

新世纪高职高专教改项目成果教材  
高等职业教育技能型紧缺人才培养试用

# JSP 程序设计

蒋文蓉 主编

高等教育出版社

## 内容提要

本书是教育部新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目成果，是组织有关教育部高职高专教育、教学改革试点院校编写的。作者从教学的角度出发，从 JSP 的基础知识、基本语法入手，配以实际的范例，细致地讲解了 JSP 网站编程的方法与技巧。

本书共有 8 章，按照循序渐进的学习原则，从基础知识入手，详细讲解了 JSP 的语言规范、HTML 表单的检验方法和 JSP 与客户机的交互等内容，并介绍了 JavaServlet、JavaBeans、JDBC 等重要技术，最后用 JSP 编写了一个比较完整的电子商务网站。本书的特点是内容紧凑、详尽，每章后还配有实验。本书配有光盘，其中不仅有本书的所有代码，还有教学和实验的内容，供学生自学使用。

本书适合高等职业学校、高等专科学校、成人高校、本科院校举办的二级职业技术学院，也适合示范性软件职业技术学院、继续教育学院、民办高校技能型紧缺人才培养使用，还可供本科院校、计算机专业人员和爱好者参考使用。

## 图书在版编目(CIP)数据

JSP 程序设计 / 蒋文蓉主编. —北京：高等教育出版社，2004.6

ISBN 7-04-014776-9

I. J... II. 蒋... 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 (2004) 第 044186 号

策划编辑 冯 英 责任编辑 关 旭 封面设计 王凌波  
版式设计 胡志萍 责任校对 康晓燕 责任印制

出版发行 高等教育出版社  
社 址 北京市西城区德外大街 4 号  
邮政编码 100011  
总 机 010-82028899

购书热线 010-64054588  
免费咨询 800-810-0598  
网 址 [http:// www.hep.edu.cn](http://www.hep.edu.cn)  
[http:// www.hep.com.cn](http://www.hep.com.cn)

经 销 新华书店北京发行所  
印 刷

开 本 787×1092 1/16  
印 张 14.5  
字 数 350 000

版 次 年 月第 1 版  
印 次 年 月第 次印刷  
定 价 24.50 元 (含光盘)

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

**版权所有 侵权必究**

# 出版说明

为认真贯彻《中共中央国务院关于深化教育改革全面推进素质教育的决定》和《面向 21 世纪教育振兴行动计划》，研究高职高专教育跨世纪发展战略和改革措施，整体推进高职高专教学改革，教育部决定组织实施《新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目计划》（教高[2000]3 号，以下简称《计划》）。《计划》的目标是：“经过五年的努力，初步形成适应社会主义现代化建设需要的具有中国特色的高职高专教育人才培养模式和教学内容体系。”《计划》的研究项目涉及高职高专教育的地位、作用、性质、培养目标、培养模式、教学内容与课程体系、教学方法与手段、教学管理等诸多方面，重点是人才培养模式的改革和教学内容体系的改革，先导是教育思想的改革和教育观念的转变。与此同时，为了贯彻落实《教育部关于加强高职高专教育人才培养工作的意见》（教高[2000]2 号）的精神，教育部高等教育司决定从 2000 年起，在全国各省市的高等职业学校、高等专科学校、成人高等学校以及本科院校的职业技术学院（以下简称高职高专院校）中广泛开展专业教学改革试点工作，目标是：在全国高职高专院校中，遴选若干专业点，进行以提高人才培养质量为目的、人才培养模式改革与创新为主题的专业教学改革试点，经过几年的努力，力争在全国建成一批特色鲜明、在国内同类教育中具有带头作用的示范专业，推动高职高专教育的改革与发展。

教育部《计划》和专业试点等新世纪高职高专教改项目工作开展以来，各有关高职高专院校投入了大量的人力、物力和财力，在高职高专教育人才培养目标、人才培养模式以及专业设置、课程改革等方面做了大量的研究、探索和实践，取得了不少成果。为使这些教改项目成果能够得以固化并更好地推广，从而总体上提高高职高专教育人才培养的质量，我们组织了有关高职高专院校进行了多次研讨，并从中遴选出了一批较为成熟的成果，组织编写了一批“新世纪高职高专教改项目成果”教材。这些教材结合教改项目成果，反映了最新的教学改革方向，很值得广大高职高专院校借鉴。

新世纪高职高专教改项目成果教材适用于高等职业学校、高等专科学校、成人高校及本科院校举办的二级职业技术学院、继续教育学院和民办高校使用。

高等教育出版社  
2002 年 11 月 30 日

# 前 言

随着电子商务应用的日渐广泛，人们对电子商务的开发技术提出了更高的要求，开发效率和执行效率都不高的 CGI 和 SSI 技术已不能满足用户的需求；而 ASP 和 PHP 技术尽管开发效率高，但执行效率低、移植性差，其应用范围也受到限制。秉承了 Java 优势的 JSP，是一种与平台无关、基于 `JavaServlet` 以及整个 Java 体系的 Web 开发技术，它代码执行效率高、开发周期短、可扩展性强且技术规范公开，因而被越来越多的网站采用。

本书主要介绍 JSP 的编程知识并通过实例将其应用到动态网站开发中去。本书从概念上详细描述了 JSP 技术原理、运行环境、语法及基本指令的运用，对 `JavaBean`、`JavaServlet`、数据库运用等 JSP 核心技术分别进行了详细的描述。教材中对 JSP 指令的描述较详细，由指令说明、指令格式、语法说明、举例等部分组成，并以丰富的实例，详细介绍了 JSP 内置对象。教材的总体特点是内容紧凑、详尽，采用文字、图案、表格等多种描述方式，每章后配有实验。另外，书后还配有光盘，其中不仅有本书的所有代码，还有教学和实验的内容，供学生自学使用。

