

高等学校计算机科学与技术教材

JSP Web 应用教程

Computer
朱 敏 朱晴婷 李媛媛 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社
<http://press.bjtu.edu.cn>

高等学校计算机科学与技术教材

JSP Web 应用教程

朱 敏 朱晴婷 李媛媛 编著

清华大学出版社
北京交通大学出版社
·北京·

内 容 简 介

本书由浅入深、详尽地介绍了 JSP 的基本原理与相关技术。并从实践出发,引导读者学习和掌握 JSP 这一网络开发技术。本书共分 8 章,介绍了 JSP 的开发和运行环境以及 JSP 的基本原理和语言规范、Java Servlets 技术、JavaBean 技术、JDBC 技术,以及运用 JSP、Servlets 和 JavaBean 开发 Web 应用等。

本书可作为高等院校计算机应用专业的教学用书,也适合自学者及网页开发人员参考使用。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

JSP Web 应用教程/朱敏,朱晴婷,李媛媛编著. —北京:清华大学出版社,北京交通大学出版社,2004.7

(高等学校计算机科学与技术教材)

ISBN 7-81082-244-6

I .J… II .①朱…②朱…③李… 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 040654 号

责任编辑:谭文芳 特邀编辑:陈晓莉

出版者:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686045, 62237564

印刷者:北京瑞达方舟印务有限公司

发行者:新华书店总店北京发行所

开本:185×260 印张:13.25 字数:339 千字

版次:2004 年 7 月第 1 版 2004 年 7 月第 1 次印刷

书号:ISBN 7-81082-244-6/TP·121

印数:1~5 000 册 定价:19.00 元

前 言

动态网页制作技术是现今流行的一种页面制作技术。JSP 是 Sun 公司提出来的解决方案。JSP 蕴含了 Java 语言的优势,是一种真正跨平台实现动态功能的开发技术。JSP 既有较高的运行效率又有较短的开发周期,扩展能力和跨平台能力特别强,任何人都可以按照其公开的规范开发出自己的产品,鉴于 JSP 技术的明显优势,它已成为网页开发的主流。

本书围绕动态网页开发技术,详细介绍了 JSP 开发技术,内容涵盖 Java Servlets、JavaBean、JDBC 等重要技术;并针对所讲述的知识点提供了具有实用价值的实例,对实际开发应用具有很好的指导意义。本书叙述浅显易懂、循序渐进,从基础知识入手,真正让读者在初步掌握 JSP 编程的基础上获得对 Web 应用程序开发的整体认识和提高。

本书不仅可作为高等院校的计算机应用专业教学用书,也适合自学者及网页开发人员参考使用。

本书各章的编写者为:第 1 章、第 7 章、第 8 章由朱敏编写,第 2 章、第 3 章、第 4 章由朱晴婷编写,第 5 章、第 6 章由李媛媛编写。

编 者

2004 年 4 月于华东师范大学

目 录

第1章 JSP 简介	(1)
1.1 JSP 的开发背景及发展历史	(1)
1.2 JSP 与其他动态网页技术	(2)
1.2.1 JSP 与 CGI 比较	(2)
1.2.2 JSP 与 ASP 的比较	(2)
1.2.3 JSP 与 Servlet 的比较	(3)
1.3 JSP 的优点	(3)
1.4 建立 JSP 的运行环境	(4)
1.4.1 JDK 的安装	(4)
1.4.2 Tomcat 的安装	(5)
小结	(7)
习题	(7)
第2章 Java 语言基础	(8)
2.1 Java 语言简介	(8)
2.1.1 什么是 Java	(8)
2.1.2 Java 语言的特点	(9)
2.1.3 Java 与 C/C++ 的差异	(11)
2.2 数据类型和表达式	(12)
2.2.1 标识符和关键字	(12)
2.2.2 简单数据类型	(13)
2.2.3 运算符和表达式	(17)
2.3 流程控制语句	(18)
2.3.1 分支选择语句	(18)
2.3.2 循环语句	(22)
2.3.3 转向语句	(24)
2.4 类、数组和字符串	(27)
2.4.1 类和对象	(27)
2.4.2 数组	(31)
2.4.3 字符串	(34)
2.5 程序包和异常处理	(36)
2.5.1 程序包	(36)
2.5.2 异常	(37)
小结	(40)
习题	(40)

