



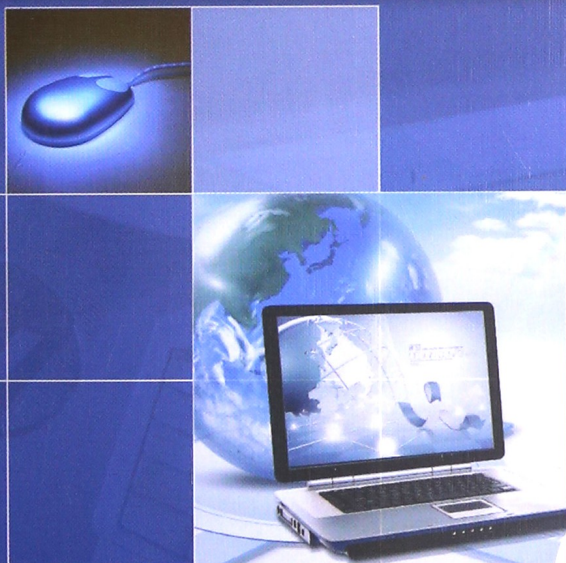
21st CENTURY
实用规划教材

21世纪全国应用型本科计算机案例型规划教材



JSP设计与开发案例教程

杨田宏 刘海学 那 勇 编 著



- 内容新颖，展示主流的Java Web 应用技术
- 案例完整，从前期分析设计到后期代码实现
- 实用性强，用实际项目来讲解具体案例开发



北京大学出版社
PEKING UNIVERSITY PRESS

014030222

TP312JA-43
305

21 世纪全国应用型本科计算机案例型规划教材

JSP 设计与开发案例教程

杨田宏 刘海学 那 勇 编 著



TP312JA-43
305



北京大学出版社
PEKING UNIVERSITY PRESS



北航

C1717161

内 容 简 介

本书基于 Sun 公司发布的 Java EE 6 SDK Update 3 开发工具包和 GlassFish Server 3.1.1 服务器,以丰富的示例为引导,图文并茂地展示了主流 Java Web 应用技术——JSP 的设计与开发。全书共有 11 章,分别从 Web 开发基础、JSP 基本语法、JSP 动作元素、JSP 内置对象、JSP 结合 JavaScript、JDBC 与数据库操作、JSP 与 JavaBean、JSP 的文件操作、Servlet 技术、JSP 的 XML 操作、Struts 应用基础等不同角度依次展示 JSP 中最基础、也是最重要的核心特性。本书力求示例的“学与做”,突出对 JSP 核心特性的设计和应用。通过对 JSP 这些特性的深入了解,将有助于读者对 Web 编程思想和编程技术的掌握和提高。

本书可作为高等院校计算机相关专业的教材,也可供 JSP 爱好者自学使用,以及供 JSP 程序员参考使用。

图书在版编目(CIP)数据

JSP 设计与开发案例教程/杨田宏,刘海学,那勇编著. —北京:北京大学出版社,2014.3

(21 世纪全国应用型本科计算机案例型规划教材)

ISBN 978-7-301-23734-2

I. ①J… II. ①杨…②刘…③那… III. ①JAVA 语言—网页制作工具—高等学校—教材 IV. ①TP312②TP393.092

中国版本图书馆 CIP 数据核字(2014)第 011488 号

书 名: JSP 设计与开发案例教程

著作责任者: 杨田宏 刘海学 那 勇 编著

策划编辑: 郑 双

责任编辑: 郑 双

标准书号: ISBN 978-7-301-23734-2/TP • 1321

出版发行: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> 新浪官方微博: @北京大学出版社

电子信箱: pup_6@163.com

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

印 刷 者: 三河市博文印刷厂

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 16 印张 362 千字

2014 年 3 月第 1 版 2014 年 3 月第 1 次印刷

定 价: 32.00 元

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

举报电话: 010-62752024 电子信箱: fd@pup.pku.edu.cn

21 世纪全国应用型本科计算机案例型规划教材

专家编审委员会

(按姓名拼音顺序)

主 任 刘瑞挺

副主任 陈 钟 蒋宗礼

委 员 陈代武 房爱莲 胡巧多 黄贤英

江 红 李 建 娄国焕 马秀峰

祁亨年 汪新民 王联合国 谢安俊

解 凯 徐 苏 徐亚平 宣兆成

姚喜妍 于永彦 张荣梅

信息技术的案例型教材建设

(代丛书序)

刘瑞挺

北京大学出版社第六事业部在 2005 年组织编写了《21 世纪全国应用型本科计算机系列实用规划教材》，至今已出版了 50 多种。这些教材出版后，在全国高校引起热烈反响，可谓初战告捷。这使北京大学出版社的计算机教材市场规模迅速扩大，编辑队伍茁壮成长，经济效益明显增强，与各类高校师生的关系更加密切。

