

Java 应用设计

宋 波 董晓梅 编著

于 戈 主审

人 民 邮 电 出 版 社

图书在版编目(CIP)数据

Java 应用设计/宋波, 董晓梅编著. —北京: 人民邮电出版社, 2002.8

ISBN 7-115-10475-1

I. J... II. ①宋...②董... III. JAVA 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 054127 号

内 容 提 要

本书从当前高等院校计算机专业教学的实际需要出发, 立足于 Internet 开发技术的前沿, 将面向对象的 Java 语言、JSP 技术、JDBC 技术、MySQL 网络数据库、实体关系图建模工具 (PowerDesigner)、UML 建模工具 (Poseidon for UML CE) 有机地结合起来, 构建了完整的基于浏览器/服务器/数据库的三层应用体系结构模式。

本书在知识体系的构成上, 以简明实用、便于学习、反映当今计算机技术最新发展方向和应用为特色, 着重突出实用性与应用性。

本书可以作为大学本科计算机相关专业的教材, 也可作为从事计算机软件开发的科技人员的参考书。

Java 应用设计

◆ 编 著 宋 波 董晓梅
主 审 于 戈
责任编辑 邹文波
执行编辑 苗 颖

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67180876
北京汉魂图文设计有限公司制作
人民邮电出版社内蒙古印刷厂印刷
新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 20.75

字数: 493 千字

2002 年 8 月第 1 版

印数: 1-5 000 册

2002 年 8 月内蒙第 1 次印刷

ISBN 7-115-10475-1/TP·2991

定价: 28.00 元(附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

目 录

第 1 部分 基础篇

第 1 章 Java 应用设计概述	1
1.1 Java 语言与 Internet	1
1.2 Java 应用设计的定义	2
1.2.1 Web 应用体系结构	2
1.2.2 基于 Java 技术的 Web 应用体系结构	3
1.2.3 Java 应用设计的定义	5
1.3 编译和运行 Java 程序	5
1.4 Java Application 程序	8
1.5 Java Applet 程序	9
1.6 Java 2 SDK 版本及主要应用领域	12
1.7 本章小结	13
思考与练习	14
第 2 章 Java 基本语法	15
2.1 基本数据类型、标识符及关键字	15
2.1.1 基本数据类型	15
2.1.2 标识符	16
2.1.3 关键字	17
2.2 常量	17
2.3 变量	18
2.4 运算符与表达式	19
2.4.1 赋值运算符与强制类型转换	19
2.4.2 算术运算符	20
2.4.3 关系运算符与逻辑运算符	21
2.4.4 其他运算符	21
2.4.5 运算符的优先级与结合性	22
2.5 条件控制语句	23
2.5.1 if 语句	23
2.5.2 switch 语句	24
2.6 循环控制语句	26
2.6.1 while 语句	27

2.6.2	do ... while 语句	28
2.6.3	for 语句	28
2.7	跳转控制语句	30
2.7.1	return 语句	30
2.7.2	break 语句	30
2.7.3	continue 语句	31
2.8	本章小结	33
	思考与练习	33
第 3 章	面向对象软件开发技术	35
3.1	面向对象的发展历史	35
3.2	面向对象技术的基本概念	35
3.3	面向对象的软件开发模型	41
3.3.1	瀑布模型	41
3.3.2	螺线模型	42
3.4	面向对象的软件开发方法	43
3.4.1	面向对象的分析	43
3.4.2	面向对象的设计	45
3.4.3	面向对象模型化技术	46
3.5	面向对象程序设计	47
3.6	本章小结	48
	思考与练习	48
第 4 章	Java 类与面向对象编程	49
4.1	定义和使用类	49
4.2	成员变量	54
4.3	方法	56
4.4	对象数组	60
4.5	本章小结	63
	思考与练习	63
第 5 章	Java 语言中的继承与多态	65
5.1	Java 语言中的继承	65
5.2	Java 语言中的多态	68
5.3	Java 语言中的包	74
5.4	Java 语言中的接口	75
5.5	本章小结	78
	思考与练习	79

