



普通高等学校计算机科学与技术应用型规划教材

Java Web应用开发教程

主 编 于 静

副主编 冯 瑶 张 虹

Java Web
YINGYONG KAIFA JIAOCHENG



北京邮电大学出版社
www.buptpress.com

普通高等学校计算机科学与技术应用型规划教材

Java Web 应用开发教程

主 编 于 静

副主编 冯 瑶 张 虹

北京邮电大学出版社

· 北 京 ·

内 容 简 介

本书针对应用型本科院校大规模发展、缺乏实用教材的现状,面向应用型本科计算机相关专业,系统介绍 Java Web 应用开发过程中的一些实用技术及系统设计与编程思想,并通过大量实战开发案例讲解 Java Web 开发技巧,同时根据作者的教学和实际开发经验,向读者提供详细的编程指南和 Web 应用程序开发中的注意事项,能够帮助读者快速上手,高效率地解决实际问题。

本书内容主要包括 Web 技术概述、Java 语言基础、HTML/JavaScript 基础、JSP 页面元素、JSP 内置对象、JDBC 数据库访问技术、JavaBean 组件技术、EL 表达式语言、JSTL 标签库、Servlet 技术、Web 设计模式、Web 2.0 与 Ajax。全书不仅介绍理论基础,更强调实际运用,特别注重提高读者运用 Java Web 相关解决实际问题的能力。

图书在版编目(CIP)数据

Java Web 应用开发教程/于静主编.--北京:北京邮电大学出版社,2010.4

ISBN 978-7-5635-2224-8

I. ①J… II. ①于… III. ①JAVA 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 064337 号

书 名: Java Web 应用开发教程

主 编: 于 静

责任编辑: 王丹丹 赵 旭

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号(邮编:100876)

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京忠信诚胶印厂

开 本: 787 mm×1 092 mm 1/16

印 张: 17

字 数: 398 千字

版 次: 2010 年 4 月第 1 版 2010 年 4 月第 1 次印刷

ISBN 978-7-5635-2224-8

定 价: 30.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

对于一名 Web 应用开发的初学者来说,主要欠缺两方面能力:一方面是科学的系统设计思想,另一方面是实践应用开发能力。本书以此为目的,以 Java 语言为基础,通过实际案例讲解 Java Web 应用开发领域中实际使用的相关技术,并最终帮助读者掌握基于 MVC 设计模式的分层系统设计思想。Java Web 是用 Java 技术来解决相关 Web 互联网领域的技术总和,包括服务器和客户端两部分。客户端的技术目前一般使用 HTML、JavaScript 等,服务器端的技术较为丰富,比如 Servlet、JSP、标签库等。

本书面对的读者对象主要包括普通高等学校应用型本科计算机相关专业的学生,综合学生特点和当前软件行业就业需求,编者总结实际教学经验与体会,设计各章节内容及教学重点,书中实例中蕴涵的思想大部分来自实际的工程项目经验,具有一定的实用性和参考性。

本书主要讲解 Web 技术概述、Java 程序设计基础、网页基础、JSP 页面元素、JSP 内置对象、JDBC 数据库访问技术、JavaBean 组件技术、EL 表达式语言与 JSTL 标签库、Servlet 技术、Web 设计模式、Web 2.0 与 Ajax 共 11 章内容。章节安排合理,内容循序渐近,建议安排 64 学时(32 学时理论、32 学时上机)来完成本课程。

第 1 章介绍 Web 应用的演化,介绍 JSP 的工作原理,说明搭建 Web 服务器和开发环境的步骤,介绍 MyEclipse 开发工具的使用,并对 HTTP 协议的特点做简要介绍。

第 2 章讲解 Java 程序设计基础,介绍 Java 中类和对象的概念,类的封装、继承及多态的特性,抽象类与接口,Java 异常处理机制,并着重讲解集合类的使用。

第 3 章学习网页 HTML 基础,包括 HTML 文档的基本结构,表格及表单的使用,重点讲解传统表格与 DIV+CSS 两种布局方法,并简要介绍了 JavaScript 脚本语言的客户端验证作用。

第 4 章介绍 JSP 页面元素,包括静态文本、注释元素、指令元素、脚本元素及动作元素,重点讲解实际中使用较多的 page 指令及表达式、脚本代码的用法。

第 5 章讨论 JSP 中 9 种内置对象,重点介绍输入输出对象:request、response 和 out 对象,作用域通信对象:session、application 和 pageContext 对象,以及其他对象的用法,并简要介绍 Cookie 的使用。

