



面向“十二五”高职高专规划教材·计算机系列

JSP编程及案例分析

■ 郑 宁 主 编
■ 周梦熊 谢红标 副主编

清华大学出版社 · 北京交通大学出版社

面向“十二五”高职高专规划教材·计算机系列

JSP 编程及案例分析

主 编 郑 宁

副主编 周梦熊 谢红标

清华大学出版社

北京交通大学出版社

·北京·

内 容 简 介

本书较为全面地介绍了 JSP 技术, 根据作者的开发经验, 按照基础篇、提高篇、数据库应用篇的次序, 由浅入深、循序渐进地介绍了 JSP 的运行系统、基本概念、语法规范及其相关内容, 并提供了大量的应用实例。

全书共分为 6 章, 内容包括 JSP 概述和基础、JSP 内置对象、JavaBean 组件、Servlet 技术、数据库操作, 并结合应用实例加以巩固。

本书适合作为高职高专计算机专业的教材, 也可供对 Web 技术感兴趣的读者自学参考。

本书电子教案及相关配套资料可到北京交通大学出版社网站下载, 网址为 <http://press.bjtu.edu.cn>。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

JSP 编程及案例分析/郑宁主编. —北京: 清华大学出版社; 北京交通大学出版社, 2010. 5
(面向“十二五”高职高专规划教材·计算机系列)

ISBN 978-7-5121-0111-1

I. ① J… II. ① 郑… III. ① JAVA 语言-主页制作-程序设计-高等学校: 技术学校-教材 IV. ① TP393.092

中国版本图书馆 CIP 数据核字(2010)第 083315 号

责任编辑: 郭东青

出版发行: 清华大学出版社 邮编: 100084 电话: 010-62776969 <http://www.tup.com.cn>
北京交通大学出版社 邮编: 100044 电话: 010-51686414 <http://press.bjtu.edu.cn>

印刷者: 北京东光印刷厂

经 销: 全国新华书店

开 本: 185×260 印张: 20.5 字数: 512 千字

版 次: 2010 年 5 月第 1 版 2010 年 5 月第 1 次印刷

书 号: ISBN 978-7-5121-0111-1/TP·592

印 数: 1~4 000 册 定价: 30.00 元

本书如有质量问题, 请向北京交通大学出版社质监组反映。对您的意见和批评, 我们表示欢迎和感谢。

投诉电话: 010-51686043, 51686008; 传真: 010-62225406; E-mail: press@bjtu.edu.cn。

前 言

随着信息技术的发展和 Internet 的普及，人们越来越依赖于互联网给生活带来的便利，纷纷利用网络技术来构建自己的站点。可以说，WWW 技术是促进 Internet 高速发展的主要因素之一。

早期的网页基本上是静态网页，主要特点是“只能层现，不能交互”，当然这在互联网起步阶段起到了一定作用，但随着网络内容的膨胀，人们越来越不能满足现状，人与人之间的网络交流等都需要实时交互，而且这种需求越来越强烈。

鉴于此，各种各样的 Web 技术应运而生，且都是建立在一系列“活跃”的交互操作上的。通常人们用客户 - 服务器这个词来描述 Web。这是一个交互的概念，一般把提出请求的一方称为客户端，而把响应请求的一方称为服务器端。这种简单的模型是静态的，它只能对对方的激励作出响应。而在活跃的 Web 中，双方都应该是活跃的，只有这样才能把客户机和服务器结合起来，产生最强的交互，这样就引出了动态网页的概念。

在 Web 领域里，有几十万乃至上百万的站点相互之间正在进行着激烈的竞争，它们想尽一切办法来吸引用户的注意力。简单的、静态的页面对用户不会有太大的吸引力，只有动态的、有条理的数据加上友好的、交互性强的界面，再加上丰富的内容，才能吸引用户。当然，数据的自动更新也非常重要。在短短的几年时间里，Web 的面貌已经发生了非常大的变化。今天，可以在 Web 页面中创建应用程序、访问数据库，无论在感觉上、操作中，还是实际的用途方面都与 Windows 中的应用程序非常类似。现在，商家可以与潜在客户、现有的客户、员工及其他人中的任何一个进行网络在线实时沟通，并实施一些在线的服务类的商业活动。

为了抢占 Internet 这个诱人的市场，在 Sun 公司的倡导下，许多公司共同参与建立了一种新的动态网页技术标准——Java Server Pages (JSP)。Sun 应用 Java 社团开发 JSP。在开发 JSP 规范的过程中，Sun 公司与许许多多主要的 Web 服务器、Web 应用服务器和开发工具供应商，以及各种各样富有经验的开发团体进行合作，结果找到了一种适合于应用和页面开发人员的开发方法，它具有极佳的可移植性和易用性。Sun 将针对 JSP 的产品授权给了工具提供商（如 Macromedia）、结盟公司（如 Apache, Netscape）、最终用户、协作商及其他人。