本书共有 8 章。第 1 章是 JSP 运行环境和其他动态网页技术，主要介绍了 JSP 的运行环境、实现原理以及 JSP 技术和其他动态网页技术的比较，并补充了一些 `JavaScript` 中关于表单检验的知识；第 2 章讨论了 JSP 的语法；第 3 章详细讲述了 JSP 内置对象的使用；第 4 章介绍了 JSP 中访问数据库的技术——`JDBC` 的工作原理、数据库的一些基本知识以及 JSP 与不同数据库连接的几种方式；第 5 章介绍了 Java 组件技术——`JavaBean`，以及如何通过该项技术提高 JSP 的编程和执行效率；第 6 章阐述了 Java 服务器小程序——`Java Servlet` 技术，并应用该技术实现了一个简单的在线电子邮件系统；第 7 章介绍了 JSP 开发中的常用技巧，把平时常见的一些问题都列举出来，并加以分析；第 8 章给出了一个综合实例，如购物车、论坛等，通过这些实例介绍一些比较完整的网站开发过程。

本书由上海第二工业大学蒋文蓉、阎季鸿编写，蒋文蓉任主编，在编写工作中，上海第二工业大学软件系的不少教师给予了热情的帮助，郑健老师在百忙之中抽出时间审阅了全稿，在此表示衷心的感谢。

本书面向的读者是广大编程爱好者、网站开发人员和数据库开发人员。本书不仅适合作为高等职业技术学校计算机专业课的教材，也可用于计算机培训学校的教材以及网站开发人员参考使用。

为了及时满足广大读者的需求，编写时间很仓促，虽然我们尽了很大努力，但难免还会有疏漏和不妥之处，恳请广大教师和读者提出宝贵意见。

编 者

2003 年 12 月于上海

# 目 录

第 1 章 JSP 运行环境和其他动态网页技术	1
1.1 动态网页技术	1
1.2 JSP 技术原理	2
1.3 JSP 与其他动态网页技术的比较	3
1.4 JSP 开发和运行环境的配置	5
1.5 JSP 中常用 Java 知识回顾	10
1.6 JavaScript 知识回顾	19
实验	25
第 2 章 JSP 的语法	26
2.1 JSP 的语法结构	26
2.2 JSP 的编译指令	33
2.3 JSP 的动作指令	41
2.4 JSP 语法实例——网站日历程序	51
实验	55
第 3 章 JSP 的内置对象详解	56
3.1 对象与内置对象	56
3.2 request 对象	58
3.3 response 对象	62
3.4 session 对象	64
3.5 application 对象	67
3.6 out 对象	70
3.7 其他对象	73
3.8 两个 JSP 内置对象应用实例	73
实验	77
第 4 章 数据库与 JDBC 技术	78
4.1 JDBC 技术工作原理	78
4.2 通过 JDBC 访问数据库	80
4.3 通过 JDBC-ODBC 桥来访问数据库	87
4.4 树状管理菜单设计实例	93
实验	109
第 5 章 JavaBean	110
5.1 什么是 JavaBean	110
5.2 JavaBean 技术概述	111
5.3 JavaBean 的应用	118
实验	124
第 6 章 JavaServlet 技术	125
6.1 Servlet 技术概述	125
6.2 Servlet 与 JSP 之间的关系	128
6.3 Servlet 应用范围及其缺陷	129
6.4 Servlet 的生命周期	129
6.5 Servlet 的常用接口类	130
6.6 JSP 内置对象与 Servlet 中类的对应关系	132
6.7 Servlet 与 JSP 结合编程	133
实验	162
第 7 章 JSP 开发中的常用技巧	163
7.1 在不同页面或者用户之间共享数据	163
7.2 创建错误处理页面	171
7.3 JSP 操作文件	176
实验	181
第 8 章 综合实例	182
8.1 商品管理	182
8.2 商品分类和查询	199
8.3 用户订单处理	205
8.4 论坛	217
实验	223
参考文献	224

# 第 1 章 JSP 运行环境和其他动态网页技术

理论	介绍 JSP 概念、特点，各种动态网页技术。
技能	学会安装和配置 JSWDK、安装和配置 Tomcat。
要求	了解 JSP 与其他动态网页技术的区别，并掌握 JSP 运行环境的安装和配置方法。

JSP 是一种与 ASP、PHP 类似的 Web 开发语言，它能快速地创建和显示 Web 页面的动态内容。本章将对 JSP 的原理和特征进行简要介绍。

## 1.1 动态网页技术

在目前呈几何增长的互联网网站中，有相当一部分仍固守着“静态”，这无形中已大大落后于时代的步伐。所谓“静态”指的就是网站的网页内容“固定不变”，当用户浏览器通过互联网的 HTTP (Hyper Text Transport Protocol) 协议向 Web 服务器请求提供网页内容时，服务器仅仅是将原已设计好的静态 HTML 文档传送给用户浏览器。其页面内容使用的仅仅是标准的 HTML 代码，最多再加上流行的 GIF 格式的动态图片，比如产生几只小狗小猫跑来跑去的动画效果。若网站维护者要更新网页的内容，就必须手动地来更新其所有的 HTML 文档。