第 6 章讨论 JDBC 技术的使用,最终使读者能够采用 JDBC-ODBC 桥,第三方驱动等方式访问数据库,使用 Statement、PreparedStatement、CallableStatement 执行 SQL 语句。

第 7 章讲解 JavaBean 组件技术在 JSP 中的应用,介绍 JavaBean 类的典型代码结构,JSP 中使用 JavaBean 的动作元素,将 JavaBean 分为封装数据和封装业务逻辑两种,并以



实际案例学习如何使用 JavaBean。

第 8 章学习 JSP 中的 EL 表达式语言与 JSTL 标准标签库的使用,重点讲解 Core 核心标签库和 Format 标签库,并简要介绍自定义标签的创建方法。

第 9 章讨论 Java Web 的核心技术 Servlet,介绍如何创建 Servlet 以及 Servlet 的生命周期,讲解如何在 Servlet 中处理客户端请求并予以响应,如何会话管理进行状态跟踪,介绍以 Servlet 原理为基础的过滤器技术。

第 10 章在前面相关技术的基础上,讲解 Java Web 应用开发的两种模型,模型 1(JSP+JavaBean)和模型 2(JSP+Servlet+JavaBean),并针对同一案例进行对比分析,总结出基于 MVC 模式的模型 2 的优势所在。

第 11 章介绍 Web 2.0 的特点及应用,重点介绍 Ajax 技术的使用效果及程序框架,并给出一个 Ajax 应用案例。

本书第 1 章由顾鸿虹编写,第 2 章由冯瑶编写,第 3 章由张琦编写,第 6 章由张虹编写,第 4、5、7~11 章由于静编写。全书由于静负责内容结构设计和统稿工作。

对于书中的疏漏和不妥之处,恳请读者批评指正。

编 者

目 录

第 1 章 Web 技术概述	1
1.1 Web 应用演化	1
1.2 JSP 工作原理	3
1.2.1 Servlet 技术	3
1.2.2 JSP 生命周期	4
1.3 搭建 Web 服务器与开发环境	5
1.3.1 安装与配置 JDK	5
1.3.2 安装与配置 Tomcat	10
1.3.3 第一个 JSP 程序	10
1.3.4 安装与配置 MyEclipse 及其插件	11
1.4 超文本传输协议	18
1.4.1 HTTP 请求	19
1.4.2 HTTP 响应	19
1.5 本章小结	20
第 2 章 Java 程序设计基础	21
2.1 类和对象	21
2.1.1 类的成员	21
2.1.2 对象初始化与构造方法	22
2.1.3 this 关键字	23
2.1.4 包	24
2.2 类的继承	25
2.2.1 父类与子类	25
2.2.2 方法重写	27
2.2.3 super 关键字	28
2.2.4 访问修饰符	29
2.3 抽象类和接口	30
2.3.1 抽象类与抽象方法	31
2.3.2 定义与实现接口	32
2.4 异常处理	33
2.4.1 异常处理机制	33

2.4.2	使用 try-catch	34
2.4.3	多重 catch	34
2.4.4	使用 finally	35
2.4.5	throw 与 throws 关键字	36
2.5	使用集合类存储对象	37
2.5.1	List 集合	38
2.5.2	Set 集合	39
2.5.3	Map 集合	40
2.6	本章小结	41
第3章	网页基础	42
3.1	HTML 基础	42
3.1.1	HTML 文档基本结构	42
3.1.2	文字段落控制	44
3.1.3	表格	46
3.1.4	表单	49
3.2	布局基础	52
3.2.1	表格布局	52
3.2.2	DIV+CSS 布局	54
3.3	JavaScript 的简单应用	56
3.3.1	什么是 JavaScript	56
3.3.2	JavaScript 的事件处理	56
3.3.3	JavaScript 的数据类型和变量	57
3.3.4	JavaScript 的对象及其属性和方法	57
3.3.5	表单验证示例	58
3.4	本章小结	61
第4章	JSP 页面元素	62
4.1	JSP 页面基本结构	62
4.2	注释元素	64
4.3	指令元素	65
4.3.1	page 指令	65
4.3.2	include 指令	67
4.3.3	taglib 指令	68
4.4	脚本元素	69
4.4.1	声明	69
4.4.2	表达式	70
4.4.3	脚本代码	71
4.5	动作元素	74