第 6 章 Java 工具类与异常类	81
6.1 Applet 类与 Applet 小程序	81
6.1.1 Applet 基本工作原理	81
6.1.2 Applet 类	82
6.1.3 Applet 小程序应用实例	83
6.2 System 类	87
6.3 Math 与 Random 类	88
6.4 String 与 StringBuffer 类	89
6.5 Java 异常类	95
6.6 处理异常	97
6.6.1 捕捉异常	97
6.6.2 处理多异常	99
6.6.3 抛出异常	100
6.7 JAR 简介	103
6.8 本章小结	104
思考与练习	105
第 7 章 Java Web 开发技术概述	108
7.1 Web 服务器的基本概念	108
7.1.1 Web 服务器	108
7.1.2 超文本传输协议 HTTP	109
7.1.3 Apache 服务器	110
7.2 服务器端脚本语言概述	111
7.2.1 CGI	112
7.2.2 ASP	112
7.2.3 Servlet	113
7.2.4 JSP	119
7.3 JSP 执行环境	120
7.3.1 Tomcat 的安装与设置	120
7.3.2 部署 Web 站点	121
7.3.3 在 Apache 服务器环境下配置 Tomcat	124
7.4 本章小结	125
思考与练习	125
第 8 章 JSP 基本语法与隐式对象	126
8.1 简单的 JSP 应用实例	126
8.2 JSP 文档的运行原理与生命周期	128
8.3 JSP 文档的基本语法	130

8.4 JSP 隐式对象	133
8.4.1 请求/响应对象	133
8.4.2 表单和请求参数	134
8.4.3 隐式输出对象	136
8.4.4 session 对象	137
8.4.5 Application 对象	141
8.4.6 page 与 config 对象	142
8.4.7 pageContext 对象	142
8.5 本章小结	143
思考与练习	144
第 9 章 JSP 编译与操作指令	145
9.1 JSP 编译指令	145
9.1.1 page 编译指令	145
9.1.2 include 编译指令	150
9.1.3 taglib 编译指令	151
9.2 JSP 操作指令	152
9.2.1 <jsp:include>与<jsp:param>操作指令	152
9.2.2 <jsp:forward>操作指令	153
9.2.3 <jsp:plugin>操作指令	154
9.3 JSP 与 JavaBeans	156
9.3.1 <jsp:useBean>操作指令	157
9.3.2 <jsp:setProperty>操作指令	158
9.3.3 <jsp:getProperty>操作指令	158
9.3.4 JavaBeans 在 JSP 中的应用实例	158
9.4 本章小结	162
思考与练习	162

第 2 部分 数据库篇

第 10 章 MySQL 数据库基础	165
10.1 MySQL 概述	165
10.2 MySQL 的安装与管理	166
10.3 MySQL 数据目录的管理	170
10.4 创建与删除 MySQL 数据库	173
10.5 表的创建、修改、删除与插入	174
10.6 本章小结	182
思考与练习	182

第 11 章 SQL 语言	185
11.1 SELECT 命令	185
11.2 MySQL 常用函数	188
11.3 高级查询技术	191
11.4 处理查询结果	193
11.5 插入、修改和删除记录	195
11.6 MySQLGUI 工具简介	198
11.7 本章小结	200
思考与练习	200
第 12 章 JSP 数据库编程技术	201
12.1 JDBC 体系结构	201
12.2 JDBC 驱动程序	203
12.3 JDBC 数据库访问流程	206
12.4 JSP 与 JDBC API 程序设计	207
12.4.1 DriverManager 类	208
12.4.2 Connection 与 Statement 接口	209
12.4.3 ResultSet 接口	212
12.4.4 PreparedStatement 接口	215
12.4.5 DatabaseMetaData 接口	218
12.5 本章小结	220
思考与练习	220

第 3 部分 设计篇

第 13 章 数据流分析技术	221
13.1 数据流分析	221
13.2 数据流图	222
13.3 建立数据流图	224
13.4 数据字典	228
13.5 本章小结	231
思考与练习	232
第 14 章 用 ER 图进行数据库建模	233
14.1 CASE 技术概述	233
14.2 PowerDesigner 8.0 简介	234
14.3 DataArchitect 概述	235
14.4 DataArchitect 的工作环境	237

14.5 建立概念数据模型	238
14.6 生成物理数据模型	241
14.7 物理数据模型生成数据库	243
14.8 逆向工程	244
14.9 本章小结	245
思考与练习	246
第 15 章 统一建模语言 UML	247
15.1 UML 概述	247
15.2 UML 图表	249
15.2.1 类图	249
15.2.2 对象图	253
15.2.3 用例图	253
15.2.4 顺序图	254
15.2.5 合作图	255
15.2.6 状态图	257
15.2.7 活动图	257
15.2.8 组件图	259
15.2.9 配置图	259
15.2.10 模型管理视图	260
15.2.11 扩展组件	261
15.3 本章小结	263
思考与练习	264
第 16 章 UML 建模工具 Poseidon	269
16.1 Poseidon for UML CE 简介	269
16.1.1 基本概念和功能	270
16.1.2 模型的处理	275
16.1.3 高级功能	276
16.2 需求建模	277
16.2.1 系统需求	277
16.2.2 用例图	277
16.3 系统分析	278
16.4 系统设计	279
16.5 系统实现	283
16.6 代码生成	284
16.7 本章小结	288
思考与练习	289