最近，Sun 将最新版本的 JSP 和 Java Servlet (JSP 1.1, Java Servlet 2.2) 的源代码发放给了 Apache，以求 JSP 与 Apache 紧密结合、共同发展。Apache、Sun 和许多其他的公司及个人公开成立了一个咨询机构，以便任何公司和个人都能免费取得信息。这样，Sun 公司就在这个领域中稳稳地站住了脚跟。

JSP 技术作为一种动态 Web 开发技术，为非专业人员开发高水平的网站提供了良好的方法。但是，应该如何来学习呢？虽然，在市场上关于这方面的书籍有不少，但基本上不是纯

粹地介绍理论知识，就是纯粹地介绍实例，将理论与实例结合的书籍很少，特别是针对工程应用型人才的培养教材少之又少。

在本书中，笔者总结了以往教学的经验，加入了不少的实例，弥补了相关教材的不足。

本书既适合作高职高专计算机专业的教材，也适合各层次对 Web 开发感兴趣的读者使用。

编 者

2010 年 4 月

目 录

第1篇 基础篇

第1章 JSP 概述	2
1.1 JSP 简介	2
1.1.1 什么是 JSP	2
1.1.2 与 Java Servlet 的关系	2
1.1.3 JSP 技术特点	3
1.1.4 与其他技术比较	4
1.2 图解开发环境	6
1.2.1 JDK 安装及配置	6
1.2.2 Tomcat 安装及配置	10
1.2.3 建立虚拟工作目录	12
1.2.4 MyEclipse 安装及配置	13
1.3 语法介绍	20
1.3.1 JSP 页面中的元素	20
1.3.2 JSP 语法概要	20
1.3.3 模板文本(静态 HTML)	22
1.4 运行第一个 JSP 程序	23
1.5 入门实例	25
1.5.1 实例 1: HTML 练习	25
1.5.2 实例 2: 显示日期时间	32
1.6 本章小结	34
1.7 习题	34
第2章 JSP 基础	35
2.1 JSP 基本语法	35
2.1.1 JSP 声明	35
2.1.2 JSP 表达式	36
2.1.3 脚本段	37
2.2 JSP 的指令	38
2.2.1 page 指令	38
2.2.2 include 指令	39
2.2.3 taglib 指令	42
2.3 JSP 的动作	43
2.3.1 jsp:include 动作	44

2.3.2 jsp:useBean 动作	46
2.3.3 jsp:setProperty 动作	50
2.3.4 jsp:getProperty 动作	51
2.3.5 jsp:forward 动作	52
2.3.6 jsp:plugin 动作	54
2.3.7 jsp:param 动作	55
2.4 JSP 的注释	56
2.5 JSP 基础实例	58
2.5.1 实例 1: JSP 中方法 定义	58
2.5.2 实例 2: 计算任意阶乘	58
2.5.3 实例 3: JSP 的出错 处理	64
2.5.4 实例 4: 使用 forward 动作	66
2.5.5 实例 5: 使用 param 动作	67
2.5.6 实例 6: 使用 include 动作	67
2.6 本章小结	68
2.7 习题	68
第3章 JSP 内置对象	71
3.1 request 对象	71
3.1.1 request 对象常用方法	71
3.1.2 表单信息读取	72
3.1.3 request 乱码问题	74
3.2 response 对象	75
3.2.1 response 对象常用方法	76
3.2.2 动态响应 contentType 属性	76
3.2.3 response 的 HTTP 文件头	80
3.2.4 response 重定向	81
3.2.5 response 的状态行	81
3.3 session 对象	82
3.3.1 session 对象常用方法	82