“静态”网页的致命弱点就是不易维护，为了不断更新网页内容，必须不断地重复制作 HTML 文档，随着网站内容和信息量的日益扩增，工作量就会大得惊人。

有人不禁要问：“什么是动态网页呢？”所谓“动态”，并不是指那几个放在网页上的 GIF 动态图片，这里动态页面的概念应包括以下内容：

① “交互性” 即网页会根据用户的要求和选择而动态改变和响应，将浏览器作为客户端界面，这将是今后 Web 发展的趋势。

② “自动更新” 即无须手动地更新 HTML 文档，便会自动生成新的页面，可以大大节省工作量。

③ “因时因人而变” 即当不同的时间、不同的人访问同一网址时会产生不同的页面。

利用 CGI、ISAPI、ASP、PHP、JSP 和其他有关的技术，一个用户可以将一个 HTTP 请求发送到一个可执行应用程序而不是一个静态的 HTML 文件。服务器将会立即运行这个指定的程序，对用户的输入做出反应，将处理结果返回客户端，或者对数据的记录进行更新。通过这个模型，就可以在服务器和客户机之间有效地进行交互。

现在用户已经对“动态”网页有了一个基本的概念，接下来应挑选合适的工具制作网页。将网页“动态”化的方法很多，挑选时要看是出于何种需求。如果维护的是一个个人网站，使用的是免费主页空间，那么绝大多数情况下只能使用 Java、JavaScript 和最新的 DHTML 技术；如果用户的主页空间提供者能给予了 CGI 权限或对 ASP、JSP 的支持，那么得到的将是真正的

动态技术。本书主要介绍 JSP 技术。

## 1.2 JSP 技术原理

JSP (Java Server Pages) 是由 Sun Microsystems 公司倡导、由许多公司参与一起建立的一种动态网页技术标准, 其网址为 <http://www.javasoft.com/products/jsp>。在传统的 HTML 文件 (\*.htm 或 \*.html) 中加入 Java 程序片段 (Scriptlet) 和 JSP 标记 (tag), 就构成了 JSP 网页 (\*.jsp)。

动态网页需要有的功能: 程序片段可以操作数据库、重新定向网页以及发送 E-mail 等。所有程序操作都在服务器端执行, 网络上传送给客户端的仅是得到的结果, 对客户浏览器的要求最低, 可以实现无 Plugin、无 ActiveX、无 Java Applet, 甚至无 Frame。

JSP 提供了一种简单、快速建立基于动态内容显示的站点技术。在 JSP 文件中可以包含 HTML 标记、NCSA 标记、Servlet 标记和 JSP 语法, 使得 HTML 的编码从 Web 页面的业务逻辑中有效地分离出来。

与传统的 CGI 方式相比, JSP 有如下优点:

- ① 后台实现逻辑是基于 Java Component 的, 具有跨平台的特点。
- ② 将应用逻辑与页面分离, 使得应用逻辑能够最大程度得到复用, 从而提高开发效率。
- ③ 运行效率比 CGI 方式高, 尤其对于数据库访问提供了连接池缓冲机制, 使运行所需资源最少。
- ④ 安全。由于后台是完全基于 Java 技术的, 安全性由 Java 的安全机制予以保障。
- ⑤ 由于与 ASP 很近似, 不需要太多的编程知识就可以动手编写 JSP。
- ⑥ 支持 XML, 使用 XML 可以使页面具有更强的表现力并可减少编程工作量。

JSP 是一种基于 Java 的脚本技术。它能将 HTML 编码从 Web 页面的开发过程中有效地分离出来, 可以访问可重用的组件, 如 Servlet、JavaBean 和基于 Java 的 Web 应用程序, 还支持在 Web 页面中直接嵌入 Java 代码。JSP 有两种访问格式:

### (1) 请求一个 JSP 文件

在这种访问格式中, JSP 文件将访问 Bean 或其他能生成动态内容并发送到浏览器的组件, 参见图 1-1。当 Web 服务器接收到对一个 JSP 文件的请求时, 把请求发送至 JSP 引擎, JSP 引擎对 JSP 文件进行语法分析并生成 Java 源文件 (被转换为 Servlet 然后编译执行)。除非已经更新了原始的 JSP 文件, 否则 Java 源文件的生成和编译仅在初次调用 Servlet 时进行。若更新了原始的 JSP 文件, JSP 引擎将检测所做的更新, 并在执行之前重新转换和编译。

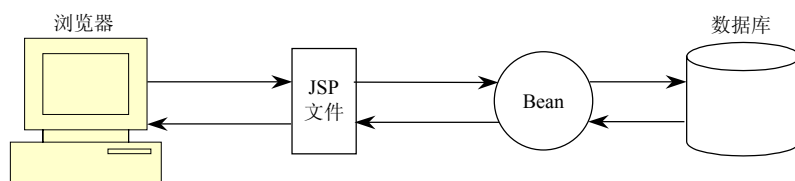


图 1-1

### (2) 请求一个 Servlet

在这种访问格式中，客户端直接请求一个 Servlet 来生成动态内容，并调用 JSP 文件将内容发送到浏览器（参见图 1-2）。该访问模型将内容生成部分从内容显示中分离了出来。JSP 引擎采用 `HttpServletRequest` 对象和 `HttpServletResponse` 对象的一些方法，通过 Servlet 将一个对象放入请求对象（通常是一个 Bean）中，并将该请求的结果传递到另一个页面（通常是一个 JSP 文件）以供显示。调用的页面从请求对象中检索 Bean，并用 JSP 来生成客户端的 HTML。