2008 年 1 月北京大学出版社第六事业部在北京召开了“21 世纪全国应用型本科计算机案例型教材建设和教学研讨会”。这次会议为编写案例型教材做了深入的探讨和具体的部署，制定了详细的编写目的、丛书特色、内容要求和风格规范。在内容上强调面向应用、能力驱动、精选案例、严把质量；在风格上力求文字精练、脉络清晰、图表明快、版式新颖。这次会议吹响了提高教材质量第二战役的进军号。

案例型教材真能提高教学的质量吗？

是的。著名法国哲学家、数学家勒内·笛卡儿(Rene Descartes, 1596—1650)说得好：“由一个例子的考察，我们可以抽出一条规律。(From the consideration of an example we can form a rule.)”事实上，他发明的直角坐标系，正是通过生活实例而得到的灵感。据说是在 1619 年夏天，笛卡儿因病住进医院。中午他躺在病床上，苦苦思索一个数学问题时，忽然看到天花板上有一只苍蝇飞来飞去。当时天花板是用木条做成正方形的格子。笛卡儿发现，要说出这只苍蝇在天花板上的位置，只需说出苍蝇在天花板上的第几行和第几列。当苍蝇落在第四行、第五列的那个正方形时，可以用(4, 5)来表示这个位置……由此他联想到可用类似的办法来描述一个点在平面上的位置。他高兴地跳下床，喊着“我找到了，找到了”，然而不小心把国际象棋撒了一地。当他的目光落到棋盘上时，又兴奋地一拍大腿：“对，对，就是这个图”。笛卡儿锲而不舍的毅力，苦思冥想的钻研，使他开创了解析几何的新纪元。千百年来，代数与几何，井水不犯河水。17 世纪后，数学突飞猛进的发展，在很大程度上归功于笛卡儿坐标系和解析几何学的创立。

这个故事，听起来与阿基米德在浴缸洗澡而发现浮力原理，牛顿在苹果树下遇到苹果落到头上而发现万有引力定律，确有异曲同工之妙。这就证明，一个好的例子往往能激发灵感，由特殊到一般，联想出普遍的规律，即所谓的“一叶知秋”、“见微知著”的意思。

回顾计算机发明的历史，每一台机器、每一颗芯片、每一种操作系统、每一类编程语言、每一个算法、每一套软件、每一款外部设备，无不像闪光的珍珠串在一起。每个案例都闪烁着智慧的火花，是创新思想不竭的源泉。在计算机科学技术领域，这样的案例就像大海岸边的贝壳，俯拾皆是。

事实上，案例研究(Case Study)是现代科学广泛使用的一种方法。Case 包含的意义很广：包括 Example 例子，Instance 事例、示例，Actual State 实际状况，Circumstance 情况、事件、境遇，甚至 Project 项目、工程等。

我们知道在计算机的科学术语中，很多是直接来自日常生活的。例如 Computer 一词早在 1646 年就出现于古代英文字典中，但当时它的意义不是“计算机”而是“计算工人”，即专门从事简单计算的工人。同理，Printer 当时也是“印刷工人”而不是“打印机”。正是

由于这些“计算工人”和“印刷工人”常出现计算错误和印刷错误，才激发查尔斯·巴贝奇(Charles Babbage, 1791—1871)设计了差分机和分析机，这是最早的专用计算机和通用计算机。这位英国剑桥大学数学教授、机械设计专家、经济学家和哲学家是国际公认的“计算机之父”。

20 世纪 40 年代，人们还用 Calculator 表示计算机。到电子计算机出现后，才用 Computer 表示计算机。此外，硬件(Hardware)和软件(Software)来自销售人员。总线(Bus)就是公共汽车或大巴，故障和排除故障源自格瑞斯·霍普(Grace Hopper, 1906—1992)发现的“飞蛾子”(Bug)和“抓蛾子”或“抓虫子”(Debug)。其他如鼠标、菜单……不胜枚举。至于哲学家进餐问题，理发师睡觉问题更是操作系统文化中脍炙人口的经典。

以计算机为核心的信息技术，从一开始就与应用紧密结合。例如，ENIAC 用于弹道曲线的计算，ARPANET 用于资源共享以及核战争时的可靠通信。即使是非常抽象的图灵机模型，也受益于二战时图灵博士破译纳粹密码工作的关系。

在信息技术中，既有许多成功的案例，也有不少失败的案例；既有先成功而后失败的案例，也有先失败而后成功的案例。好好研究它们的成功经验和失败教训，对于编写案例型教材有重要的意义。