4.5.1	<jsp:include>动作	74
4.5.2	<jsp:forward>动作	75
4.5.3	<jsp:param>动作	77
4.6	本章小结	78
第 5 章	JSP 内置对象	79
5.1	内置对象介绍	79
5.2	输入输出对象	82
5.2.1	request 对象	82
5.2.2	response 对象	90
5.2.3	out 对象	94
5.3	作用域通信对象	95
5.3.1	session 对象	95
5.3.2	application 对象	102
5.3.3	pageContext 对象	103
5.4	其他对象	106
5.4.1	page 对象	106
5.4.2	config 对象	107
5.4.3	exception 对象	108
5.5	Cookie 的使用	110
5.6	本章小结	112
第 6 章	JDBC	113
6.1	JDBC 概述	113
6.1.1	JDBC 基本概念	113
6.1.2	JDBC 驱动程序	114
6.2	java.sql 包	115
6.3	数据库操作	116
6.3.1	配置 ODBC 数据源	116
6.3.2	创建数据库连接	119
6.3.3	关闭数据库连接	120
6.3.4	Statement 类和 PreparedStatement 类	120
6.3.5	CallableStatement	124
6.3.6	ResultSet 结果集	126
6.4	一个 JDBC 实例	127
6.5	本章小结	133
第 7 章	JavaBean 在 JSP 中的应用	134
7.1	JavaBean 概述	134

7.1.1	JavaBean 组件技术	134
7.1.2	JavaBean 代码结构	134
7.2	在 JSP 中使用 JavaBean	136
7.3	JSP 动作与 JavaBean 生命周期	138
7.3.1	<jsp:useBean> 动作	139
7.3.2	<jsp:setProperty> 动作	140
7.3.3	<jsp:getProperty> 动作	140
7.4	封装数据的 JavaBean	142
7.4.1	使用 JavaBean 的表单交互	142
7.4.2	使用 JavaBean 的数据传参	143
7.5	封装业务逻辑的 JavaBean	146
7.6	JavaBean 其他应用	154
7.6.1	基于 JavaMail 的邮件发送 JavaBean	154
7.6.2	使用 JavaBean 实现数据分页显示	158
7.6.3	基于 JSPSmartUpload 的文件上传 JavaBean	163
7.6.4	基于 JGraph 的验证码 JavaBean	165
7.7	本章小结	168
第 8 章	EL 表达式语言与 JSTL 标签库	169
8.1	EL 表达式语言	169
8.1.1	EL 语法	169
8.1.2	EL 内置对象	171
8.2	JSTL 标签库	173
8.2.1	JSTL 简介	174
8.2.2	添加 JSTL 支持	175
8.3	Core 标签库	176
8.3.1	通用标签	177
8.3.2	条件标签	178
8.3.3	迭代标签	179
8.3.4	URL 相关标签	180
8.4	Format 标签库	182
8.5	SQL 标签库	186
8.6	XML 标签库	187
8.7	自定义标签库	189
8.7.1	自定义标签分类	190
8.7.2	创建自定义标签库	190
8.8	本章小结	197
第 9 章	基于 Servlet 的 Web 开发	198
9.1	Servlet 概述	198

9.1.1 JSP 与 Servlet	199
9.1.2 第一个 Servlet	199
9.1.3 Servlet 生命周期	201
9.2 处理客户请求与响应	203
9.2.1 处理客户表单数据	204
9.2.2 读取 HTTP 请求头信息	208
9.2.3 处理 HTTP 响应头信息	209
9.2.4 Servlet 通信	212
9.3 会话管理	215
9.3.1 会话状态概述	215
9.3.2 会话状态跟踪 API	216
9.4 过滤器	218
9.4.1 创建过滤器	219
9.4.2 解决请求数据中文乱码问题	223
9.5 本章小结	225
第 10 章 Web 设计模式	226
10.1 Java Web 应用开发的两种模型	226
10.1.1 模型 1:JSP+JavaBean	226
10.1.2 模型 2:JSP+Servlet+JavaBean	227
10.2 两种模型案例对比分析	227
10.2.1 问题描述与数据库设计	227
10.2.2 使用模型 1 实现	228
10.2.3 使用模型 2 实现	231
10.3 MVC 模式的优点	234
10.4 本章小结	235
第 11 章 Web 2.0 和 Ajax	236
11.1 Web 2.0 概述	236
11.1.1 Web 2.0 的应用	237
11.1.2 Web 2.0 的理论和技術	239
11.2 Ajax 简介	239
11.2.1 Ajax 定义	239
11.2.2 Ajax 工作原理	240
11.2.3 Ajax 核心技术	242
11.2.4 Ajax 程序框架	243
11.3 Ajax 应用实例	246
11.4 本章小结	258
参考文献	259