第 17 章 Web 应用设计模式	290
17.1 Web 中心体系结构	290
17.1.1 Web 浏览器客户机	290
17.1.2 Web 服务器	291
17.1.3 应用程序服务器	291
17.1.4 数据库服务器	291
17.2 J2EE 设计模式	292
17.3 MVC 设计模式	293
17.3.1 只使用 JSP 的 MVC 设计模式	294
17.3.2 使用 Servlet 和 JSP 的 MVC 设计模式	295
17.4 MVC 设计模式案例	296
17.4.1 系统基本需求	296
17.4.2 模型设计	298
17.4.3 控制器设计	302
17.4.4 视图设计	303
17.5 Web 应用程序的规划和设计	305
17.6 本章小结	308
课程设计（实训）课题	309
附录 A 光盘内容清单	311
附录 B Java/Servlet/JSP/UML 资源列表	312
参考文献	313

第 1 部分 基础篇

第 1 章 Java 应用设计概述

Java 作为一种支持 Internet 应用开发的编程语言, 由于其独到的面向对象、跨平台、分布式应用等特点给程序员带来了一种崭新的计算概念, 使 Web 应用从最初的单纯提供静态信息, 发展到现在提供各种各样的动态服务, 从而使 Internet 发生了巨大的变化。

本章首先介绍 Java 语言与 Internet 的关系, 然后介绍 Web 应用的基础知识, 并引出“Java 应用设计”的概念, 从而使读者对“Java 应用设计”有一个概括性地了解。同时还介绍了 Java 程序设计的软件环境、执行流程、Java 2 SDK 的版本以及主要应用领域。

1.1 Java 语言与 Internet

Java 是 Sun 公司于 1995 年 6 月推出的新一代面向对象程序设计语言, 特别适合于 Internet 应用程序的开发。“连接 Internet, 用 Java 语言编程”, 已经成为 IT 专业人士的一种时尚。作为一种支持 Internet 应用开发的程序设计语言, Java 由于其独到的面向对象、跨平台、分布式、简捷性、健壮性、安全性等特点, 深受广大 Internet 应用开发人员的欢迎, 因而得到迅速的普及和广泛的应用。Java 作为软件开发的一种革命性的技术, 其地位已经被确定, 这主要表现在如下几个方面。

(1) 计算机产业的许多大公司购买了 Java 许可证, 包括 IBM、DEC、Adobe、Oracle, 以及 Sun 公司的竞争对手微软。这就说明 Java 已经得到了工业界的认可。

(2) 众多的软件开发商开始支持 Java 的软件产品。例如, Borland 公司开发的基于 Java 的快速应用程序开发环境 JBuilder, IBM 公司开发的 Visual Age for Java, Oracle 公司的 JDeveloper 9i 等。数据库厂商 Oracle、Sybase 等都在开发 CGI (Common Gateway Interface) 接口, 支持 HTML 和 Java。今天是以网络为中心的计算时代, 不支持 HTML 和 Java, 应用程序的应用范围就只能局限于相同的硬件平台。

(3) Intranet 正在成为企业信息系统最佳的解决方案, 而 Java 将在其中发挥不可替代的作用。Intranet 的目的是将 Internet 中部分实用技术用于企业内部的信息系统。它的优点主要是成本低、易于使用和管理。用户不管使用何种类型的机器和操作系统, 用户界面是统一的浏览器, 而数据库、Web 页面、应用程序 (Java Applet、Servlet 与 JSP) 则保存在 Web 服务器上。因此, 无论是开发人员还是管理人员, 以及用户本身都可以受益于这种解决方案。

2001 年 2 月 5 日, Sun 公司在美国举行了 Sun ONE (Sun Open Net Environment, 开放网络环境) 的新闻发布会, 更促进了 Java 技术的发展。Sun One 是 Sun 公司所提供的适用于开发的智能化 Web 服务的新一代架构, 支持跨越多网络的系统, 其中包括传统的 Web、无线 Web 和家庭网络。

Sun ONE 的宗旨是要保证使用任何工具, 开发智能 Web 服务, 使之运行于任何平台, 可以无缝地实现互操作。Sun ONE 体系结构基于一些开放和流行的标准、技术和协议, 例如, XML (Extensible Markup Language)、Java 和 HTTP 等。Sun ONE 的架构积聚了 Sun 公司的全部技术资源, 是 Sun 公司未来在软件应用领域称雄的法宝。

1.2 Java 应用设计的定义

浏览器以其使用的便利性、强大的表示能力, 以及迅速定位、获取信息的能力, 已经成为人类在日常生活中通过 Internet 获取各类信息的必备工具。浏览器还可以完成从遍及一个企业或跨越多个企业的不同系统的数据的收集和显示工作。因此, 现在大部分软件项目的开发都采用浏览器作为信息传输的用户接口。