我国正在实现中华民族的伟大复兴，教育是民族振兴的基石。改革开放 30 年来，我国高等教育在数量上、规模上已有相当的发展。当前的重要任务是提高培养人才的质量，必须从学科知识的灌输转变为素质与能力的培养。应当指出，大学课堂在高新技术的武装下，利用 PPT 进行的“高速灌输”、“翻页宣科”有愈演愈烈的趋势，我们不能容忍用“技术”绑架教学，而是让教学工作乘信息技术的东风自由地飞翔。

本系列教材的编写，以学生就业所需的专业知识和操作技能为着眼点，在适度的基础知识与理论体系覆盖下，突出应用型、技能型教学的实用性和可操作性，强化案例教学。本套教材将会有机融入大量最新的示例、实例以及操作性较强的案例，力求提高教材的趣味性和实用性，打破传统教材自身知识框架的封闭性，强化实际操作训练，使本系列教材做到“教师易教，学生乐学，技能实用”。有了广阔的应用背景，再造计算机案例型教材就有了基础。

我相信北京大学出版社在全国各地高校教师的积极支持下，精心设计，严格把关，一定能够建设出一批符合计算机应用型人才培养模式的、以案例型为创新点和兴奋点的精品教材，并且通过一体化设计、实现多种媒体有机结合的立体化教材，为各门计算机课程配齐电子教案、学习指导、习题解答、课程设计等辅导资料。让我们用锲而不舍的毅力，勤奋好学的钻研，向着共同的目标努力吧！

刘瑞挺教授 本系列教材编写指导委员会主任、全国高等院校计算机基础教育研究会副会长、中国计算机学会普及工作委员会顾问、教育部考试中心全国计算机应用技术证书考试委员会副主任、全国计算机等级考试顾问。曾任教育部理科计算机科学教学指导委员会委员、中国计算机学会教育培训委员会副主任。PC Magazine《个人电脑》总编辑、CHIP《新电脑》总顾问、清华大学《计算机教育》总策划。

前 言

计算机编程技术从出现至今，已经历了相当长的发展历程。作为计算机教育工作者，我们越来越体会到：程序设计中最重要的不是流水化的代码编制，而是程序设计者本身。如果程序设计者本身的编程素养不高，那么最多只能是“生产”代码的工具而已。《JSP 设计与开发案例教程》一书的创作灵感和设计组织就是基于这样的思想：不停留在 JSP 技术复杂的理论机制和语言规范，而尝试揭示 JSP 编程技术最基本的“应用”特性，作为一种将读者从繁琐编码中解放出来的尝试，使更多 JSP 技术爱好者成为具有独立“设计”理念的程序工程人员。

为了能按这种方式来更好地理解本书内容，首先读者必须掌握 JSP 与编程的一些基本概念。本书讨论了 JSP 编程技术最基本的一些“应用”特性及 JSP 用以设计实现它们的方法。我们对每一章的安排都建立在如何较好地展示 JSP 编程技术在某种特定“应用”方面的设计实现基础上，希望以这种方式，逐步引导读者一步一步地进入 JSP 的世界，并最终领会 JSP 的神奇所在。

但需要特别注意的是，我们所提供的每一个“设计”思路和方法都不是通向 JSP 编程世界及解决问题的唯一途径，读者需要借助自身现有的知识体系加深对本书“设计”思想的理解。对 JSP 的一系列特性集合进行整体综合的考虑、设计，而非仅仅简单地编码、移用，从而使读者真正体会到 JSP 的强大。

同时，本书设计了一系列规模适中的学习模块，以保证读者可以在一个合理的时间内完成学习。因此，本书在内容安排设计时认真考虑了人们学习 JSP 的方式。在以往的教学过程中，我们编制了大量教学案例，经过反复地试验和修订，从中汲取集成了教学经验中的经验总结。例如，在以往教学中，学习者的反映有效地帮助我们认识到哪些知识点是比较困难的，需特别加以留意；哪些知识点容易引起混淆，需要加深解释；等等。其中很多都是由一系列离散的小问题组成的。因此，在本书中，我们进行了尝试和努力，试图使学习者在学习过程中尽可能少地出现这些问题。下面是本书将努力实现的目标：

(1) 对书中的每章精选示例进行讲解，用以较好地展示 JSP 某项特性的“应用”。

(2) 为了更好地激发学习者的学习兴趣，使他们能充分理解知识讲解的每个细节，本书所采用的示例尽可能简短。在这个环节上，为了追求在“有限时间”内的学习效果，我们遵循了简化的原则，而选择性地牺牲了部分示例的完整性。

(3) 示例讲解通俗，步骤详细，全书始终贯穿着“跟我做”的思想，加上丰富的插图和提示说明，让每个知识点都一目了然。

(4) 要揭示的 JSP 的“应用”特性，尽量贴近学习者的思想历程和学习习惯，据此精心编排顺序讲解，同时，每次都把学习内容向前推进一小步，便于读者对内容的积累、消化、吸收和进一步利用。当然，读者也可以根据自己实际的知识掌握情况，有选择地进行跳跃式学习。