第 1 章 Web 技术概述



本章要点

- JSP 工作原理
- JSP 生命周期
- Web 服务器和开发环境的搭建
- 第 1 个 JSP 程序
- HTTP 协议的特点

随着互联网技术的不断发展,普通的静态网页已不能满足网上信息交流的需求,具有交互功能的动态网页得到了广泛的应用。Web 程序设计技术就是用于实现动态交互式功能的网页制作技术,通过 Web 程序语言(CGI、PHP、ASP、JSP、ASP. Net 等)设计的动态网页可以根据用户的即时操作和即时请求,网页内容发生相应地变化,从而可以实现功能强大的交互式操作。

本章简要介绍了 Web 应用的演变,以及 Servlet 和 JSP 的关系,详细讲解 JDK 的安装及部署过程,以及如何编写及运行第 1 个“HelloWorld”小程序,最后又介绍了客户端和服务器通信的 HTTP 协议的请求和响应。

1.1 Web 应用演化

设计动态网页的技术有很多,并且这方面的技术在不断地发展变化,它们有各自的特点。

1. CGI

CGI(Common Gateway Interface)是最古老的 Web 编程技术,是一段在服务器端运行的程序,是面向客户端 HTML 页面的接口,就像一座桥,把网页和 Web 服务器中的执行程序连接起来,把 HTML 接收的指令传递给服务器,把服务器执行的结果返回给 HTML 页面。

CGI 可以在任何服务器和操作系统上实现,任何程序语言都可以编写 CGI。CGI 属于底层操作,远不及 ASP、JSP 和 PHP 容易。因为 ASP 等都提供良好的运作环境,底层

操作对于它们而言是不必要的。如果把 CGI 比作机器语言,那么 ASP 等就是汇编语言了。但是由于 CGI 的底层操作,可以为大型系统创作之所用。

2. PHP

PHP(Personal Home Page)是一种 HTML 内嵌的语言,是在服务器端运行的嵌入 HTML 文档的脚本语言,语言的风格有类似于 C 语言,现在被很多的网站编程人员广泛地运用。PHP 独特的语法混合了 C、Java、Perl 以及 PHP 自创的新语法,它可以比 CGI 或 Perl 更快速地执行动态网页。

PHP 执行效率比较高,而且 PHP 支持几乎所有流行的数据库以及操作系统。但是每种数据库的开发语言都完全不同,这就需要开发人员将同样的数据操作作用不同的代码写出多种代码库,增加了程序员的工作量。另外 PHP 是完全免费的,所有代码都是开源的,缺少正规的商业支持,无法实现商品化应用的开发。

3. ASP

ASP(Active Server Pages)是微软推出的用以取代 CGI 的技术。简单来说,ASP 是一套服务器端的脚本运行环境,通过 ASP 可以结合 HTML 网页、ASP 指令和 ActiveX 元素建立动态、交互、高效的 Web 服务器应用程序。

ASP 提供了创建交互网页的简便方法,只要将一些简单的指令嵌入到 HTML 文件中,就可以从表单中收集数据。ASP 还可以利用 ADO(Active Data Object,微软开发的一种数据访问模型)方便地访问数据库,使得开发基于 WWW 的应用系统成为可能。

4. JSP

JSP(Java Server Pages)也是当前比较热门的 Web 技术,是由 SUN 公司发布的。JSP 为创建高度动态的 Web 应用提供了一个独特的开发环境。

JSP 与 Microsoft 的 ASP 技术非常相似。两者都提供了在 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。在 ASP 或 JSP 编程环境下,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 和 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 和 JSP 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。

JSP 的特点如下。

(1) 将内容的生成和显示进行分离。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。生成的内容逻辑上被封装在标识和 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,那么其他人员,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,并且将结果以 HTML(或者 XML)页面的形式发送回浏览器。这有助于作者保护自己的代码,而又保证了任何基于 HTML 的 Web 浏览器的完全可用性。

(2) 生成可重用的组件。绝大多数 JSP 页面依赖于可重用的、跨平台的组件

(JavaBean或者 Enterprise JavaBean 组件)来执行应用程序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。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。