3.3.2 session 对象的 ID	83	的使用	112
3.3.3 session 对象与 URL 重写 ..	83	3.9 本章小结	113
3.3.4 session 对象的使用	85	3.10 习题	114
3.4 application 对象	87		
3.5 out 对象	88	第 2 篇 提 高 篇	
3.5.1 out 对象常用方法	88	第 4 章 JavaBean 组件	116
3.5.2 out 对象之 print() 和 println() 对比	89	4.1 什么是 JavaBean	116
3.6 exception 对象	89	4.2 编写和使用 JavaBean	116
3.6.1 exception 对象常用方法	89	4.2.1 编写 JavaBean	116
3.6.2 exception 对象的使用	89	4.2.2 使用 JavaBean	117
3.6.3 运行中可能出现的问题 ..	90	4.3 获取和设置 JavaBean 属性	121
3.7 其他内置对象	90	4.3.1 获取 JavaBean 属性	121
3.8 JSP 内置对象实例	91	4.3.2 设置 JavaBean 属性	122
3.8.1 实例 1: 防刷新计数器	91	4.4 JavaBean 应用实例	126
3.8.2 实例 2: 用户注册信息	93	4.4.1 实例 1: 编写 JavaBean ..	126
3.8.3 实例 3: 获取请求相关 信息	95	4.4.2 实例 2: 使用 JavaBean ..	127
3.8.4 实例 4: 表单信息读取	95	4.4.3 实例 3: 获取 JavaBean 属性	129
3.8.5 实例 5: 提交表单计算 平方根	96	4.4.4 实例 4: 设置 JavaBean 属性	130
3.8.6 实例 6: 中文乱码问题	97	4.4.5 实例 5: 简单计算器	133
3.8.7 实例 7: 动态响应 contentType 属性	99	4.4.6 实例 6: 彩色验证码	137
3.8.8 实例 8: 定时刷新页面 ..	101	4.4.7 实例 7: 文本的读写	139
3.8.9 实例 9: response 重定向	102	4.4.8 实例 8: 猜数字游戏	141
3.8.10 实例 10: 修改 response 状态码	103	4.5 本章小结	144
3.8.11 实例 11: session 与 URL 重写	103	4.6 习题	144
3.8.12 实例 12: session 的 ID ..	105	第 5 章 Servlet 技术	145
3.8.13 实例 13: session 对象的 使用	107	5.1 Servlet 简介	145
3.8.14 实例 14: application 对象 的使用	109	5.1.1 什么是 Servlet	145
3.8.15 实例 15: out 对象的 print() 与 println() 对比	110	5.1.2 Servlet 基本结构	145
3.8.16 实例 16: out 对象的 使用	111	5.1.3 Servlet 的映射	146
3.8.17 实例 17: exception 对象		5.1.4 Servlet 生命周期	147
		5.2 初始化参数	148
		5.2.1 Servlet 初始化参数	148
		5.2.2 实例: 测试 Servlet 初始化 参数	149
		5.2.3 ServletContext 初始 化参数	151
		5.2.4 实例: 测试 ServletContext 初始化参数	151

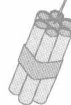
5.3 Servlet 请求分派	152
5.3.1 请求分派	152
5.3.2 重定向与请求分派	155
5.3.3 实例:重定向/请求 分派	156
5.4 理解会话	159
5.4.1 会话管理机制	160
5.4.2 会话对象中读取/保存 数据	160
5.5 Servlet 过滤器	161
5.5.1 过滤器工作原理	161
5.5.2 Servlet 过滤器 API	161
5.5.3 过滤器相关接口工作 流程	162
5.5.4 过滤器配置	163
5.5.5 过滤器链配置	165
5.5.6 实例 1:简单过滤器	166
5.5.7 实例 2:过滤器链	169
5.6 Servlet 监听器	172
5.6.1 接口及相关事件类	172
5.6.2 监听器配置	174
5.6.3 实例 1:上下文相关监听器 应用	174
5.6.4 实例 2:HttpSessionListener 应用	176
5.6.5 实例 3:HttpSessionAttribute- Listener 应用	179
5.6.6 实例 4:HttpSessionBinding Listener 应用	180
5.6.7 实例 5:请求相关监听器 应用	181
5.7 Servlet 上下文	183
5.8 MVC 迷你教程	184
5.8.1 MVC 模式介绍	184
5.8.2 实战 MVC	185
5.9 Servlet 实例	190
5.9.1 实例 1:手工完成 Servlet 程序	190
5.9.2 实例 2:用户登录验证	193
5.9.3 实例 3:Web 定时器	197

5.9.4 实例 4:调查反馈	200
5.9.5 实例 5:会话管理	204
5.9.6 实例 6:重定向与请求 分派	212
5.10 本章小结	214
5.11 习题	214

第 3 篇 数据库应用

第 6 章 JSP 数据库操作	216
6.1 数据库基础	216
6.1.1 关系数据库简介	216
6.1.2 Access 数据库	217
6.1.3 SQL Server 数据库	218
6.1.4 Oracle 数据库	218
6.1.5 MySQL 数据库	219
6.2 JDBC 简介	224
6.2.1 什么是 JDBC	224
6.2.2 JDBC 的用途	225
6.2.3 JDBC 的类型	225
6.2.4 JDBC 的入门	225
6.3 JSP 中使用数据库	229
6.3.1 数据库的连接过程	229
6.3.2 几种数据库的连接	229
6.4 数据操作	232
6.4.1 添加记录	234
6.4.2 查询记录	235
6.4.3 更新记录	244
6.4.4 删除记录	246
6.5 JSP 数据库应用实例	247
6.5.1 实例 1:留言板	247
6.5.2 实例 2:通讯录	261
6.6 本章小结	278
6.7 习题	278
附录 A 留言板系统	279
A1 数据存储	279
A2 Java 程序代码	280

第 1 篇 基 础 篇



第 1 章 JSP 概述

本章将介绍 JSP 的概念，用图解方式对 JSP 开发环境进行讲解，并对 JSP 语法进行简单介绍，通过一个 JSP 程序的执行来使读者对 JSP 有个大概了解，最后给出几个实例便于上机操作。通过本章的学习，可以使读者了解 JSP 的背景和开发设计步骤。