(5) 每章都有明确的学习目标和学习重点，并围绕学习目标展开相关的学习内容布置

和安排,这样做不仅能让读者的学习目的更明确,也能让学习内容更易于理解,使其对学习产生更大的信心。

(6) 为使读者巩固对每章新知识的掌握,我们在每章结束时都安排了一系列类型丰富的练习,大多数练习比较简单,有助于检测和加深读者对知识的掌握;但也有些题目比较具有挑战性,这是为那些有兴趣深入研究的读者准备的。

(7) 本书配备了所有示例的源代码,以方便读者使用。所有源代码都作为保留版权的免费软件提供,为保证大家能够更便利地获得这些源代码,我们将在作者个人主页及其他正式源代码下载处定期发布本书源代码的最新版本,读者在尊重本书作者版权的情况下,可在课堂及其他场所引用这些代码。

(8) 本书编者的 E-mail 是 liuhaixue996@sohu.com,如果读者在学习过程中遇到困难或对本书有任何建议,可以给我们发送电子邮件,我们会及时回复,所有读者的反馈都将帮助我们修改及重新调整学习内容的重心,直到我们最后认为它成为一个较为完善的教学载体为止。

本书所演示的所有示例都基于 Sun 公司发布的最新的 Java EE 6 SDK Update 3 开发工具包和 GlassFish Server 3.1.1 服务器,至本书完稿为止,尚未推出更新版本。包括在本书中涉及的一些其他工具包、插件及类库等,我们都特别指出了所采用的版本及下载地址和安装部署方式,为了避免由于版本号原因导致程序代码运行错误,请读者选择安装相应的版本号软件运行程序。关于不同环境下程序代码运行的可靠性测试,以及对后续更新版本软件的支持,编者将会在以后的工作中不断地加以补充完善。

我们理解错误可能破坏读书的乐趣,并浪费大量的时间,因此,我们要尽量减少书中的错误。但无论作者花多大精力来避免,错误总是从意想不到的地方冒出来。如果您认为自己发现了一个错误,请为我们及时指出。我们很乐意倾听您对本书的意见,或者您也可以告诉我们要如何改进,您可以直接向 liuhaixue996@sohu.com 发送电子邮件。我们非常重视您的意见并尽最大努力改进我们的工作。当然,我们会尽力回答关于本书内容的所有问题,但是我们却无法回答您从书中代码衍生出的代码程序中的错误。

长期以来,我们一直与华东师范大学应用技术研究所、同济大学软件学院、上海财经大学信息管理与工程学院、上海应用技术学院机器嗅觉实验室、上海电力学院计算机科学工程系、上海创件信息科技有限公司、HP 中国软件研发中心、濮阳教育委员会、支付宝(中国)网络技术有限公司等科研院所、企事业单位保持着密切的合作伙伴关系。非常感谢他们能与我们一同交流探讨,并无私地分享各自的科研成果、思想理念、技术方案等宝贵的知识财富,正是有了这些才源源不断地聚集形成了本书的思想源泉和支撑根基,才能使我们在整个创作过程中都能专注于核心环节的编写,并顺利地完成本书的设计。

在这里我们要感谢顾君忠教授、贺梁教授、刘云翔教授、曹奇英教授、冯佳昕教授、胡建人副教授、吕钊副教授、杨静副教授、吴君老师等,他们给予的一些意见和支持发挥了非常关键的作用,并帮助我们澄清、纠正了专业知识方面的一些概念;感谢孟玲玲博士、崔修涛博士、杨燕博士、薛梅博士、倪琳博士、王明佳博士、李江峰博士等,他们与我们一同分享研究的经验,和我们花费了数月的时间将教学内容合并到一起,并探讨如何使学生感受到最完美的学习体验;感谢刘璇高级工程师、李吉存工程师、翁群工程师、费静婷工程师、上海应用技术学院机器嗅觉实验室、华东师范大学 MMIT 实验室为本书技术的实

现提供的大力支持；感谢我们教授过的那些学生，他们也是我们的老师，他们提出的问题使我们的教学内容愈发成熟起来……曾向我们提供过支持的朋友当然远远不止这些，由于篇幅所限，不能一一列举，我们真心感激所有这些朋友的支持。

这虽然不是我们第一次出书，但对这本书的出版异常重视。感谢北京大学出版社的郑双老师，是他为我们的工作清除了所有可能遇到的障碍，使我们感受了一次愉快的出书经历，同时，也要感谢编辑部的其他老师，正是他们强大的信心与毅力，才使我们最终梦想成真。