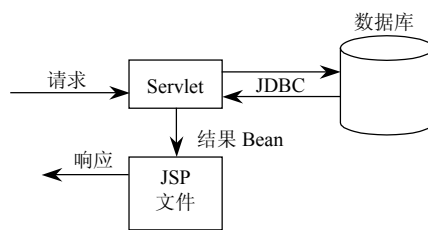


图 1-2

### 1.3 JSP 与其他动态网页技术的比较

ASP (Active Server Pages)，是一种 Web 服务器端的开发环境，利用它可以产生和运行动态的、交互的、高性能的 Web 服务应用程序。ASP 采用脚本语言 VB Script (Java Script) 作为自己的开发语言。

PHP (Personal HomePage) 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，并整合 PHP 自己的特性，使 Web 开发者能够快速写出动态页面。PHP 支持目前绝大多数数据库，而且是完全免费的。可以从 PHP 官方站点 (<http://www.php.net>) 自由下载，不受限制地获得源码，甚至可以在其中加进自己需要的特色。

JSP (Java Server Pages) 是 Sun 公司推出的新一代站点开发语言，它完全解决了目前 ASP、PHP 的一个通病——脚本级执行。Sun 公司借助自己在 Java 上的不凡造诣，使 Java 在 Java 应用程序和 Java Applet 之外又演生出 JSP。JSP 可以在 Servlet 和 JavaBean 的支持下，完成功能强大的站点程序。

3 种常用的动态网页技术的共同点：

① 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。

② 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，来分析和执行程序代码。程序代码的执行结果被重新嵌入 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。

下面，引用一些实验结果数据来比较这三者之间的性能差异。

#### 1. 循环性能测试

##### (1) JSP

循环次数：10 000×10 000 次

运行时间：少于 2 s

##### (2) PHP

循环次数：1 000×1 000

运行时间：8 s

## (3) ASP

循环次数: 1 000×1 000

运行时间: 9 s

可以看到, JSP 程序的循环次数是 ASP、PHP 的 100 倍时, 它们的运行时间才在同一个数量级。

## 2. 数据库连接测试

由于 JSP 在连接数据库时需要通过 JDBC, 最好通过一个连接数据库的 JavaBean 来完成。这里只给出测试方法和结果。

测试方法: 对 Oracle 8 进行 2 000 次的插入 (Insert)、更新 (Update)、查询 (Select) 和删除 (Delete) 操作。

实验结果:

JSP: 19 s

PHP: 57 s

ASP: 58 s

由此可见 JSP 的运行速度远远超出 PHP、ASP。

作为动态 Web 开发语言, JSP 与其他的 Web 开发语言相比有什么不同, 又有什么长处呢?

可以肯定地说, 动态网页技术有很多种, 而 JSP 只是其中的一种选择。对资深的网络管理员而言, 动态网页的写作界面应随着需求而变动, 综合比较表 1-1 中各项性能以后, 可以看出 JSP 是最适合的主页开发工具。

表 1-1 JSP 和其他语言比较

程序界面	PHP	ASP	CGI	NSAPI	ISAPI	JSP
操作系统	UNIX/Linux、Win32 等均可	Win32	均可	均可	Win32	UNIX/Linux、Win32 等均可
Web 服务器种类	数种 (Apache、IIS/PWS 等)	IIS	均可	Netscape Server	IIS	很多
执行效率	快	快	慢	极快	极快	快
稳定性	好	中等	最好	差	差	好
支持函数数量	多	少	不定	中等	少	很多
程序语言	PHP	VB	不限	C/C++	C/Delphi	Java
与网页的结合性能	佳	佳	差	差	差	佳
系统安全性	好	极差	最好	好	一般	很好
开发时间长短	短	短	中等	长	长	短
修改时间长短	短	短	中等	长	长	短
改版速度快慢	快	慢	无	慢	慢	较快
目前应用网站数量	极多	多	多	极少	少	多
学习的难易度	低	低	高	极高	高	低

从表 1-1 可以看出, PHP 可用在数种 Web 服务器上, 传统 CGI 不受任何一种操作系统或 Web 服务器平台的限制, NSAPI 一定要在 Netscape 服务器 (如 Netscape Enterprise Server 或 FastTrack Server) 上才可以执行, 但可支持多种操作系统 (UNIX 或 Win32); ASP 及 ISAPI 只

在 IIS 上有完整的功能。

在稳定性方面，由于 NSAPI 或 ISAPI 是动态链接的方式，因此在执行时若出现问题，会使得 Web 服务器瘫痪。在实际应用 ASP 的过程中发现，ASP 会周期性地使系统不稳定，需要不断重新启动操作系统。JSP 在许多网站的应用中，长期使用都没有问题，程序的稳定性也不错。当然最稳定的是传统的 CGI 程序，因为它由操作系统负责控制，不会因 CGI 程序的错误导致 Web 服务器的不稳定。

在开发及维护时间上，JSP、PHP 及 ASP 都有不错的表现。而 NSAPI 及 ISAPI 则需要较长的开发过程。但是在稳定性达到要求以后，这两种界面程序便成为效率最佳的选择。

如比较与网页结合的能力可以发现，只有 JSP、PHP 和 ASP 才能内嵌 HTML 语言。这是 JSP、PHP 和 ASP 成为当前最流行的网页开发语言的主要原因之一，但这也是影响开发周期的重要因素之一。