需重点掌握的内容：

- ◇ JSP 技术特点
- ◇ JSP 环境配置
- ◇ JSP 语法

1.1 JSP 简介

WWW 是目前 Internet 上最主要的信息服务类型，它深入影响生活中的方方面面，如政治、商业及教育等各个领域。HTML 语言是 WWW 的基础，而 JSP 是开发和维护 Web 站点的一种重要技术。在 Sun 正式发布 JSP 之后，它作为一种新的 Web 应用开发技术很快便引起了人们的关注，从此 JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境。

1.1.1 什么是 JSP

JSP 是 Java Server Pages（Java 服务器系统页面）的缩写，它是由 Sun 公司倡导的、由许多公司参与一起建立的一种动态网页技术标准，它在 HTML 语言的基础上使用 Java 语言对网页进行编程，JSP 技术能够实现普通静态 HTML 和动态 HTML 混合编码。

下面就是一个简单的 JSP 页面：

```
<HTML>
  <HEAD>
    <TITLE>第一个 JSP 页面</TITLE>
  </HEAD>
  <BODY>
    <H1>欢迎</H1>
    <!-- out.println 用来输出内容 -->
    <% out.println("hello world!"); %>
    <A HREF="http://itsoft.hdu.edu.cn">杭州电子科技大学软件工程学院
    </A>
  </BODY>
</HTML>
```

1.1.2 与 Java Servlet 的关系

Java Servlet 是 Java 语言的一部分，提供了用于服务器编程的 API，Java Servlet 编写的 Java 程序称为一个 Servlet。Servlet 通过 HTML 与客户交互信息。

Servlet 的最大缺点是不能有效地管理页面的逻辑部分和页面的输出部分，导致 Servlet

代码非常混乱，用 Servlet 来管理网站是一件很困难的事情。

为了克服 Servlet 的缺点，Sun 公司以 Java Servlet 为基础，推出了 JSP。JSP 具有 Servlet 的几乎所有的优点。当一个客户请求一个 JSP 页面时，JSP 引擎根据 JSP 页面生成一个 Java 文件，即一个 Servlet。用 JSP 支持 JavaBean 这一特点，可以有效地管理页面的逻辑部分和页面的输出部分。另外，JSP 也可以和 Servlet 有效地结合，分离页面的逻辑部分和页面的输出部分。

1.1.3 JSP 技术特点

关于 JSP 技术特点，主要有以下几点。

1. 将内容的生成和显示进行分离

使用 JSP 技术，Web 页面开发人员可以使用 HTML 或者 XML 标识来设计和格式化最终页面，使用 JSP 标识或者小脚本来生成页面上的动态内容。生成内容的逻辑被封装在标识和 JavaBean 组件中，并且捆绑在小脚本中，所有的脚本在服务器端运行。如果核心逻辑被封装在标识和 JavaBean 中，那么其他人（如 Web 管理人员和页面设计者）能够编辑和使用 JSP 页面，而不影响内容的生成。

在服务器端，JSP 引擎解释 JSP 标识和小脚本，生成所请求的内容（如通过访问 JavaBean 组件，使用 JDBC 技术访问数据库或者包含文件），并且将结果以 HTML（或者 XML）页面的形式发送回浏览器。这既有助于保护自己的代码，又可保证任何基于 HTML 的 Web 浏览器的完全可用性。

2. 一次编译，到处运行

由于 JSP 页面的内置脚本语言基于 Java 语言，而所有的 JSP 页面都要被编译成为 Servlet，这样 JSP 页面就具有 Java 技术的所有优点，包括健壮的存储管理和安全性等。当然其中最重要的一点就是“一次编译，到处运行”。

JSP 技术是与设计平台完全无关的，包括它的动态 Web 页面、Web 服务器和底层的服务器组件。你可以在任何平台上编写 JSP 页面，在任何 Web 服务器或者 Web 应用服务器上运行，或者通过任何 Web 浏览器访问。有了这个优点，随着越来越多的供应商将 JSP 支持添加到他们的产品中，你就可以使用自己所选择的服务器和工具了。更改工具或服务器并不会影响到当前的应用。

3. 强调可重用的组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件（JavaBean 或者企业版的 JavaBean 组件）来执行应用程序中所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件，或者使得这些组件为更多的使用者或者客户团体所使用。这些组件有助于将网页的设计与逻辑程序的编写分开，节约了开发时间，同时充分发挥了 Java 和其他脚本语言的跨平台的能力和灵活性。基于组件的方法加速了总体开发过程，并且使得各种组织在他们现有的技能和优化结果的开发努力中得到平衡。

4. 采用标记简化页面的开发

Web 页面开发人员不一定是熟悉脚本语言的编程人员。JSP 技术封装了许多功能，这些功能是在易用的、与 JSP 相关的 XML 标记中进行动态内容生成时所必需的。标准的 JSP 标记能够访问和实例化 JavaBean 组件，设置或者检索组件属性，下载