第 3 章 JSP 语法	(42)
3.1 JSP 基本语法	(42)
3.1.1 JSP 基本语法示例	(42)
3.1.2 注释	(44)
3.1.3 Escape 表示法	(46)
3.2 JSP 脚本元素	(47)
3.2.1 JSP 声明元素	(47)
3.2.2 JSP 表达式元素	(49)
3.2.3 JSP 脚本小程序元素	(50)
3.3 JSP 指令元素	(51)
3.3.1 page 指令元素	(51)
3.3.2 include 指令元素	(53)
3.3.3 taglib 指令元素	(54)
3.4 JSP 标准操作元素	(55)
3.4.1 include 操作	(55)
3.4.2 forward 操作	(59)
3.4.3 plugin 操作	(60)
3.4.4 useBean 操作	(62)
小结	(62)
习题	(63)
第 4 章 JSP 内置对象	(64)
4.1 request 对象	(64)
4.1.1 客户端使用 HTML 表单发送数据	(64)
4.1.2 获取客户端数据	(66)
4.1.3 HTML 表单控件	(69)
4.1.4 request 对象的常用方法	(78)
4.2 response 对象和 out 对象	(81)
4.2.1 response 对象	(81)
4.2.2 out 对象	(86)
4.3 application 对象和 session 对象	(86)
4.3.1 application 对象	(87)
4.3.2 session 对象	(89)
4.4 其他内置对象	(94)
4.4.1 config 对象	(94)
4.4.2 exception 对象	(94)
4.4.3 page 对象和 pageContext 对象	(94)
小结	(94)
习题	(95)
第 5 章 JSP 与 Servlets	(96)
5.1 Servlets 的基本概念	(96)

5.1.1	Servlets 的结构	(96)
5.1.2	Servlets 的接口	(97)
5.1.3	Servlets 的生命周期	(100)
5.2	JSP 和 Servlets	(101)
5.2.1	JSP 和 Servlets 的关系	(101)
5.2.2	JSP 和 Servlets 的区别	(101)
5.2.3	JSP 使用 Servlets	(102)
5.2.4	Servlets 访问 JSP	(103)
5.3	Servlets 的应用	(104)
5.3.1	Servlets 的开发	(104)
5.3.2	Servlets 的运行	(106)
小结		(115)
习题		(115)
第 6 章	JSP 与 JavaBean	(116)
6.1	JavaBean 原理	(116)
6.1.1	JavaBean 简介	(116)
6.1.2	JavaBean 的属性	(118)
6.1.3	JavaBean 的事件	(122)
6.2	JavaBean 的开发	(126)
6.3	在 JSP 中使用 JavaBean	(127)
6.3.1	在 JSP 中使用 JavaBean 相关标签	(128)
6.3.2	在 JSP 中使用 JavaBean 应用实例	(130)
6.4	Servlets 与 JavaBean	(132)
小结		(140)
习题		(140)
第 7 章	JSP 与 JDBC 数据库连接	(141)
7.1	JDBC 基础	(141)
7.1.1	JDBC 概要	(141)
7.1.2	数据库的连接	(142)
7.1.3	数据源	(142)
7.1.4	数据库操作	(145)
7.1.5	ResultSet 简介	(149)
7.2	用 JDBC 访问数据库	(153)
7.3	JSP 使用 JavaBean 操作数据库	(156)
7.4	访问数据库实现	(160)
7.4.1	检索数据分页处理	(160)
7.4.2	用 Hashtable 处理数据库操作	(163)
小结		(167)
习题		(167)

第 8 章 Web 应用开发和实现	(168)
8.1 JSP、Servlets 和 JavaBean 结合应用开发	(168)
8.1.1 创建数据库和数据源	(169)
8.1.2 JavaBean	(169)
8.1.3 Servlets 开发	(177)
8.1.4 JSP 开发	(180)
8.2 Web 应用实现	(197)
小结	(200)
习题	(200)
参考文献	(201)