就系统安全性而言，ASP 是最差的，如果没有安装微软的 IISServicePack，使用命令“:: \$DATA”就可以看到 ASP 的源代码，这对开发人员来说，是不能容忍的。当然，传统的 CGI 程序，由于是由操作系统直接管理的，破解的难度最高。

总而言之，运行在 Web 后端的开发程序，运行效率极高而开发又极为方便的选择还不存在，但是 JSP 语言目前是处于开发手段与效率的最佳平衡点上。

## 1.4 JSP 开发和运行环境的配置

### 1.4.1 JSP 开发环境的安装

这里推荐使用 UltraEdit 文本编辑工具。它是功能极其强大的文本编辑器，现在的版本已经能够支持 JSP 语法，在它的编辑环境下 JSP 语句以不同颜色显示。如图 1-3 所示为 UltraEdit 文本编辑环境。

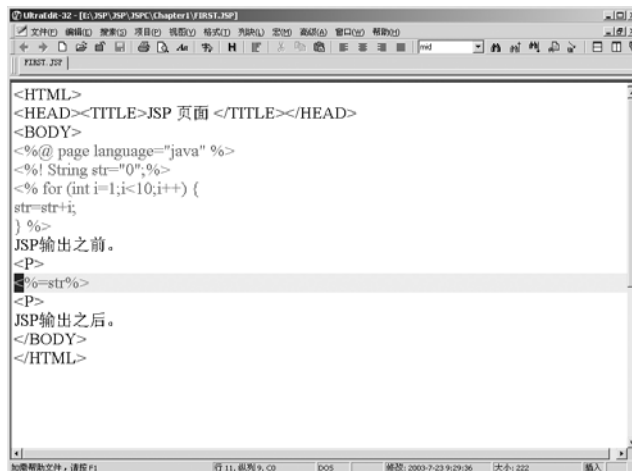


图 1-3

### 1.4.2 JSP 运行环境的配置

如果想用 JSP 来开发网站，需要安装一些应用程序服务器软件，来专门提供对 JSP/Servlet 的支持。这样就可以使用主服务器响应普通网页的请求，应用程序服务器软件来响应动态网页的请求。

运行 JSP 的服务器软件有很多种，如 JSWDK、Tomcat、Websphere 等，但考虑到大多数读者所使用的是 Windows 系列操作系统，因此这里主要介绍 Windows 操作系统下的 JSP 运行环境的安装和配置。

#### 1. JSWDK 的安装和配置

① 要想在 JSWDK 环境下运行 JSP，需要先安装 JDK。这里选择 JDK1.3 版本。要注意：以前的版本不支持 JSP 和 Servlet 技术，因此尽可能使用最新版本的 JDK，读者可以在 Sun 公司的主页 <http://java.sun.com> 上下载最新版本的 JDK。

运行 JDK 的安装程序，按照提示完成 JDK 的安装，这里安装的目的路径为 [x:] \ JDK1.3。下面进行 CLASSPATH 和 PATH 的设置。

② 修改系统环境参数。如果是在 Windows 9x/2000 上安装，将“set JAVA\_HOME=[x:]jdk1.3”这一行加到 autoexec.bat 文件中；如果是 Windows NT，可在“控制面板”→“系统”→“环境”页中选择 path 参数并在末尾加入“; [x:]jdk1.3\bin”。同样地可增加新的环境参数“CLASSPATH=[x:]jdk1.3\lib\tools.jar”，然后重新启动系统即可。

如果是中文系统，系统注册表会有问题，可用 regedit 打开注册表查找文件 javasoft，位置为 hkey\_local\_machine→software→jvasoft，找到“Java 运行时环境”，将“Java 运行时环境”改为“Java Runtime Environment”。

③ JSWDK 的启动。安装 JSWDK 之后，只需执行 startserver.bat 命令即可启动服务器，如图 1-4 所示。

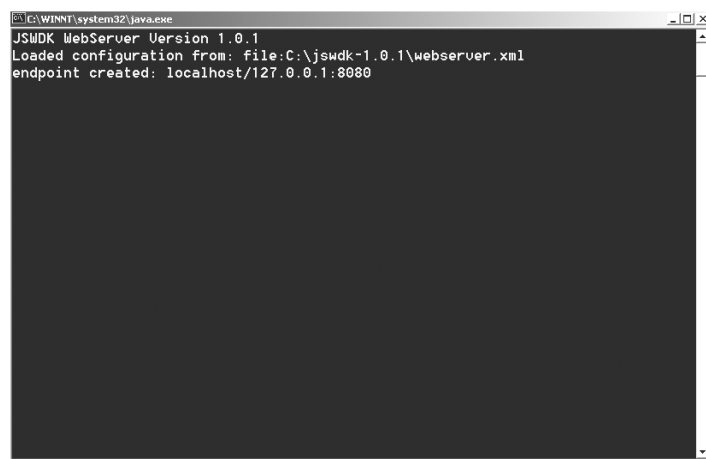


图 1-4

如果是在 Windows 9x 下，则在运行 startserver.bat 时可能出现错误，此时可用鼠标右键单

击 startserver.bat 文件图标，在弹出的菜单中选择“属性”命令，然后在弹出的属性窗口中选择“内存”选项卡，然后修改“常规内存”中“初始环境”的值，将“自动”改为“4096”，单击“确定”即可。

④ JSWDK 的测试。在默认配置下服务器端口为 8080，使用 <http://localhost:8080> 即可打开默认的 index.html 页面，如图 1-5 所示。