Applet, 以及执行用其他方法更难以编码和耗时的功能。通过开发定制化标识库, JSP 技术是可以扩展的。今后, 第三方开发人员和其他人员可以为常用功能创建自己的标识库。这使得 Web 页面开发人员能够使用熟悉的工具和如同标记一样的执行特定功能的构件来工作。

1.1.4 与其他技术比较

1. 简介

目前, 最常用的三种动态网页语言是 ASP (Active Server Pages)、JSP (Java Server Pages) 和 PHP。

ASP 是一种 Web 服务器端的开发环境, 利用它可以产生和执行动态的、互动的、高效的 Web 服务应用程序。ASP 采用脚本语言 (VBScript 等) 作为自己的开发语言。

PHP 是一种跨平台的服务器端的嵌入式脚本语言。它大量地借用 C、Java 和 Perl 语言的语法, 并耦合 PHP 自己的特性, 使 Web 开发者能够快速地写出动态页面。它支持目前绝大多数数据库。还有一点, PHP 是完全免费的, 你可以从 PHP 官方站点 (<http://www.php.net>) 自由下载。而且你可以不受限制地获得源码, 甚至可以从中加进你自己需要的特色。

JSP 是 Sun 公司推出的新一代网站开发语言, JSP 可以在 Servlet 和 JavaBean 的支持下, 完成功能强大的站点程序。

这三者都提供在 HTML 代码中混合某种程序代码、由语言引擎解释执行程序代码的能力。但 JSP 代码被编译成 Servlet 并由 Java 虚拟机解释执行, 这种编译操作仅在对 JSP 页面的第一次请求时发生。在 ASP、PHP 和 JSP 环境下, HTML 代码主要负责描述信息的显示样式, 而程序代码则用来描述处理逻辑。普通的 HTML 页面只依赖于 Web 服务器, 而 ASP、PHP、JSP 页面需要附加的语言引擎分析和执行程序代码。程序代码的执行结果被重新嵌入到 HTML 代码中, 然后一起发送给浏览器。ASP、PHP、JSP 三者都是面向 Web 服务器的技术, 客户端浏览器不需要任何附加的软件支持。

2. 技术特点

(1) ASP 的技术特点。

- ◆ 使用 VBScript、JScript 等简单易懂的脚本语言, 结合 HTML 代码, 即可快速地完成网站的应用程序。
- ◆ 无须编译, 容易编写, 可在服务器端直接执行。
- ◆ 使用普通的文本编辑器, 如 Windows 的记事本, 即可进行编辑设计。
- ◆ 与浏览器无关, 客户端只要使用可执行 HTML 代码的浏览器, 即可浏览 ASP 所设计的网页内容。ASP 所使用的脚本语言 (VBScript、JScript) 均在 Web 服务器端执行, 客户端的浏览器不能够执行这些脚本语言。
- ◆ ASP 能与任何 ActiveX 脚本语言兼容。除了可使用 VBScript 或 JScript 语言来设计外, 还通过 plugin 的方式, 使用由第三方所提供的其他脚本语言, 比如 Perl 等。

(2) PHP 的技术特点。PHP 可以编译成与多种数据库相连接的函数。目前, PHP 与 MySQL 是绝佳的组合。你还可以自己编写外围的函数去间接存取数据库。通过这样的途径, 当你更换使用的数据库时, 可以轻松地修改变码以适应这样的变化。PHPLIB 就是最常用的、可以提供一般事务需要的一系列基库。但 PHP 提供的针对各种数据库的接口支持不统

一, 比如对 Oracle、MySQL、Sybase 的接口, 彼此都不一样。这也是 PHP 的一个弱点。

(3) JSP 的技术特点。

详见“1.1.3 JSP 技术特点”。

3. 应用范围

ASP 是 Microsoft 开发的动态网页语言, 也继承了微软产品的一贯传统, 只能执行于微软的服务器产品 IIS (Internet Information Server, Windows NT 平台) 和 PWS (Personal Web Server, Windows 98 平台) 上。UNIX 下也有相关组件来支持 ASP, 但是 ASP 本身的功能有限, 必须通过 ASP + COM 的组合来扩充, 而 UNIX 下的 COM 实现起来非常困难。

PHP 3 可在 Windows、UNIX、Linux 的 Web 服务器上正常执行, 还支持 IIS、Apache 等一般的 Web 服务器, 用户更换平台时, 无须变换 PHP 3 代码, 可即插即用。

JSP 同 PHP 3 类似, 几乎可以执行于所有平台, 如 Windows NT、Linux、UNIX。在 Windows NT 下 IIS 通过一个外加服务器, 如 JRUN 或者 ServletExec, 就能支持 JSP。知名的 Web 服务器如 Apache 支持 JSP, 由于 Apache 广泛应用在 Windows NT、UNIX 和 Linux 上, 因此 JSP 有更广泛的执行平台。虽然现在 Windows NT 操作系统占了很大的市场份额, 但是在服务器方面 UNIX 的优势仍然很大, 而新崛起的 Linux 更是来势不小。从一个平台移植到另外一个平台, JSP 和 JavaBean 甚至不用重新编译, 因为 Java 字节码都是标准的与平台无关的。