第 1 章 JSP 简介

JSP (Java Server Pages, Java 服务器主页) 技术为创建显示动态生成内容的 Web 页面提供了一个简捷而快速的方法。JSP 技术的设计目的使得构造基于 Web 的应用程序更加容易和快捷, 这些应用程序能够与各种 Web 服务器、应用服务器、浏览器和开发工具共同工作。

1.1 JSP 的开发背景及发展历史

在万维网短暂的历史中, 它已经从一个大部分显示静态信息的网络演化到动态地对信息进行交互和操作的网络。在各种各样的应用程序中, 对于使用基于 Web 的客户端, 几乎没有任何限制。

基于浏览器客户端-服务器的应用程序比传统的基于客户机-服务器的应用程序有许多好处。这些好处包括几乎没有限制的客户端访问及极其简化的应用程序部署和管理。如要更新一个应用程序, 管理人员只需要更改一个基于服务器的程序, 而不需要更改成千上万的安装在客户端的应用程序。因此, 软件工业正迅速地向建造基于浏览器客户端-服务器的多层次应用程序迈进。

这些快速增长的基于 Web 的应用程序要求开发技术上的改进, 静态 HTML (Hypertext Markup Language, 超文本标记语言) 对于显示相对静态的内容是不错的选择; 新的挑战在于创建交互的基于 Web 的应用程序, 在这些程序中, 页面的内容是基于用户的请求或者系统的状态, 而不是预先定义的文字。

对于这个问题的一个早期解决方案是使用 CGI-BIN 接口; 开发人员编写与接口相关的单独程序, 以及基于 Web 的应用程序, 后者通过 Web 服务器来调用前者。这个方案存在严重的扩展性问题, 即每个新的 CGI (Common Gate Interface, 公共网关接口) 要求在服务器上新增一个进程。如果多个用户并发地访问该程序, 这些进程将消耗该 Web 服务器所有的可用资源, 并且系统性能降低到极其低下的地步。某些 Web 服务器供应商已经尝试通过为他们的服务器提供“插件”和 API (Application Programming Interface, 应用程序接口) 来简化 Web 应用程序的开发。这些解决方案与特定的 Web 服务器相关, 不能解决跨多个供应商的解决方案问题。例如, 微软的 ASP (Active Server Pages, 动态服务端页面) 技术使得在 Web 页面上创建动态内容更加容易, 但是它们也只能工作在微软的 IIS (Internet Information Server, Internet 信息服务器) 和 PWS (Personal Web Server, 个人 Web 服务器) 上。虽然还存在其他的解决方案, 但是都不能使一个普通的页面设计者能够轻松地掌握。例如, 像 Servlets 这样的技术就可以使得用 Java 语言编写交互应用程序的服务器端的代码变得容易。但采用这种方法, 整个网页必须都在 Servlets 中制作。如果开发人员或者 Web 管理人员想要调整页面显示, 就不得不编辑并重新编译该 Servlets。采用这种方法, 生成带有动态内容的页面仍然需要应用程序的开发技巧。

显然, 目前需要的是一个业界范围内的创建动态内容页面的解决方案。这个方案将解决当前方案所受到的限制, 即:

- ☞ 能够在任何 Web 应用程序服务器上运行;

- ✧ 将应用程序逻辑和页面显示分离;
- ✧ 能够快速地开发和测试;
- ✧ 简化开发基于 Web 的交互式应用程序的过程。

JSP 技术能够满足以上这些要求。JSP 规范是 Web 服务器、应用服务器、交易系统, 及开发工具供应商之间广泛合作的结果。Sun 公司开发出这个规范来整合和平衡已经存在的对 Java 编程环境 (例如, Servlets 和 JavaBeans) 进行支持的技术和工具, 其结果是产生了一种新的、开发基于 Web 的应用程序的方法, 给使用基于组件应用逻辑的页面设计者以强大的功能。