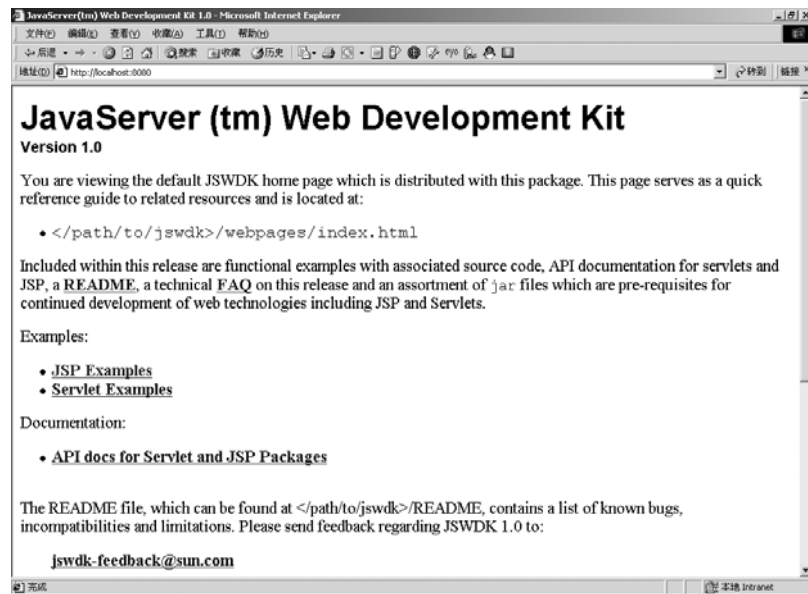


图 1-5

用户每次开机以后都要执行文件 startserver.bat 才可启动 JSP 服务器。这是在 DOS 窗口中运行的程序，如果要退出，可执行 stopserver.bat。

如果 8080 端口已经被占用，可以打开 JSWDK 目录下的 webserver.xml 文件，对此端口重新进行设置（实际上该文件包含了所有的配置），将 port 设为其他端口即可。

如果用户要建立自己的执行目录，可以在 webserver.xml 文件中增加如下设置：

```
<webApplication id="myweb" mapping="/myweb" docBase="myweb"/>
```

打开 JSP 页面的方式为：<http://localhost:8080/myweb/xxx.jsp>

JSP 中的 Java 代码均在服务器端执行。因此，在浏览器中使用“查看源文件”菜单是无法看到 JSP 源代码的，只能看到结果的 HTML 代码。

## 2. Tomcat 的安装和配置

① 首先安装好 JDK1.3，然后下载 Tomcat。下载网址为：

<http://jakarta.apache.org/builds/tomcat/release/v3.1/bin/jakarta-tomcat.zip> 2.26mb

将 jakarta-tomcat.zip 包内文件解压缩到 [x:]tomcat。这里 [x:] 是驱动器名。

要启动 Tomcat 很简单，首先可以查看 tomcat\bin\startup.bat 文件，利用控制面板里的“系统”→“高级”→“环境变量”对话框，新建环境变量 TOMCAT\_HOME 和 JAVA\_HOME，使