(3) 采用标识简化页面开发。Web 页面开发人员不会都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 标识中,进行动态内容生成所需要的、标准的 JSP 标识能够访问的 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和耗时的功能。通过开发定制标识库,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如同标识一样的执行特定功能的构件来工作。

(4) 由于 JSP 页面的内置脚本语言是基于 Java 编程语言的,而且所有的 JSP 页面都被编译为 Java Servlet,JSP 页面就具有 Java 技术的所有好处,包括健壮的存储管理和安全性。

(5) 可靠且移植方便。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。随着越来越多的供应商将 JSP 支持添加到它们的产品中,可以使用自己所选择的服务器和工具,更改工具或服务器并不影响当前的应用。

1.2 JSP 工作原理

JSP 页面在运行的时候,首先会将整个 JSP 页面编译成一个 Java 文件,而服务器运行的时候,实际上是在运行 Java 的类文件,这样就达到了一次编写多处运行的目的。这个类文件就是 Java Servlet。

1.2.1 Servlet 技术

1. Servlet 简介

Servlet 是使用 Java Servlet API 编写的、适合于 B/S 模式的、运行在 Web 服务器端的 Java 类,具有独立于平台和协议的特性,可以生成动态的 Web 页面。Servlet 和客户端的通信采用“请求/响应”模式,其工作原理如图 1.1 所示。

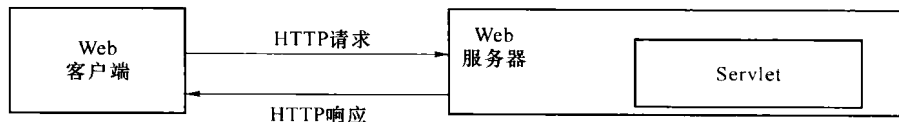


图 1.1 Servlet 的工作原理

2. Servlet 的优点

(1) 可移植性:Servlet 利用 Java 语言开发,具有 Java 的跨平台性,Servlet 程序可以在任何操作系统上运行。

(2) 功能强大:包括网络和 URL 访问、通过 JDBC 访问远程数据库、通过对象序列化

使用 JavaBean、通过 JNDI 使用 EJB、通过 JPA 访问 Web 服务等。

(3) 性能优良:Servlet 程序在加载执行之后,它的实例在一段时间内会一直驻留在服务器的内存中,若有请求,服务器会直接调用 Servlet 实例来服务。并且当多个客户请求一个 Servlet 时,服务器会为每个请求者启动一个线程来处理,所以效率高。

(4) 可靠性:Servlet 有强类型检查功能,并且利用 Java 的垃圾回收机制避免内存管理上的问题。另外,Servlet 能够安全地处理各种错误,不会因为发生程序上逻辑错误而导致整体服务器系统的崩溃。

3. JSP 与 Servlet 的关系

JSP 与 Servlet 的关系如图 1.2 所示,可以概括为两点:

(1) JSP 的实现是基于 Servlet 的,JSP 页面在运行之前要被解释成 Java Servlet。

(2) 当 JSP 容器接到对一个 JSP 页面的请求后,首先判断与 JSP 文件对应的 Servlet 类的名字,如果该类不存在或比 JSP 文件陈旧,容器就会重新创建一个等价的 Servlet 类并编译它。