4. 性能比较

对这三种语言分别做循环性能测试及存取 Oracle 数据库测试。在循环性能测试中, JSP 只用了令人吃惊的 4s 就结束了 20000×20000 的循环。而 ASP、PHP 测试的是 2000×2000 循环 (少一个数量级), 却分别用了 63s 和 84s。三者分别对 Oracle 8 进行 1000 次 Insert、Update、Select 和 Delete 测试, JSP 需要 13s, PHP 需要 69s, ASP 则需要 73s。

5. 前景分析

目前在国内 PHP 与 ASP 应用最为广泛。由于 JSP 是一种相对较新的技术, 国内采用得较少。但在国外, JSP 已经是比较流行的一种技术, 尤其是电子商务类的网站, 多采用 JSP。国内采用 PHP 的网站有新浪网 (Sina)、中国人 (Chinaren) 等, 但由于 PHP 本身存在的一些缺点, 使得它不适合应用于大型电子商务网站, 而更适合应用于一些小型的商业站点。首先, PHP 缺乏规模支持。其次, 缺乏多层结构支持。对于大负荷站点, 解决方法只有一个: 分布计算。数据库、应用逻辑层、表示逻辑层彼此分开, 而且同层也可以根据流量分开, 群组成二维数组。而 PHP 则缺乏这种支持。另外, PHP 提供的数据库接口支持不统一, 这就使得它不适合运用在电子商务中。

ASP 和 JSP 则没有以上缺陷, ASP 可以通过 COM/DCOM 获得 ActiveX 规模支持, 通过 DCOM 和 Transaction Server 获得结构支持; JSP 可以通过 Sun Java 的 Java Class 和 EJB 获得规模支持, 通过 EJB/CORBA 及众多厂商的 Application Server 获得结构支持。

三者中, JSP 应该是未来发展的趋势。世界上一些大的电子商务解决方案提供商都采用 JSP/Servlet。比较出名的如 IBM 的 E-business, 它的核心是采用 JSP/Servlet 的 WebSphere。国内的网上银行也都采用 JSP/Servlet。

总之, ASP、PHP、JSP 三者都有相当数量的支持者, 由此也可以看出三者各有所长。可根据三者的特点选择一种适合自己的语言。

1.2 图解开发环境

JSP 开发环境有很多种，本书采用的软件环境是：

✓ 操作系统：Windows XP

✓ 开发工具：JDK6 + Tomcat 6.0.18 + MyEclipse 6.0.1 AllInOne 试用版

下面对相关的工具安装及配置做图解说明。

1.2.1 JDK 安装及配置

1. 软件下载

首先到 Sun 官方站点 (<http://java.sun.com/javase/downloads/index.jsp>) 下载 JDK6，如图 1-1 所示。

单击 Download 按钮，Platform 项选择 Windows 平台，如图 1-2 所示，复选框打钩后继续。

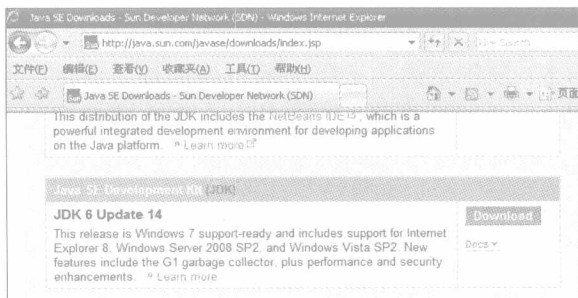


图 1-1 JDK 官方站点



图 1-2 下载 JDK [1]

单击 Continue 按钮后进入文件下载页面，如图 1-3 所示。

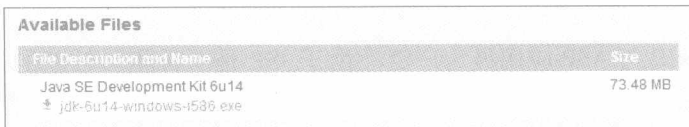


图 1-3 下载 JDK [2]

最后，单击 jdk-6u14-windows-i586.exe 下载。

2. JDK 安装

下载完成后双击 jdk-6u14-windows-i586.exe 进入安装阶段，如图 1-4 所示。

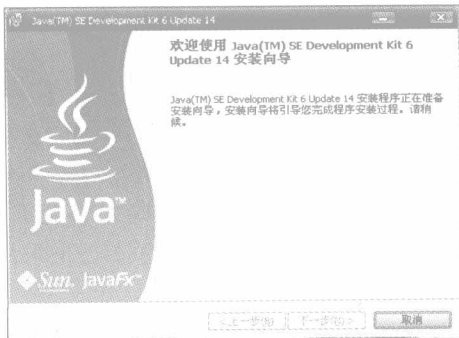


图 1-4 JDK 安装 [1]