1.2 JSP 与其他动态网页技术

1.2.1 JSP 与 CGI 比较

JSP 与 CGI 最大的不同就是, 编写 JSP 就好像编写 HTML 页面, 只要了解简单的 JSP 语法就行了, 无需熟悉 Java 等脚本语言。而 CGI 程序基本上就是用脚本语言直接写的, 编写 CGI 程序, 要求对脚本语言十分熟悉。另外, 使用 CGI 程序时, 它对每一次访问都要重新处理, 这样在同时处理多个用户请求时, 总显得力不从心。而 JSP 只编译一次, 以后驻留于内存中, 再次调用时不必重新处理, 适于应付多用户情况。

1.2.2 JSP 与 ASP 的比较

ASP 是由 Microsoft 提出的动态网页解决方案。它可以完成产生动态网页、处理 HTML 表单、创建与数据库相结合的网页、跟踪用户会话和利用定制的组件等众多的功能。

JSP 和 ASP 有很多共同之处, 即, 它们都提供在 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功能。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 JSP 和 ASP 页面还需要附加语言引擎来分析和执行程序代码。分析和执行完毕后, 程序代码的执行结果以 HTML 代码形式发送给浏览器。JSP 和 ASP 都是面向 Web 服务器的技术, 客户端浏览器则不需要任何附加的软件支持。总而言之, 两者都为动态交互网页的制作提供了技术环境支持; 两者都能够为程序开发人员提供程序的编写与自带组件设计从逻辑上分离的技术; 两者都能够替代 CGI, 使网站的建设发展变得较为简单和快捷。

JSP 的明显优势如下。

- ✧ 平台和服务器的独立性。基于 Java 的 JSP 与平台无关, 它可以使用任何 Web 服务器 (包括 Apache、IIS 和 PWS), 而 ASP 技术主要依赖微软的技术支持, 如果要跨平台的话, 需要第三方 ASP 的引入产品。

- ✧ 使用的脚本语言。JSP 使用 Java 作为它的脚本语言, 其面向对象的编程思想较为优秀, 而 ASP 主要使用 VB、VBScript 和 JavaScript 等作为其脚本语言。

- ✧ JSP 跨平台的可重用性。JSP 组件 (Enterprise JavaBeans, JavaBeans) 都是跨平台可重用的。EJB (Enterprise JavaBeans, 企业级 JavaBeans) 组件可以访问传统的数据库, 并能以分布式系统模式工作于 UNIX 和 Windows 平台, 而 ASP 还无法完全实现一些企业级的功能。

☞ 多样化和功能强大的开发工具支持。Java 已经有了许多非常优秀的开发工具而且有许多可以免费获取，且其中有许多开发工具已经可以顺利地运行于多种平台之下。

1.2.3 JSP 与 Servlet 的比较

JSP 主要关注于 HTML（或者 XML（Extensible Markup Language，可扩展标记语言））与 Java 代码的结合，以及加入其中的 JSP 标记。如果一个支持 JSP 的服务器遇到一个 JSP 页面，它首先查看该页面是否被编译成为一个 Servlets。由此可见，JSP 被编译成 Servlets，即被转变为纯 Java，然后被装载入服务器执行。当然，这一过程，根据不同的 JSP 引擎而略有不同。

简单地说，Sun 公司首先发展出 Servlets，其功能比较强劲，体系设计也很先进，只是，它输出 HTML 语句还是采用了老的 CGI 方式，即逐句输出，所以，编写和修改 HTML 非常不方便。

后来 Sun 公司推出了类似于 ASP 的嵌套型网页开发技术 JSP，把 JSP 标记嵌套到 HTML 语句中，这样就大大简化和方便了网页的设计和修改。新型的网络语言如 ASP，PHP（Hypertext Preprocessor，超级文本预处理语言）都是嵌套型的。从企业级应用的 n 层结构角度看，一个企业级应用最少分三层：数据信息层、商业逻辑层和表示层。当然也可以更复杂。

Servlets 用来编写商业逻辑层的功能很强大，但是对于编写表示层就很不方便。JSP 则主要是为了方便编写表示层而设计的。当然也可以编写商业逻辑层。写惯了 ASP，PHP，CGI，经常会不自觉地表示层和商业逻辑层混在一起。比如把数据库处理信息放到 JSP 中，其实，它应该放在商业逻辑层中。