由于时间仓促和编者水平有限，对书中有些问题的研究不够，内容难免有偏失不当之处，还望读者多批评指教，谢谢大家！

编 者

2013 年 6 月

目 录

第 1 章 Web 开发基础	1	3.1.1 JSP 动作元素的组成及作用	48
1.1 相关理论知识	2	3.1.2 JavaBean 组件技术	51
1.1.1 Web 应用系统基本原理	2	3.2 相关实践知识	53
1.1.2 Java Web 应用目录结构	3	3.2.1 实现不同 JSP 页面间的跳转 ...	53
1.1.3 Java Web 应用开发流程	4	3.2.2 用 JavaBean 实现用户信息 注册	56
1.2 相关实践知识	5	3.3 实验安排	60
1.2.1 安装 Java EE 6 SDK Update 3 开发工具包	5	3.4 相关知识总结与拓展	60
1.2.2 Java 运行环境配置	9	3.4.1 知识网络拓展	60
1.2.3 运行和管理 GlassFish Server 3.1.1 服务器	11	3.4.2 其他知识补充	62
1.3 实验安排	14	习题	63
1.4 相关知识总结与拓展	14	第 4 章 JSP 内置对象	65
1.4.1 知识网络拓展	14	4.1 相关理论知识	66
1.4.2 其他知识补充	15	4.1.1 JSP 内置对象的组成	66
习题	15	4.1.2 JSP 内置对象的方法	67
第 2 章 JSP 基本语法	17	4.2 相关实践知识	70
2.1 相关理论知识	18	4.2.1 实现网站计数器功能	70
2.1.1 Java 语言的基本组成	18	4.2.2 实现错误异常的捕获和 处理	73
2.1.2 JSP 的执行过程	26	4.3 实验安排	79
2.1.3 JSP 页面的组成元素	28	4.4 相关知识总结与拓展	79
2.2 相关实践知识	32	4.4.1 知识网络拓展	79
2.2.1 编写第一个 JSP 页面	32	4.4.2 其他知识补充	81
2.2.2 使用 Eclipse Java EE IDE 创建项目	36	习题	81
2.3 实验安排	40	第 5 章 JSP 结合 JavaScript	84
2.4 相关知识总结与拓展	40	5.1 相关理论知识	85
2.4.1 知识网络拓展	40	5.1.1 客户端编程原理及使用	85
2.4.2 其他知识补充	44	5.1.2 JavaScript 基础编程技术	86
习题	45	5.2 相关实践知识	92
第 3 章 JSP 动作元素	47	5.2.1 客户端信息验证	92
3.1 相关理论知识	48	5.2.2 客户端 JavaScript 和 服务器端 JSP 的数据交互	99

5.3 实验安排	102	8.4.2 其他知识补充	165
5.4 相关知识总结与拓展	102	习题	165
5.4.1 知识网络拓展	102	第 9 章 Servlet 技术	168
5.4.2 其他知识补充	105	9.1 相关理论知识	169
习题	105	9.1.1 Servlet 基础	169
第 6 章 JDBC 与数据库操作	107	9.1.2 Servlet 的结构与配置	170
6.1 相关理论知识	108	9.1.3 Servlet 在 JSP 中的应用	172
6.1.1 JDBC 基础	108	9.2 相关实践知识	172
6.1.2 SQL	109	9.3 实验安排	180
6.1.3 JDBC 编程	112	9.4 相关知识总结与拓展	181
6.2 相关实践知识	114	9.4.1 知识网络拓展	181
6.3 实验安排	129	9.4.2 其他知识补充	184
6.4 相关知识总结与拓展	129	习题	185
6.4.1 知识网络拓展	129	第 10 章 JSP 的 XML 操作	187
6.4.2 其他知识补充	132	10.1 相关理论知识	188
习题	132	10.1.1 XML 基础语法	188
第 7 章 JSP 与 JavaBean	134	10.1.2 Java 语言 XML 处理 API	191
7.1 相关理论知识	135	10.1.3 JSP 的 XML 操作分类	194
7.1.1 JavaBean 的设计	135	10.2 相关实践知识	198
7.1.2 JSP 中 JavaBean 的使用	137	10.3 实验安排	209
7.2 相关实践知识	138	10.4 相关知识总结与拓展	209
7.2.1 JavaBean 作用范围测试	138	10.4.1 知识网络拓展	209
7.2.2 使用 JavaBean 访问数据库	142	10.4.2 其他知识补充	210
7.3 实验安排	145	习题	210
7.4 相关知识总结与拓展	146	第 11 章 Struts 应用基础	213
7.4.1 知识网络拓展	146	11.1 相关理论知识	214
7.4.2 其他知识补充	146	11.1.1 Struts 应用框架介绍	214
习题	146	11.1.2 Struts 2 的配置与应用	215
第 8 章 JSP 的文件操作	149	11.1.3 Struts 2 的标签库	221
8.1 相关理论知识	150	11.2 相关实践知识	222
8.1.1 文件和目录的基本操作	150	11.3 实验安排	230
8.1.2 Java 中文件处理的相关类	151	11.4 相关知识总结与拓展	230
8.2 相关实践知识	156	11.4.1 知识网络拓展	230
8.3 实验安排	164	11.4.2 其他知识补充	236
8.4 相关知识总结与拓展	164	习题	237
8.4.1 知识网络拓展	164	参考文献	239