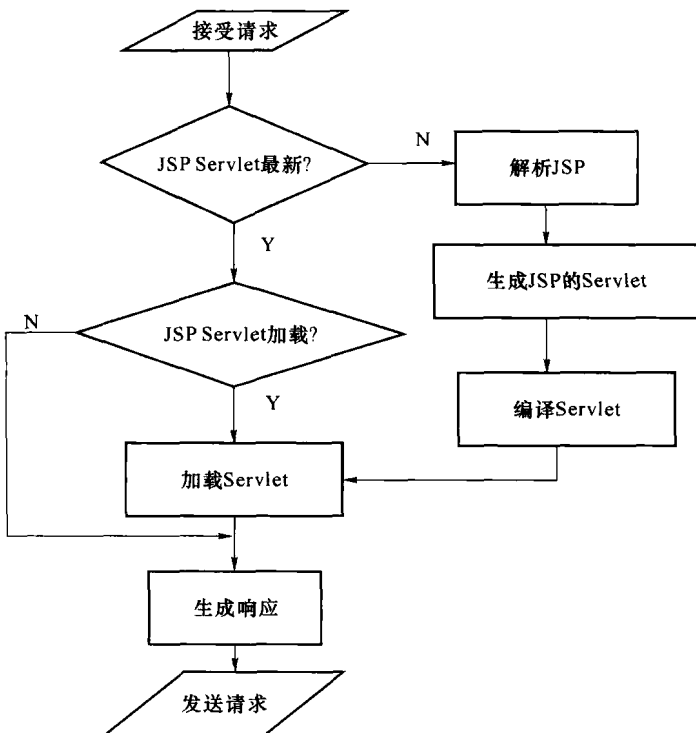


图 1.2 JSP 与 Servlet 的关系示意图

1.2.2 JSP 生命周期

如果客户端请求的是一个 JSP 页面,这时服务器会将该页面编译为一个 Servlet 类,并自动加载形成 Servlet 实例,并把执行结果返回到客户端。因此,JSP 网页在执行时会

经历以下几个阶段。

(1) 编译阶段

在客户端发出 Request 请求后, JSP 容器将 JSP 转译成 Servlet 的源代码, 再将产生的 Servlet 的源代码编译成为 Servlet 类。

(2) 初始化阶段

Servlet 容器加载 Servlet 类, 创建 Servlet 实例并调用 Servlet 的 init 方法进行初始化。

(3) 响应客户请求阶段

对于到达 Servlet 容器的客户请求, Servlet 容器创建特定于这个请求的对象, 然后调用 Servlet 的 service 方法处理请求信息, 响应至客户端。

(4) 终止阶段

当 Web 应用被终止, 或 Servlet 容器终止运行, 或 Servlet 容器重新装载 Servlet 的新实例时, Servlet 容器会先调用 Servlet 的 destroy 方法。在 destroy 方法中, 可以释放 Servlet 所占用的资源。

1.3 搭建 Web 服务器与开发环境

1.3.1 安装与配置 JDK

1. 安装 JDK

JDK(Java Development Kit)软件开发工具包是整个 Java 的核心, 包括了 Java 运行环境 JRE(Java Runtime Environment)、Java 工具及 Java 基础的类库等。JDK 可在 SUN 公司网站免费下载: <http://java.sun.com/javase/downloads/index.jsp>。进入页面, 选择正确的操作系统和语言版本后即可下载 JDK 的安装版(本书使用 jdk-6u1-windows-i586-p.exe)。

运行下载后的安装程序, 会弹出一个许可协议页面, 单击“接受”按钮即可进入下一步(注意: 本书中的程序均是运行在 Windows XP 系统下的)。

图 1.3 中显示了 JDK 所包含的组件: 开发工具主要包括 Java 语言的编译器, 开发专用 JRE 等; 演示程序及样例包括各种 Java 程序(应用程序以及 Applet)的源码; 源代码则是指 Java 核心类库(API)的源程序; 公共 JRE 则是 Java 程序的运行环境。选择好安装的目的文件夹后, 单击“下一步”按钮就可以继续安装所选组件了, 如图 1.4 所示。如果在图 1.3 的安装步骤中选择安装公共 JRE 组件, 则在安装 JDK 快结束时弹出 JRE 的安装界面, 如图 1.5 所示。

