的。安装配置及 Web 应用程序的部署见本书附录。

3. MySQL 数据库,推荐使用 MySQL 5.0,可以从 <http://dev.mysql.com/downloads/> 下载。安装配置和使用见本书附录。

4. MySQL GUI Tools,提供图形界面的 MySQL 服务器端管理和客户端查询浏览及其他辅助工具,可从 <http://dev.mysql.com/downloads/gui-tools/5.0.html> 下载。安装配置和使用见本书附录。

5. MySQL JDBC 连接包,即 MySQL 的 JDBC 驱动程序,可从 <http://dev.mysql.com/downloads/connector/j/5.1.html> 下载。安装和使用见本书附录。

三、完成过程

在下面的完成过程中 TOMCAT_HOME 代表 Tomcat 的安装路径。

1. 开发一个简单的静态个人网页。

(1) 在 TOMCAT_HOME\webapps 下建立子目录 test。

(2) 在记事本中录入以下代码并以文件名 myHomePage.html 保存到 TOMCAT_HOME\webapps\test 下。

```
<H1>Hello World</H1>
<P>Welcome to my home page. Please don't run away!</P>
<HR>
<P>
Email me:
<A HREF="mailto:webmaster@mywebsite.com">webmaster@mywebsite.com</A>
</P>
```

(3) 启动 Tomcat 服务器。

(4) 打开浏览器，在地址栏中录入 `http://localhost:8080/test/myHomePage.html`（目录和文件名大小写一定要一致），查看是否出现如图 1.1 所示的页面效果。



图 1.1 myHomePage.html 页面 xiaoguo

(5) 如果不对，可以检查代码，修改后再重复（4）。

2. 开发一个带网站计数器的个人主页。

(1) 在记事本中录入以下代码并以文件名 `myHomePage.jsp` 保存到 `TOMCAT_HOME\webapps\test` 下。

```
<%@ page contentType="text/html"%>
<%@page import="java.sql.*" %>
<%
    int counter=0;
    try{
        //加载 mysql 驱动类
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        //使用用户名 root 和密码 123，连接数据库 test
        Connection
conn=DriverManager.getConnection("jdbc:mysql://localhost/test","root","123");
        //创建语句对象
        Statement stmt=conn.createStatement();
        //用语句对象执行查询，将结果保存到结果集对象 rs 中
        ResultSet rs=stmt.executeQuery("select * from accessTimes ");
        //如果结果集不为空，取出第一记录的字段 times 的值
        if(rs.next()) counter=rs.getInt("times");
        //更新 times 字段的值，使其加 1
        stmt.executeUpdate("update accessTimes set times=times+1");
        //关闭结果集，语句和连接
        rs.close();
```

```

        stmt.close();
        conn.close();
    }catch(Exception e){e.printStackTrace();}

%>

<H1>Hello World</H1>
<P>Welcome to my home page. Please don't run away!</P>
<HR>
<P>
Email me:
<A HREF="mailto:webmaster@mywebsite.com">webmaster@mywebsite.com</A>
</P>

This page has been accessed <font color=red size=7><%=counter%></font> times

```

(2) 以用户名为 root，密码为 123（或其他 MySQL 安装时设置的用户名和密码）登录 MySQL，创建数据库 test，在其中创建表 accessTimes，该表只有一个整型（int）字段 times，并在其中插入一条 times 字段为 1 的记录。

(3) 在 TOMCAT_HOME\webapps\test 下创建子目录 WEB-INF，将 MySql JDBC 连接包 mysql-connector-java-5.1.6-bin.jar 复制到 WEB-INF\lib 下。

(4) 启动 Tomcat 服务器。

(5) 打开浏览器，在地址栏中录入 http://localhost:8080/test/myHomePage.jsp（目录和文件名大小写一定要一致），并刷新页面 3 次，查看是否出现如图 1.2 所示的页面效果。



图 1.2 myHomePage.jsp 页面刷新 3 次后的效果

(6) 如果不对，可以检查代码，修改后再重复（5）。

四、总结

1. Web 应用程序要遵循规定的部署。

2. 对比 myHomePage.html 和 myHomePage.jsp，可以发现，动态网页 myHomePage.jsp

就是在静态网页 myHomePage.html 中使用<%和%>标签加入了一些动态内容。

3. Web 应用程序使用的 Java 包要保存到特定的目录下, 这里是在 TOMCAT_HOME\webapps\test\WEB-INF\lib 下, 如这里的 MySQL 数据库连接包 mysql-connector-java-5.1.6-bin.jar。

4. JSP 通过在文件前面使用<%@page import="java.sql.*" %>引入 Java 包, 通过在<%和%>之间加入 Java 代码完成一些功能, 这里就是访问数据库获得访问次数, 通过<%=变量名%>输出变量的值。

5. 通过将访问次数放在数据库中并随时更新, 实现计数器。

6. 学生的其他总结和体会。

本章要点:

从程序员和最终用户的角度, 介绍 C/S 架构应用程序和 B/S 架构应用程序的特点。

从处理流程方面, 介绍静态网页和动态网页的不同。

介绍 Web 应用开发相关技术。

介绍本教材使用的案例系统: PetStore 系统。

介绍本教材工作任务及后续内容组织。

随着 Internet 技术的进步和 Web 的普及, B/S 架构, 即 Browser/Server (浏览器/服务器) 架构已经成为当今应用软件的首选体系结构。

1.1 C/S 架构与 B/S 架构

软件系统的结构可以分为 3 类: 单机版, 如记事本, 单机游戏; C/S 架构, 如 QQ; B/S 架构, 如网上银行。单机程序很简单, 也很好理解, 下面从开发人员和最终用户的角度对 C/S 架构和 B/S 架构进行分析。

1.1.1 C/S 架构

过去, 软件系统比较多地采用 C/S (Client/Server) 架构, C/S 架构也称为客户-服务器的请求响应模式, 如图 1.3 所示。这种架构的软件系统的特点是: 整个系统分为客户端和服务端, 客户端和服务端端的程序都需要程序员开发, 客户端和服务端端的通信部分是程序必须实现的部分, 也是难点; 用户要使用这类系统必须安装客户端。

我们常用的 QQ 就是 C/S 架构的软件系统, 腾讯公司需要开发客户端和服务端端的程序。我们只有安装 QQ 的客户端, 才可以通过腾讯公司的服务器与朋友聊天, 而且客户端要经常升级才能使用新的功能。

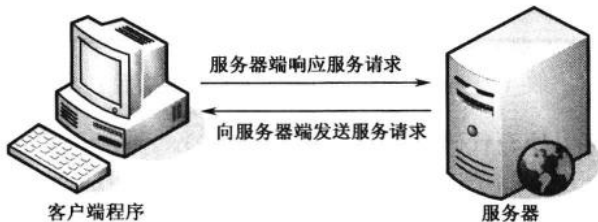


图 1.3 C/S 架构

1.1.2 B/S 架构

B/S 软件架构可以看做是特殊的 C/S，是对 C/S 的扩展。这种架构的软件系统其客户端都是通用的浏览器（如 IE 等），服务器端由 Web 服务器（如 IIS 或 Tomcat 等）和数据库服务器（如 SQL Server, Oracle, MySQL 等）构成，Web 页面和应用程序等都存放在 Web 服务器上，数据存放在数据库服务器中，如图 1.4 所示。

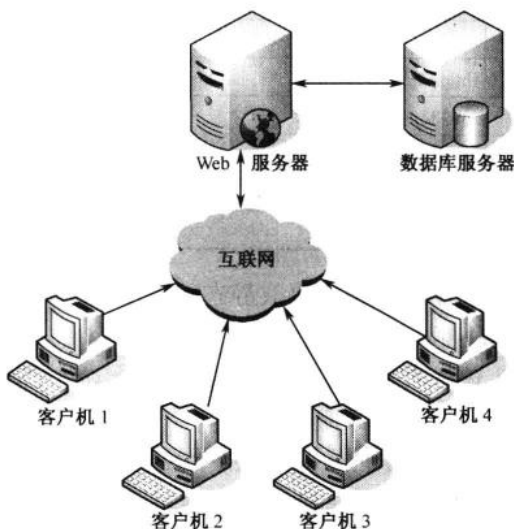


图 1.4 B/S 架构

客户端不需要开发。客户端和服务端之间的通信由 Web 服务器完成，也不需要开发。程序员只需要专注开发实现服务器端业务逻辑功能的网页和应用程序即可。由于客户端都是通用的浏览器，用户使用不需要安装额外的软件。

各大银行提供的网上银行就是比较典型的 B/S 架构的软件系统。

B/S 架构是 Web 技术发展的结果，所以 B/S 架构的软件系统又常被称为 B/S 应用程序或 Web 应用程序。

1.2 Web 相关概念

Web 或 World Wide Web 是互联网上提供的一项服务，为用户访问、管理和浏览信息提供交互环境。Web 的特点如下：

- 以客户/服务方式提供服务，服务端运行的是 Web 服务器程序，在客户端运行浏览器程序。浏览器与服务器通过 HTTP 协议进行通信。
- 信息以 Web 页面的形式提供，存储在 Web 服务器上并被 Web 服务器传送到浏览器上显示。每一个页面都被分配一个 URL (Uniform Reference Locator)，以便唯一地识别页面。URL 的格式为 `protocol://name-of-server/path-and-or-document-name`，其中 `protocol` 为协议，除了 HTTP 外，还有其他的在 URL 中有效的协议，见表 1.1。`name-of-server` 为 Web 服务器名，`path-and-or-document-name` 为路径或文档名称。如果 Web 服务器没有为 Web 页使用标准端口 80，URL 也可以包含端口号，即采用这样的格式：

`protocol://name-of-server:port-number/path-and-or-document-name`