1.3 JSP 的优点

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 来设计页面的静态内容；使用 JSP 生成页面的动态内容。JSP 是服务器端技术。在服务器端，JSP 引擎解释 JSP，生成所有客户端请求的内容，然后将结果以 HTML 或者 XML 页面的形式发送回客户端。这有助于开发人员既保护自己的代码，又保证任何基于 HTML 的 Web 浏览器的完全可用性。

2. 强调可重用性组件

绝大多数 JSP 页面依赖于可重用的、跨平台的组件（JavaBeans 或者 Enterprise JavaBean）来执行应用程序所要求的更为复杂的处理。JavaBeans 已经是很成熟的技术，基于组件的方法可供开发人员共享，大大加速了总体开发过程。

3. 采用标识简化页面开发

JSP 技术封装了许多功能，这些功能是在生成易用的、与 JSP 相关的 HTML 或者 XML 的动态内容时所需要的。标准的 JSP 标识能够访问和实例化 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于编码的耗时功能。通过开发定制标识库，JSP 技术是可以扩展的。开发人员可以为常用功能创建自己的标识库，从而使 Web 页面开发人员

能够使用如同标识一样的工具来执行特定功能的构件。

4. 健壮性和安全性

由于 JSP 的内置脚本语言是基于 Java 语言的, 而且所有的 JSP 都被编译成 Servlets, JSP 就具有 Java 技术的所有好处, 包括健壮的储存管理和安全性。

5. 良好的移植性

作为 Java 平台的一部分, JSP 拥有 Java 编程语言“一次编写, 各处运行”的特点, 即具有良好的移植性。

6. 企业级的扩展性和性能

当与 J2EE (Java 2 Platform, Enterprise Edition, Java2 平台、企业级) 和 EJB 技术整合时, JSP 页面将提供企业级的扩展性和性能, 这对于在虚拟企业中部署基于 Web 的应用是必需的。

1.4 建立 JSP 的运行环境

由于 JSP 使用 Java 作为程序设计脚本语言, 因此需要建立 Java 的运行环境。编译和调试运行 Java 程序, 需要安装 JDK (Java Develop Kit, Java 开发工具包)。另外, JSP 是基于 Web 的 Java 应用程序, 因而它需要有特定的运行环境, 即解释器。由于 Java 语言是跨平台的, 所以能解释 Java 语言的 Web 服务器与平台无关。由 Apache 小组开发的 Tomcat 是一个开放源代码的自由软件, 可以自由获得而无需购买, 它是 Sun 公司推荐的产品, 并得到过 Sun 公司的鼎力帮助。

JSP 非常合适初学者, 以及在教学上使用。下面分别介绍 JDK 和 Tomcat 的安装。

本书所使用的平台是 Windows 2000。

1.4.1 JDK 的安装

1. 下载 JDK

JDK 是 Sun 公司制造的免费使用的软件包, 可以从 Sun 公司站点 <http://java.sun.com> 下载, 比较稳定的版本是 1.4.1, 在 Windows 平台下, 下载 j2sdk-1_4_0-win.exe 文件。