它们各自指向 Tomcat 和 JDK 的根目录即可。然后开启一个 DOS 窗口执行 startup.bat, 如图 1-6 所示。

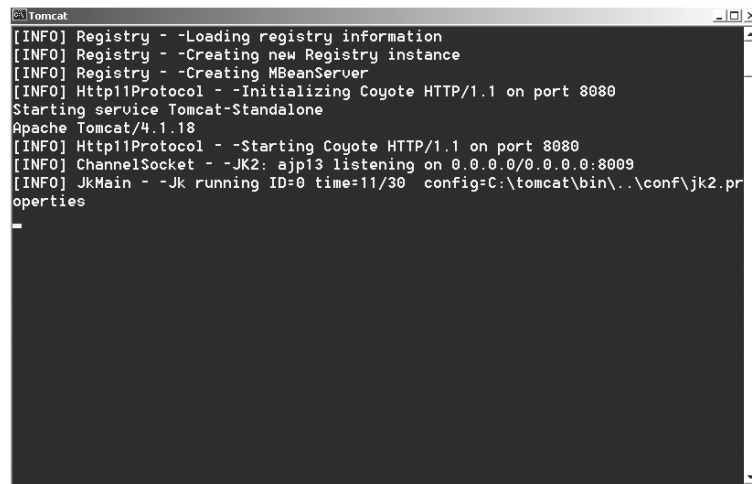


图 1-6

② 访问 <http://localhost:8080> (如图 1-7 所示), 可以看到 Tomcat 的一些示例。

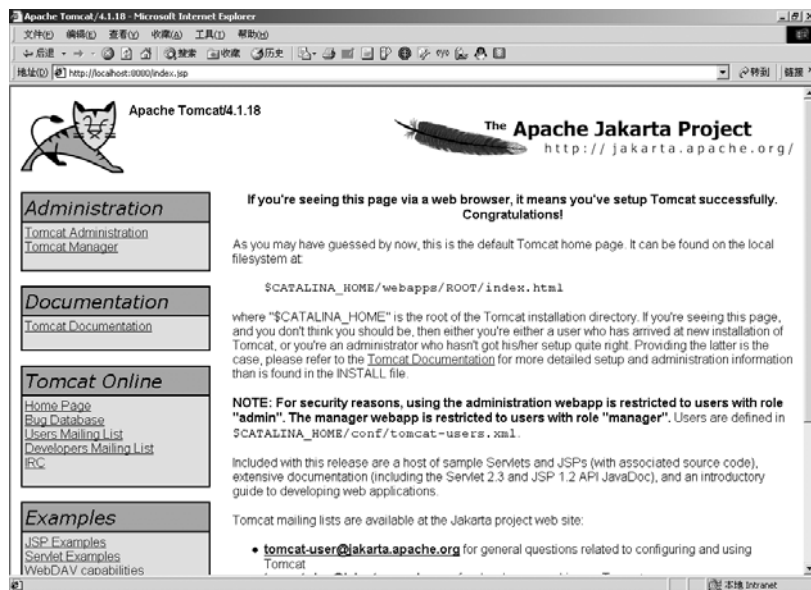


图 1-7

Tomcat 是本书推荐使用的服务器软件, 在接下来的章节中将详细讲述 Tomcat 的使用。

### 1.4.3 Tomcat 运行不成功的错误分析

如果安装了 Tomcat 和 JDK 后运行 startup.bat 不成功, 解决方法是:

- ① 检查所安装的 JDK, 如果没有 JDK, 则重新安装 JDK1.3, 重新启动计算机。
- ② 检查 classpath 和 path 路径的设置, 看设置的值是否正确。如果有错, 经修正后需重新启动计算机, 这是因为系统环境的改变有时需要重新启动计算机才能生效。
- ③ 反复检查前面 2 项, 仍不成功, 则需将 JDK 环境删除, 再重新设置。
- ④ 如果操作系统是 Windows 9x 系列, 需要检查 startup.bat 文件的属性, 其常规内存的初始环境必须大于 4096。

综上所述, 出现错误的时候, 要反复检查, 同时重新启动计算机, 直至 JDK 及 Tomcat 的设置成功。

### 1.4.4 Tomcat 的使用

这里提供了一些关于 Tomcat 的基础信息, 主要有:

- ① Tomcat 二进制版本。
- ② 与 Tomcat 相关的脚本的主要内容。
- ③ 与 server.xml 相关的主要内容, Tomcat 的主要配置文件。
- ④ 如何设置 Tomcat 与宿主 Web 服务器一起工作的说明。
- ⑤ 如何应用 Tomcat 配置一个现实中的 Web 站点。
- ⑥ 希望此文档足以使新用户开始使用 Tomcat。

如找不到某方面的内容可按以下顺序进行:

- ① 查找 Tomcat faq。
- ② 查找 Tomcat 包清单。
- ③ 向 Tomcat Online 中的 Users Mailing List 发送问题。

假设已将 Tomcat 解压缩, 将得到下列目录结构及其描述:

- ① bin 包含启动/关闭 Tomcat 服务器的脚本。
- ② conf 包含不同的配置文件, 包括 server.xml (Tomcat 的主要配置文件) 和为不同的 Tomcat 配置的 Web 应用设置默认值的文件 web.xml。
- ③ doc 包含各种 Tomcat 文档。
- ④ lib 包含 Tomcat 使用的 jar 文件。在 UNIX 平台中, 此目录下的任何文件都被加到 Tomcat 的 classpath 中。
- ⑤ logs Tomcat 放置日志文件的地方。
- ⑥ src ServletAPI 源文件。内有必须在 Servlet 容器内实现的空接口和抽象类。
- ⑦ webapps Web 项目示例。
- ⑧ work Tomcat 自动生成, 放置 Tomcat 运行时的临时文件(如 JSP 文件编译后产生的 Java 文件和 Class 文件)。若在 Tomcat 运行时删除此目录, JSP 页面将不能运行。
- ⑨ classes 可以在此目录中添加一些附加的类到类路径中。任何加到此目录中的类都可在 Tomcat 的类路径中找到。

下面列出对一般用户最重要的脚本。

- ① Startup 用于在后台启动 Tomcat。
- ② Shutdown 用于关闭 Tomcat。

#### 1.4.5 Tomcat 的配置

Tomcat 的主要配置文档是 server.xml, 可以通过配置 server.xml 设置服务器的端口及虚拟路径。

##### 1. 配置服务器的端口

- ① 在 server.xml 文件中找到如下代码:

```
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
    port="8080" minProcessors="5" maxProcessors="75"
    enableLookups="true" redirectPort="8443"
    acceptCount="100" debug="0" connectionTimeout="20000"
    useURISValidationHack="false" disableUploadTimeout="true" />
</Connector >
```

- ② 修改端口行代码 port="8080", 便可以修改计算机的端口。

##### 2. 配置服务器的虚拟路径

- ① 在 server.xml 文件中找到如下代码:

```
<Context path="/examples" docBase="examples" debug="0"
    reloadable="true" crossContext="true">
</Context>
```

- ② 可以添加如下代码, 增加一个虚拟路径:

```
<Context path="/yjh" docBase="E:\yjh\exp" debug="0"
    reloadable="true" crossContext="true"/>
</Context>
```

总之, Tomcat 服务器是一个比较简单的服务器, 在设计方面符合建立商业站点的规范, 可以利用它进行编程与测试。

## 1.5 JSP 中常用 Java 知识回顾

学习 JSP 之前, 需要回顾一下 Java 知识, 包括 Java 的数据类型、运算符、表达式、基本语句的操作以及定义过程等。本节中, 将重点回顾关于异常处理和文件类流的知识。

### 1.5.1 异常处理

#### 1. 异常的概念

Java 中的异常 (Exception) 是指当程序中出现不同寻常的情况时生成的一个对象。这种异常对象的数据成员存储着故障的情况。一般抛出 (thrown) 异常是指抛出包含异常情况的对象, 它可以作为参数, 传递给专门为处理相关问题而编写的程序代码。