第 1 章

Web 开发基础

教学目标

- (1) 了解 Web 应用系统基本原理、JSP 应用系统目录结构和 JSP 应用开发流程;
- (2) 熟悉 Java EE 6 SDK Update 3 开发工具包的下载、安装和使用;
- (3) 掌握运行和管理 GlassFish Server 3.1.1 服务器;
- (4) 熟悉 Java 运行环境参数的设置和使用, 了解常用 Java 运行环境命令的使用。

教学任务

- (1) 学习 JSP 技术之前需要掌握的基础知识;
- (2) 安装 Java EE 6 SDK Update 3 开发工具包;
- (3) 配置 Java 运行环境;
- (4) 运行、管理 GlassFish Server 3.1.1 服务器, 测试示例程序的部署和演示。

1.1 相关理论知识

1.1.1 Web 应用系统基本原理

World Wide Web(万维网)技术作为 Internet 上信息资源共享的解决方案,是构建在浏览器/服务器(Browser/Server, B/S)模型及 HTTP 协议的基础上的。我们可以将以 B/S 结构为基础的 Web 应用程序的运作模式简单地描述为从请求到处理,再到响应(见图 1-1)。

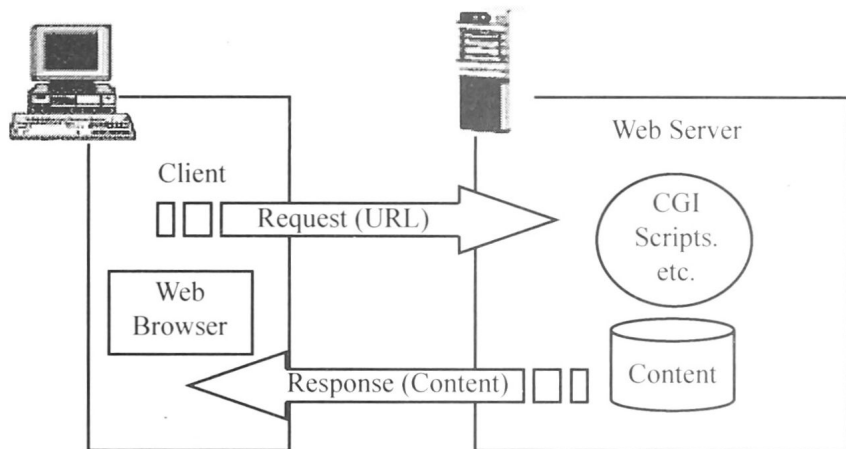


图 1-1 B/S 结构示意图

从图 1-1 中我们可以看到由浏览器提交的请求传给 Web 服务器,由服务器调用相应的网页应用程序进行处理,处理的结果——网页由 Web 服务器作为对请求的应答发送给浏览器。至于对提交的信息如何处理,就交由 Web 应用程序的开发人员编写相关的代码来决定。如果需要对数据库、文档等资源内容进行访问,开发人员还可以利用 Web 服务器所提供的接口对其进行操作。

了解了 B/S 结构,也就理解了 Web 应用程序的原理。常见的聊天室、BBS 论坛等,都是基于 B/S 结构的网页应用程序。我们再以 JSP 为例,了解一个基于 JSP 的 Web 应用系统页面交互的过程,大致过程描述如下。

(1) 浏览器发出对 Web 页面的请求。

(2) 浏览器利用 URL 辨别 Web 服务器的地址,找到自己需要的主页,并给出 Web 服务器需要的其他信息,这些信息包括浏览名称(Internet Explorer)、版本(8.0)、操作系统(Windows 7)及用来填写的表等。

(3) 如果请求的是 HTML 文件,Web 服务器就简单地找到该文件,然后传送该文件的内容到浏览器,浏览器得到内容后就开始将基于 HTML 的代码译成页面。

(4) 当请求的是 JSP 页面时,并不立即发送文件的内容。首先会执行脚本,由脚本产生一些 HTML 代码,然后将这些代码传送给 JSP 执行引擎,再由它将 HTML 文件传送给 Web 浏览器。JSP 的 Web 应用系统交互运行模式如图 1-2 所示。