2. 安装 JDK

运行安装程序 j2sdk-1_4_0-win.exe, 选择安装路径后将会自动完成安装。

3. 设置环境变量

设置 Java 运行环境主要用到如下 3 个环境变量。

(1) JAVA_HOME

设定 Java 的安装路径。

(2) PATH

它是 Windows 所固有的, 需追加 JAVA_HOME\bin 目录, 这样在执行 JAVA_HOME\bin

文件夹下的命令时就不用输入长长的路径了, 如: javac、java 等命令。

(3) CLASSPATH

CLASSPATH 是运行 Java 非常重要的环境变量, Java 在编译和运行应用程序时都要通过它去找到需要的类文件。

设置环境变量的具体操作如下。

☞ 在桌面鼠标右键单击【我的电脑】, 在弹出菜单中选择【属性】, 单击【高级】, 如图 1-1 所示。单击【环境变量】, 出现如图 1-2 所示的对话框。

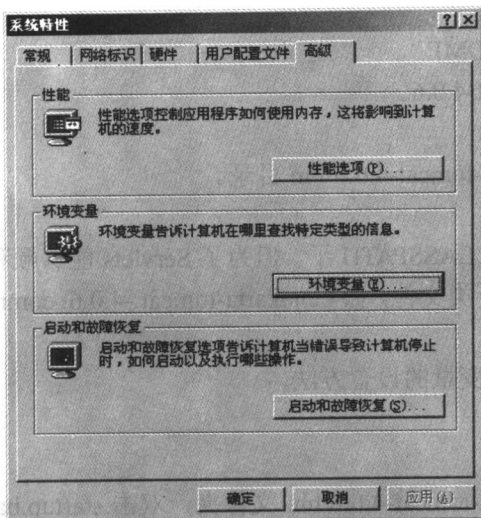


图 1-1 设置环境变量



图 1-2 添加环境变量

☞ 单击【系统变量】中的【新建】按钮, 打开如图 1-3 所示对话框, 添加 JAVA_HOME 环境变量, 并将其变量设置为 JDK 的安装目录, 如 C:\j2sdk1.4.0。

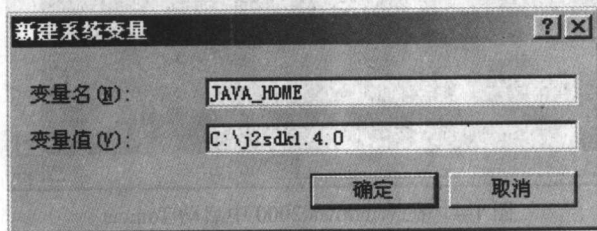


图 1-3 添加 JAVA_HOME 环境变量

☞ 分别将 JAVA_HOME\bin 目录, 如 C:\j2sdk1.4.0\bin 和 JAVA_HOME\lib\tools.jar, 如 C:\j2sdk1.4.0\lib\tools.jar 追加至环境变量 PATH 和 CLASSPATH。到此, Java 的运行环境安装设置完毕。

1.4.2 Tomcat 的安装

1. 下载 Tomcat

可以从 Apache 站点 <http://jakarta.apache.org> 下载, 虽然 Tomcat 已经有了新版本 5.0, 但在

此建议使用比较稳定的版本 Tomcat 4.0.6。下载 jakarta-tomcat-4.0.6.zip 文件。

2. 安装 Tomcat

Tomcat 的安装非常简单,只要解压 jakarta-tomcat-4.0.6.zip 文件即可,建议解压到盘区 \jakarta-tomcat-4.0.6 目录下。例如将文件 jakarta-tomcat-4.0.6.zip 解压到 C:\jakarta-tomcat-4.0.6 文件夹下。

3. 设置环境变量

(1) TOMCAT_HOME (或) CATALINA_HOME

设定 Tomcat 的安装路径,如 C:\jakarta-tomcat-4.0.6。

(2) PATH

追加 Tomcat 安装目录\bin 目录。如 C:\jakarta-tomcat-4.0.6\bin 目录。

(3) CLASSPATH

从 Tomcat 4.0 开始,已经不依赖环境变量的 CLASSPATH 了。但为了 Servlets 能够顺利编译,须设定 Tomcat 安装目录\common\lib\servlet.jar 类文件,如 C:\jakarta-tomcat-4.0.6\common\lib\servlet.jar 类文件。

环境变量的设置方法请参考 JDK 安装的环境变量的设置方法。

4. 运行 Tomcat

进入 Tomcat 安装目录\bin 目录,如 C:\jakarta-tomcat-4.0.6\bin 文件夹,双击 startup.bat。Tomcat 启动后如图 1-4 所示。