如图 1-5 所示, 单击“接受”按钮进入如图 1-6 所示对话框。



图 1-5 JDK 安装 [2]

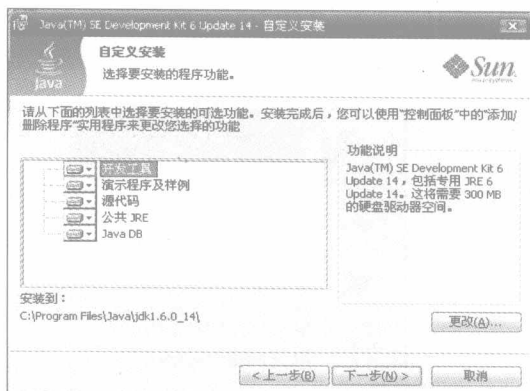


图 1-6 JDK 安装 [3]

这里可以直接单击“下一步”按钮, 如果想更改安装路径, 也可以单击“更改”按钮进行修改。

单击“更改”按钮后的界面如图 1-7 所示。这里选择安装在 E:\jdk1.6.0\目录下。单击“下一步”按钮, 进入正式安装过程。“正在安装”界面如图 1-8 所示。状态是“正在复制新文件”, 等待片刻, 安装即完成, 如图 1-9 所示。



图 1-7 JDK 安装 [4]

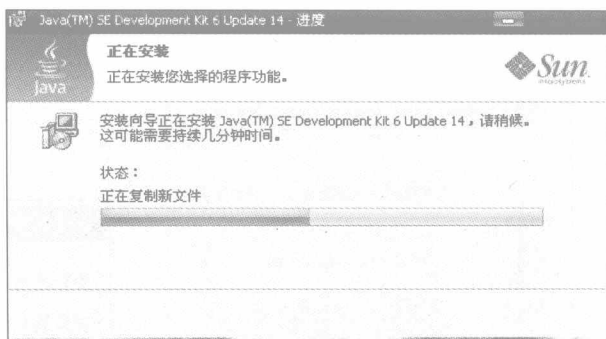


图 1-8 JDK 安装 [5]

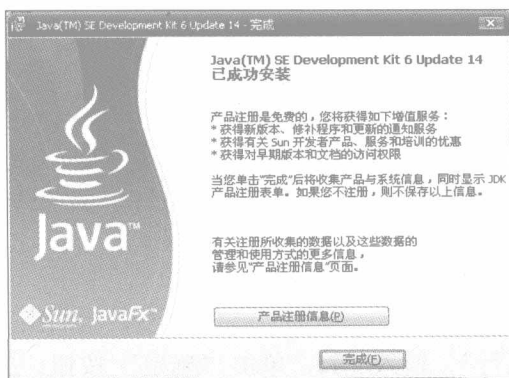


图 1-9 JDK 安装 [6]

3. JDK 环境变量配置

右击“我的电脑”，如图 1-10 所示，在弹出的快捷菜单中选择“属性”，进入如图 1-11 所示对话框。单击“高级”标签，打开“高级”选项卡。

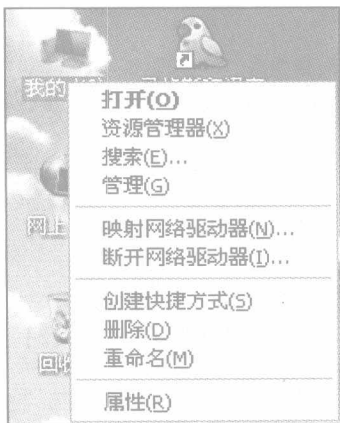


图 1-10 快捷菜单

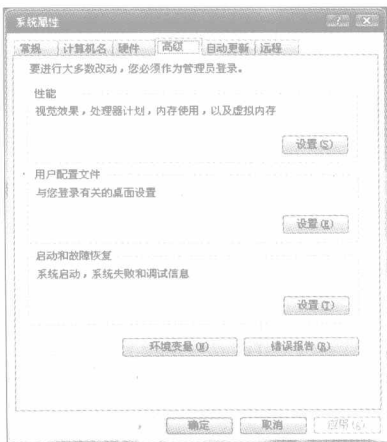


图 1-11 系统属性

单击“环境变量”按钮，进入如图 1-12 所示“环境变量”对话框。

单击“新建”按钮，弹出“新建系统变量”对话框，如图 1-13 所示，这里输入变量名 JAVA_HOME，变量值为 E:\jdk1.6.0，最后单击“确定”按钮，这样一个名为 JAVA_HOME 的环境变量就设置好了。