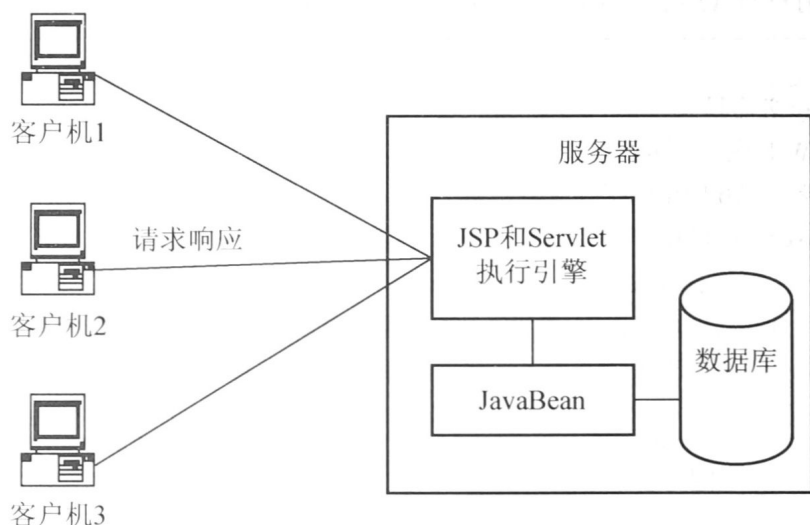


图 1-2 JSP 的 Web 应用系统交互运行模式

1.1.2 Java Web 应用目录结构

Sun 公司曾经对 Java Web 应用做出如下解释:“Java Web 应用由一组 Servlet/JSP、HTML 文件、相关 Java 类,以及其他可以被绑定的资源构成,它可以在由各种供应商提供的符合 Servlet 规范的容器中运行。”

从上述对 Java Web 应用的描述可以看出,Java Web 应用可以在多种符合 Servlet 规范的容器中运行。在 Java Web 应用中可以包含如下内容。

(1) Servlet: 标准 Servlet 接口的实现类,运行在服务器端,包含了被 Servlet 容器动态调用的程序代码。

(2) JSP: 包含 Java 程序代码的 HTML 文档,运行在服务器端。当客户端请求访问 JSP 文件时,Servlet 容器首先把它编译成 Servlet 类,然后动态调用它的程序代码。

(3) 相关的 Java 类: 开发人员自定义的与 Web 应用相关的 Java 类。

(4) 静态文档: 存放在服务器端的文件系统中。当客户端请求访问这些文件时,Servlet 容器首先从本地文件系统中读取这些文件的数据,再把它发送到客户端。

(5) 客户端类: 由客户端运行的类。Applet 是典型的客户端类,当客户端请求访问 Applet 时,Servlet 容器首先从本地文件系统中读取 Applet 的.class 文件中的数据,再把它发送到客户端,由客户端运行 Applet。

(6) web.xml 文件: Java Web 应用的配置文件,采用 XML 格式。该文件必须位于 Web 应用的 WEB-INF 子目录下。

为了能让 Servlet 容器顺利地找到 Java Web 应用中的各个组件,Servlet 规范规定,Java Web 应用必须采用固定的目录结构,即每种类型的组件在 Web 应用中都有固定的存放目录。Servlet 规范还规定,Java Web 应用的配置信息存放在 WEB-INF/web.xml 文件中,Servlet 容器从该文件中读取配置信息。在发布某些 Web 组件(如 Servlet)时,需要在 web.xml 文件中添加相应的关于这些 Web 组件的配置信息。

Java Web 应用程序必须使用的规范目录结构如图 1-3 所示。

应用程序根目录

-- WEB-INF 目录：必需目录

-- web.xml：Web 应用部署描述文件

-- classes 目录：Web 应用的类文件存放处

-- lib 目录：Web 应用使用的第三方类库文件存放处

-- TLD 文件：标签库描述文件

-- 其他静态文件：HTML、CSS、JavaScript、图片等

-- *.jsp：JSP 文件，通常存放在 Web 应用程序的根目录下，有时为便于管理也可以存放在根目录下的其他目录下

图 1-3 Java Web 应用程序必须使用的规范目录

其中，WEB-INF 目录是不允许浏览客户看到该目录下的文件的，该目录下的文件供 Web 服务器专用，包含 Web 应用程序的部署描述文件(web.xml)和两个用于存储已编译 Java 类和库 JAR 文件的子目录。这些子目录的名称分别为 classes 和 lib。Java 类包括 Servlet、辅助程序类和预编译的 JSP(如果需要)。在运行时，Servlet 容器的类加载器先加载 classes 目录下的类，再加载 lib 目录下的 JAR 文件(Java 类库的打包文件)中的类。因此，如果两个目录下存在同名的类，classes 目录下的类具有优先权。而 web.xml 是 WEB-INF 目录下最重要的文件，它描述了程序的部署、配置信息，为 Web 服务器所使用。

属于 Web 应用程序的所有 Servlet、类、静态文件和其他资源都组织在目录层次中。

以 GlassFish Server 3.1.1 应用服务器为例，假定开发一个名为 DefaultWebApp 的 Java Web 应用，首先，应该创建这个 Web 应用的目录结构具体如下。