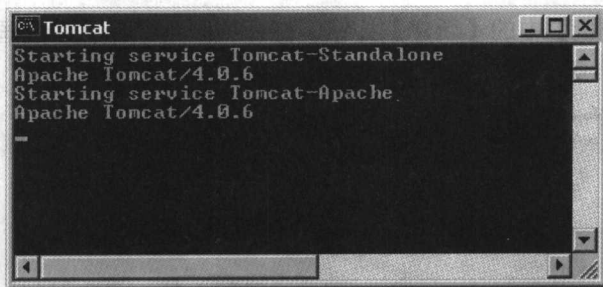


图 1-4 在 Windows 2000 中启动 Tomcat

5. 运行 JSP 程序

完成 Tomcat 的安装之后,在 Tomcat 的安装目录下已生成了 webapps 目录,且此目录下的子目录 ROOT 已在 Tomcat 环境设置文件“Tomcat 安装目录\conf\server.xml”作为浏览器访问 Web 应用程序的默认路径加以设置定义,在 server.xml 文件中,能找到<Context path="" docBase="ROOT" debug="0"/>这样的设置定义语句,因此, Tomcat 的安装完毕后,已生成了一个浏览器访问 Web 应用程序的默认路径,即: Tomcat 安装目录\webapps\ROOT。可以将 JSP 应用程序放在此目录下,通过在浏览器中打开http://localhost:8080/JSP 文件,运行 JSP 应用程序。

也可以创建自己的浏览器访问 Web 应用程序的路径,以方便开发应用。具体方法是:在

Tomcat 安装目录下, 在 conf 目录中的 server.xml 文件中查找到<Context>标签, 并在任何一对<Context>和</Context>标签后, 设置定义浏览器访问 Web 应用程序的路径, 如:

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

其中 path 的值是在浏览器中打开的、跟在http://localhost:8080 后的访问 Web 应用程序的路径, 称之为虚拟目录, docbase 的值是硬盘中的目录的实际路径, 即绝对路径, reloadable 的值定义了当应用程序中的类文件重新修改编译后, Tomcat 是否需要重新启动, 当值为 true 时, 不需要重新启动, 当值为 false 时, 需要重新启动。完成设置定义后, 重新启动 Tomcat, 这时可以通过在浏览器中打开http://localhost:8080/myApp/JSP 文件, 运行放在 E:\myApp 下的 JSP 应用程序。

小结

本章首先介绍 JSP 的开发背景及发展历史, 通过对 JSP 与其他动态网页技术的比较, 说明并总结了 JSP 的优点, 最后介绍如何建立 JSP 的运行环境。

习题

1. 要运行 JSP 程序, 必须使用_____这些工具和设备。
A. Web 浏览器 B. Java 开发工具包 C. Web 服务器 D. 客户端
2. JSP 的优点有_____。
A. 强调可重用性组件 B. 健壮性和安全性 C. 硬盘 D. 良好的移植性
3. JSP 的内置脚本语言是基于_____的。
4. 网络语言如 ASP, PHP, JSP 都是_____。
5. 安装 JSP 的运行环境。

第 2 章 Java 语言基础

JSP 技术是构建在 Java 语言基础上的, 并使用 Java 作为它的脚本语言, 它的很多特性和应用都来自 Java, 所以在学习 JSP 之前, 本章对 Java 语言进行介绍, 已经学习过 Java 语言的读者可以跳过本章, 直接进入下一章的学习。

2.1 Java 语言简介

2.1.1 什么是 Java

Java 环境可用来开发能在任何计算平台上运行的应用软件, 它实际上是一种非常基本且结构紧凑的技术, 它对 WWW (World Wide Web, 全球信息网) 以及商业的总体影响已可同电子表格对 PC 机的影响相比拟。

1. Java 的来源和发展

Java 来自于 Sun 公司的 Green 项目, Green 项目的任务是为了解决电器产品的数字系统共享信息的问题。日常生活中, 烤面包机、电冰箱、电视机、游戏机和其他大量的电器产品都包含越来越复杂的 CPU (Central Processing Unit, 中央处理器), 如何使这些产品能够一起工作、相互交流呢? 当传统的语言证明是不适用的时候, Java 语言就应运而生了。Java 能够提供给开发人员和用户简便性, 开发人员拥有广大的用户基础、平台独立性、低廉的开发费用和始终如一的执行环境, 对用户而言, 可以得到生动的内容、即时软件和增强的安全性。