图 1-12 环境变量

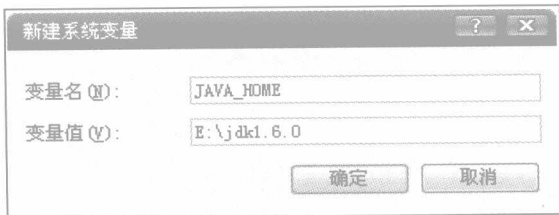


图 1-13 新建系统变量

紧接着，需要找到 Path 变量并编辑它，如图 1-14 所示，在变量值的最前面添加如下内容：“.;%JAVA_HOME%\bin;”，再单击“确定”按钮，这样 Path 系统变量就修改好了。

说明 这里的 %JAVA_HOME% 表示取出上面系统变量名为 JAVA_HOME 的变量值 E:\jdk1.6.0，这样做的好处是以后 JDK 安装路径变化了，只需修改系统变量 JAVA_HOME 的对应值就可以，而不用再修改 Path 变量。

至此，JDK 环境变量配置完成。

4. 测试是否配置成功

选择“开始”|“运行”，输入 cmd，单击“确定”按钮，进入命令行模式，界面如图 1-15 所示。

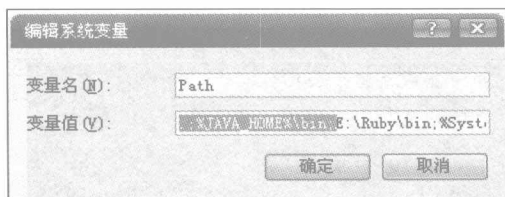


图 1-14 编辑系统变量

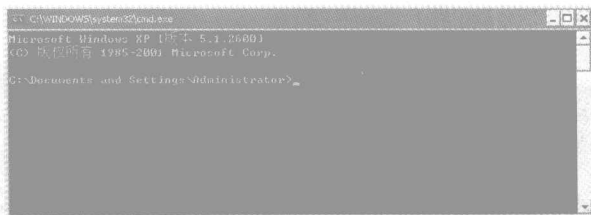


图 1-15 命令行模式

输入 java 后, 显示信息, 如图 1-16 所示。

输入 javac 后, 界面如图 1-17 所示。

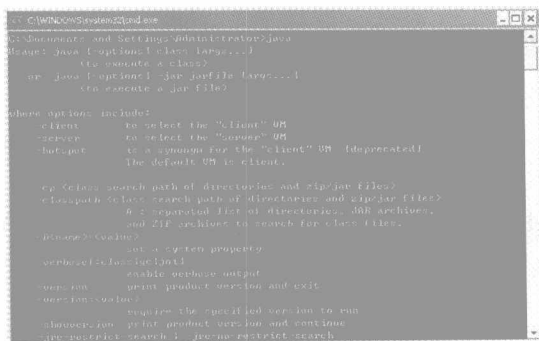


图 1-16 输入 java 后显示信息

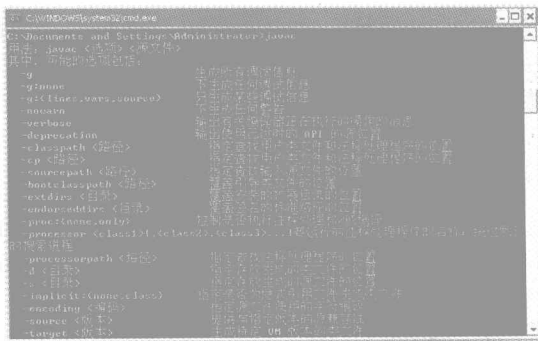


图 1-17 输入 javac 后显示信息

如果以上信息能够正常显示, 说明 JDK 已安装成功。下面用一个简单例子进一步验证。

用记事本写一个简单的 Java 程序, 程序功能为输出 “Hello World!”, 文件命名为 HelloWorld.java, 保存在 E 盘根目录下, 代码如下:

```
class HelloWorld
{
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

在命令行模式, 进到 E 盘根目录下, 输入 javac HelloWorld.java 进行编译。如图 1-18 所示, 没有错误提示, 说明已经成功通过编译, 编译仅仅检查语法是否正确, 并不表示能够正常运行。

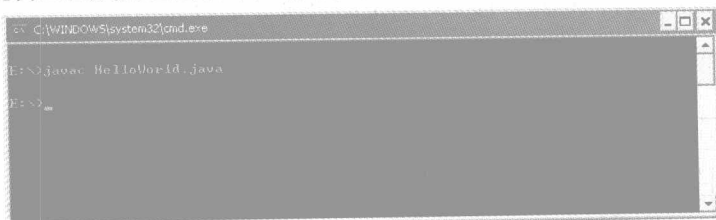


图 1-18 编译类

此时, 输入 dir *.class 命令来查找所有后缀名为 class 的文件。

如图 1-19 所示, 可以看出, 经过编译后, 生成了一个字节码文件, 名叫 HelloWorld.class, 接着运行这个字节码文件。在命令行模式下输入 java HelloWorld (注意字母大小写要严格区