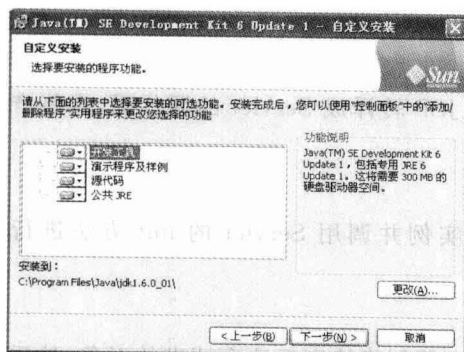


图 1.3 安装 Java SE JDK

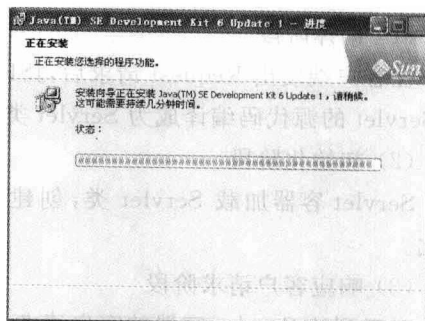


图 1.4 正在安装 Java SE JDK

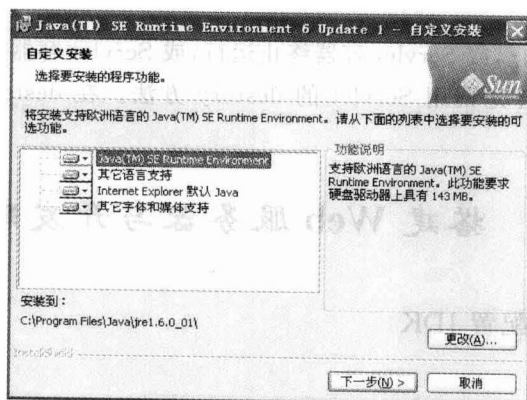


图 1.5 安装 JRE

图 1.5 中显示了 JRE 安装包中的组件,包括 JRE、浏览器默认的 Java 虚拟机、多语言支持和媒体支持等。单击“下一步”按钮后,完成 JRE 的安装,最终完成整个 JDK 的安装。安装成功后会显示如图 1.6 所示的成功信息。

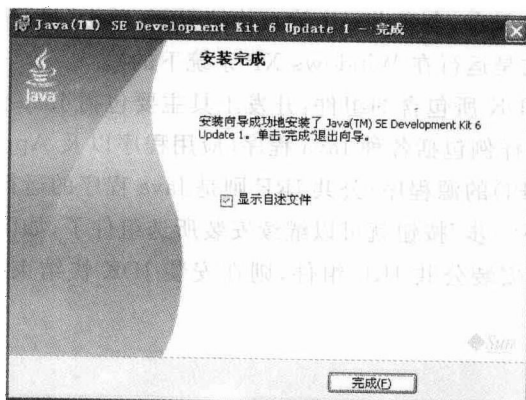


图 1.6 成功安装 Java SE JDK

安装结束后,在指定的 JDK 安装目录中应该至少包含 bin 文件夹和 jre 文件夹。bin 文件夹中主要存储了编译器(javac.exe)、调试器(jdb.exe)等开发工具。jre 文件夹下存储了 JDK 专用的 jre,其中也有一个 bin 文件夹,主要存储虚拟机(java.exe 和 javaw.exe)、Java Web Start(javaws.exe)等运行工具。

2. 配置 JDK

JDK 安装完成以后,为了在命令行状态中,任何路径下都能访问编译器和虚拟机,还需要设置一下环境变量。右击“我的电脑”,选择“属性”,在弹出的“系统属性”窗口中单击“高级”选项卡,单击最下面的“环境变量”按钮,弹出“环境变量”编辑窗口。

(1) 新建 JAVA_HOME 变量

在系统变量面板中单击“新建”,变量名输入“JAVA_HOME”(不区分大小写),变量值输入 JDK 的安装目录,单击“确定”按钮,如图 1.7 所示(设置此变量,便于以后运行 Tomcat、Eclipse 等)。

(2) 新建 ClassPath 变量

在系统变量面板中单击“新建”按钮,变量名输入“ClassPath”,变量值输入:;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar,单击“确定”按钮。

注意:%JAVA_HOME%前面的“.”是不能省略的,其中“.”用于表示在当前目录下,而“;”是各个部分的分隔符。

ClassPath 告诉编译器和虚拟机去哪里寻找需要的类库,只有类在 ClassPath 中,Java 命令才能识别。如果没有指定该变量的话,编译器和虚拟机会仅寻找两个位置,一个是当前工作的目录,另一个是 JDK 核心类库 rt.jar(在 jre\lib 目录中)。

(3) 编辑 Path 变量

单击系统变量中的 Path 变量,然后单击“编辑”,在变量值前面加入%JAVA_HOME%\bin;,如图 1.8 所示,即把 JDK 和 JRE 中的 bin 目录加入到 Path 中,同样用分号隔开。这样,就可以在任何目录下面访问虚拟机和编译器了。

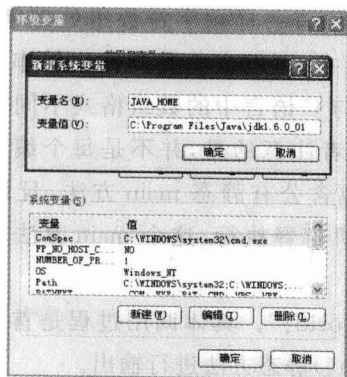


图 1.7 设置环境变量 JAVA_HOME



图 1.8 设置环境变量 Path