Java 的真正普及是从 20 世纪 90 年代中期 WWW 的迅速崛起开始的, 它创造了“网络就是计算机”这句名言。它使 Web 页面栩栩如生, 而且它能够方便地编写和部署在任何一个网络、任何一个操作系统上运行的应用软件。Java 是一种网络计算的开发平台, 具有可缩放性, 它的应用可以从芯片组一直到大型计算机和服务, 这种平台可从电话一直扩展到极端重要的应用, 建立在企业现在的硬件和软件体系结构基础上, 可以创造出一个全部联网的企业。

2. Java 产品

简单地说, Java 是 Sun 公司开发的一种编程语言。但发展至今, 它得到了工业界广泛的支持, 已远不止是一种编程语言, 它的目标是构建企业的完整计算环境, 从开发工具、服务器软件和客户机一直到网络管理工具一套完整而全面的解决方案。它的主要组成部分如下。

- ☞ Java 开发工具和服务: 有助于简化开发 Java 应用软件的产品和服务。

- ☞ JavaOS: 一种结构紧凑的操作系统, 可使 Java 应用软件在网络计算机和蜂窝电话等设备上运行。

- ☞ Java 虚拟机: 一层软件, 嵌入 Windows, Macintosh 和 UNIX 等计算机操作系统, 或在这些操作系统上运行, 从而使计算机能运行 Java 应用软件。

- ☞ Java API: 标准软件界面, 位于 Java 应用软件和数据库检索及安全性等服务软件之间。

- ☞ Java 芯片：一系列为 Java 优化的伪处理器。
- ☞ Solstice WorkShop：Java 启动的网络管理工具。

2.1.2 Java 语言的特点

Java 是一个被广泛使用的网络编程语言，也是一种新的计算概念。作为一种编程语言，它简单、面向对象、不依赖于机器的结构，具有可移植性、健壮性、安全性，并且提供了并发的机制、具有很高的性能；其次，它最大限度地利用了网络，Java 的小应用程序（Applet）可在网络上传输而不受 CPU 和环境的限制；另外，Java 还提供了丰富的类库，使程序设计者可以很方便地建立自己的系统。

Java 的特点表现为以下几个方面。

1. 面向对象

面向对象是 Java 最重要的特性。面向对象的程序设计将程序实现的每一个具体功能作为类，然后由类生成对象。Java 提供简单的类机制及动态的接口模型，在对象中封装它的属性变量和操作，实现模块化和信息隐藏，而类则提供一组对象的抽象原型，并且通过继承机制支持静态和动态风格代码继承和重用。

2. 操作平台无关性

Java 语言突破了各种操作系统平台和不同机器的机器码限制的方法，定义出自己的一套虚拟机，以及在这套虚拟机上使用的机器码 Java Bytecode。Java 虚拟机是一种用软件方法实现的抽象化计算机，它是基于下层操作系统和硬件平台，执行 Java 的 Bytecode 程序。图 2-1 所示是一个 Java 程序从编译到执行的整个过程。

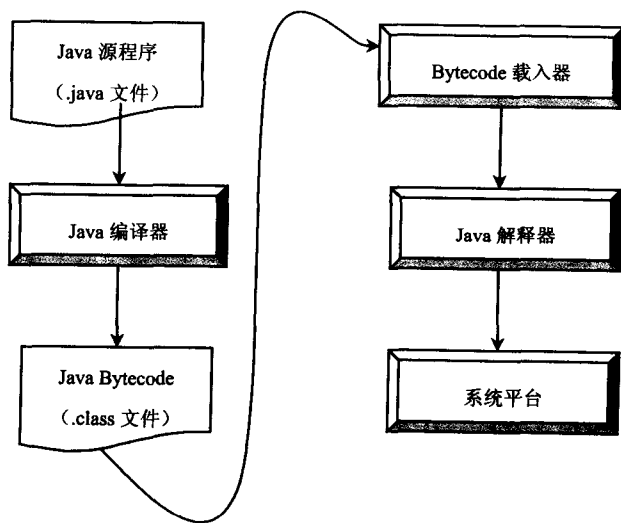


图 2-1 Java 程序的执行流程

一个编写好的 Java 源程序先要通过 Java 编译器编译，产生出 Java 虚拟机的机器码 Bytecode，再经过 Java 解释器将 Bytecode 转换成实际使用的机器和操作系统上的机器码去执