当接收异常对象作为形式参数时，称这个异常被捕获（catch）。

造成异常情况的条件千差万别，可以把它们分为 4 类，如表 1-2 所示。

表 1-2 造成异常情况的条件

异常类型	产生异常的条件
代码或者数据错误	举例来说，试图进行对象的非法转型，试图使用超界的数组索引，或者整数算术表达式中除数为 0
标准方法异常	如果在 String 类中使用 substring()方法，系统就会抛出 StringIndexOutOfBoundsException 异常
抛出自己的异常	在需要的时候抛出自己的异常
Java Errors	可能由运行编译程序的 Java 虚拟机引起，但通常由程序中的错误造成代码或者数据错误

异常通常表现为一个错误，之所以这样称呼是因为 Java 程序中的错误就是规则的异常情况。当然异常并不总表现为错误，它也可以表现为程序中出现了一些特别不寻常的事件，需要引起特别注意。

如果在处理程序正常操作的代码中，用户试图处理各种可能出现的很不寻常的出错情况，那么用户的程序结构就会变得非常复杂并且难于理解。

引入异常的概念有两个好处：

- ① 它把出错的代码与正常执行的代码分开。
- ② 提供一种针对特定错误做出强制响应的办法。有许多类型的异常，用户必须要在自己的程序中对它们进行处理，否则用户的代码将无法编译。

## 2. 异常的类型

图 1-8 为异常的类型结构图。

### (1) Error

Error 类及其子类描述的是 Java 将字节码装入内存的过程中和对字节码进行检查的过程中遇到的问题以及 Java 的内部错误及资源耗尽的情况。这类异常由 Java 直接处理，用户程序不需理会。

### (2) Exception

Exception 类对象是由 Java 程序处理或抛出的对象。它有各种不同的子类分别对应不同类型的异常。其子类分为：运行时的异常类和其余的异常类（一般称检查型异常）。

① RuntimeException。对于 RuntimeException 类的子类表示的异常，如果用户的程序代码可能生成这些异常，那么必须在程序中包含一些代码来处理它们。如果程序中的一个方法有可能生成此类异常，必须要么在该方法中处理异常，要么记录下该方法可能抛出的异常。如果不这样做，程序将无法编译。

标准包 java.lang 中定义的 RuntimeException 子类如表 1-3 所示。

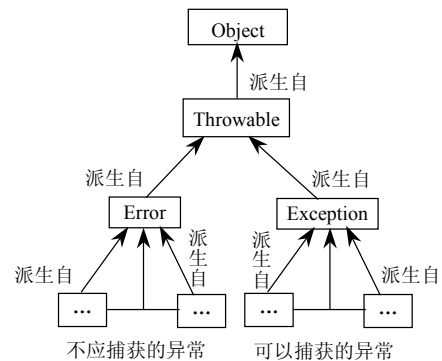


图 1-8

表 1-3 RuntimeException 各子类及产生的条件

类 名	表示的异常条件
ArithmeticException	出现非法算术操作，如试图把一个整数除以零
IndexOutOfBoundsException	试图使用一个超界的索引来引用一个对象。这个对象可以是一个数组、一个 String 对象或者一个 Vector 对象。Vector 类在标准包 java. util 中定义
NegativeArraySizeException	试图用一个负维数值定义一个数组
NullPointerException	当需要使用一个对象变量引用一个对象进行正常操作时（例如调用一个方法或者访问一个数据成员的时候）该变量包含的值为 null
ArrayStoreException	试图把一个不允许的类型的对象存储在一个数组中
IllegalArgumentException	传递给一个方法的实际参数与该方法的形式参数类型不符
ClassCastException	试图把一个对象转型成一种非法类型，这个对象不是指定的类，也不是指定类的子类或者超类
SecurityException	程序执行了违反安全规定的非法操作。例如试图由一个 applet 程序读取本地计算机上的一个文件
IllegalMonitorStateException	一个线程试图在监视器上等待一个不是由该线程拥有的对象
IllegalStateException	试图在不允许调用方法的时间调用一个方法
UnsupportedOperationException	如果要求执行一个不支持的操作，则抛出此异常

② 检查型异常。Exception 的子类中除了运行时产生异常以外的部分称为检查型异常，程序中使用的方法会抛出检查型异常，Java 编译器要求用户在程序中必须捕获处理或声明抛出这些异常。常见的检查型异常有以下 3 种：

- 包 java.lang 中声明的标准检查型异常
- 包 java.io 中声明的标准检查型异常
- 包 java.net 中声明的标准检查型异常

### 3. 异常的处理

如果想在异常发生的地方进行异常处理，则可以在该方法的代码段中包含 3 类代码：try、catch 和 finally 代码块（简称块）。

#### (1) try 代码块

try 代码块中包含的是可能引起一个或者多个异常的代码。用户希望捕获的、会抛出异常的代码必须位于 try 块中。

代码块是关键字 try 后面括号包围着的、会引起异常的代码。

```
try
{
//Code that can throw one or more exceptions
}
```

注：当想捕获一个异常的时候，可能会引起异常方法中的代码必须包含在一个 try 块中。

#### (2) catch 代码块

catch 代码块包含着计划用于处理一个 try 块中可能抛出的某种特定类型异常的代码。这个 catch 代码块必须立即跟随着包含会抛出特定异常的 try 代码块。catch 代码块由关键字 catch 和一对括号构成，括号中有一个形式参数指出这个代码块处理的异常类型，后面是括号中用于处理异常的代码。