(1) DefaultWebApp\：将 HTML 和 JSP 等文件放入 Web 应用程序的文档根目录中。在 GlassFish Server 3.1.1 的默认安装中，此目录称为 DefaultWebApp，位于路径 glassfish3\glassfish\domains\domain1\applications 下。

(2) DefaultWebApp\WEB-INF\web.xml：用于配置 Web 应用程序的 Web 应用程序部署描述文件。

(3) DefaultWebApp\WEB-INF\classes：包含服务器端类，如 HTTP、Servlet 和实用工具类。

(4) DefaultWebApp\WEB-INF\lib：包含 Web 应用程序使用的 JAR 等第三方类库文件。

JSP 页面和 Servlet 可以访问 GlassFish Server 中可用的所有服务和应用程序编程接口(Application Programming Interface, API)。这些服务包括 EJB、Java 数据库连接(Java Data Base Connectivity, JDBC)、Java 消息服务(Java Messaging Service, JMS)、XML 等。

1.1.3 Java Web 应用开发流程

这里，我们介绍一种 Java Web 应用开发的主要流程，读者可参照该流程和自身的实际经验完成 Java Web 应用开发设计(见图 1-4)。

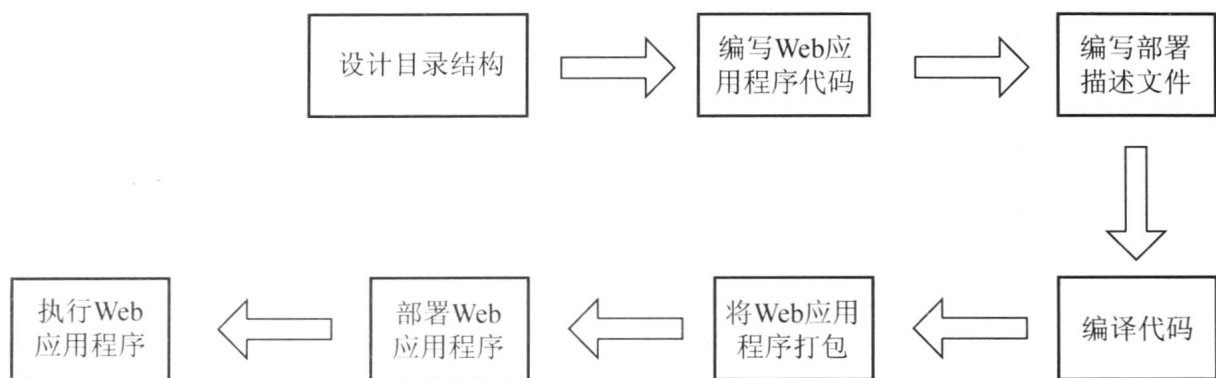


图 1-4 Java Web 应用开发流程

(1) 设计目录结构：根据具体业务需要，遵照规范的目录结构设计好 Web 应用程序的目录结构。

(2) 编写 Web 应用程序代码：编写业务逻辑所需的 Java 代码。

(3) 编写部署描述文件：把 Servlet、初始化参数等定义到部署描述文件 web.xml 中。

(4) 编译代码：把编写好的 Java 源代码编译成字节码。

(5) 将 Web 应用程序打包：把整个 Web 应用程序打成 WAR 包，以方便部署。

(6) 部署 Web 应用程序：把打好的 WAR 包部署到 Web 服务器上。

(7) 执行 Web 应用程序：启动 Web 服务器，利用客户端浏览器进行访问测试。

提示!!!

在实际的开发过程中，一般会使用各种集成开发环境(Integrated Development Environment, IDE)工具，使用 IDE 工具进行 Web 应用程序开发时，只需要开发人员完成前三个步骤，其他步骤 IDE 工具可以自动完成。

1.2 相关实践知识

1.2.1 安装 Java EE 6 SDK Update 3 开发工具包

(1) 双击可执行文件 java_ee_sdk-6u3-jdk7-windows.exe，打开 Java EE 6 SDK 的安装初始界面(见图 1-5)。

该步骤中的文件 java_ee_sdk-6u3-jdk7-windows.exe 即为 Java EE 6 SDK Update 3 开发工具包。Java 开发工具包版本的选择可根据不同的实际开发应用需要，自行从 Oracle 官方网站下载(可参照 1.4.1 节中的“Java EE 6 SDK Update 3 及其他版本 Java 开发工具包下载地址”)。

(2) 单击【下一步】按钮，选择安装类型(见图 1-6)。



图 1-5 Java EE 6 SDK 的安装初始界面



图 1-6 选择安装类型界面

Java EE 6 SDK Update 3 提供了【典型安装】和【自定义安装】两种方式，选择默认的【典型安装】方式即可。