1.2.1 Web 应用体系结构

最简单的 Web 应用体系结构, 是由一个浏览器和一个 Web 服务器组成。浏览器和服务器的可以位于一台机器上, 例如, Internet Explorer 和 Microsoft Personal Web Server 就通常一起运行于一台机器上进行 Web 应用程序的开发。但在实际的应用开发中, 浏览器与服务器通常位于不同的机器上。运行浏览器的机器称为 Web 客户机, 运行服务器的机器称为 Web 服务器。浏览器一般采用基于 TCP/IP (Transmission Control Protocol/Internet Protocol) 的网络连接与 Web 客户机相连, 这种连接可以是因特网 (Internet) 或者是内部网 (Intranet) 连接, 如图 1.1 所示。

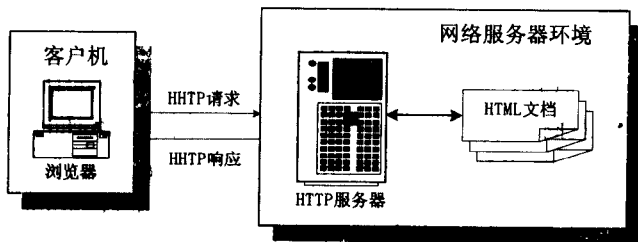


图 1.1 最简单的 Web 应用体系结构

1. 获取文档

当用户通过浏览器向 Web 服务器发出一个请求时, Web 服务器将接收请求, 查找文档, 并生成一个响应 (将文档传输给浏览器), 然后浏览器显示该文档。

用于传输 Web 网页的协议一般是超文本传输协议 (Hypertext Transfer Protocol, HTTP)。利用 HTTP, 当客户机的浏览器发出一个文档的 HTTP 请求时, 一个 Web 服务器将接收该请

求并通过发送一个 HTTP 响应做出应答。在响应成功的情况下, 客户机与服务器之间实现信息资源交换的步骤如下:

- (1) 连接: 客户机建立或打开与目标服务器的一个连接。
- (2) 请求: 客户机向服务器发送一个消息, 请求信息资源或服务。
- (3) 响应: 服务器向客户机发送返回的一个响应。
- (4) 断开: 关闭服务器和客户机之间的连接。

Web 服务器检索所请求的文件并将其作为响应的一部分加以返回。该文件一般是采用超文本标记语言 (Hypertext Markup Language, HTML) 构造的 Web 网页, 也可以是一个声音、影像以及简单的文本文件。

2. URL

Internet 上的资源或对象用统一资源定位器 (Uniform Resource Locator, URL) 进行描述, URL 是用于标识服务器所请求的资源的一种寻址机制。URL 的组成图 1.2 所示。

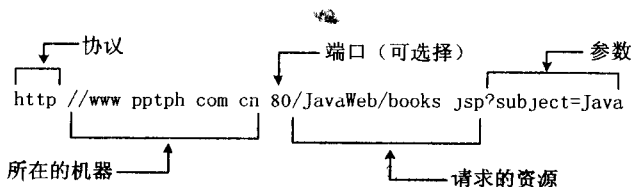


图 1.2 URL 组成示意图

(1) URL 的协议 (Protocol) 部分标识信息交换将使用的通信协议。一般比较常用的是超文本传输协议 (HTTP) 与文件传输协议 (File Transfer Protocol, FTP)。

(2) URL 的所在机器 (Machine 或 Host) 部分标识提供资源访问的机器的网络地址。机器可以输入互联网 (Internet Protocol, IP) 地址或机器名称, 但是机器名称最终要转换成一个对应的 IP 地址。这种转换可以用几种方式完成, 对于 Internet 来说, 通常使用一个域名服务 (Domain Name Service, DNS) 把名称转换为对应的 IP 地址。

(3) 可选择端口 (Port) 可以放置在机器部分的后边。端口是分配通信通道的 TCP/IP 方法。使用 TCP/IP 协议的每个网络服务将为想要连接到该服务的客户机监听特定的端口。服务包括 FTP, 简单邮件传输协议 (Simple Mail Transfer Protocol, SMTP), HTTP 等。如果不指定端口, 则表示使用默认端口, 如 HTTP 服务器的默认端口号是 80。

(4) URL 的请求的资源 (Resource) 部分标识想要请求的资源或文件的名称和路径。对于服务 HTML 文件的实例来说, 请求的资源就是 HTML 文件的名称。当在 URL 中只使用 Web 服务器域名而忽略目录和文件名时, 此时调用的文件名是 Index.html 或 Default.html。

(5) 参数 (Parameters) 部分为 HTTP URL 所特有, 参数与资源名之间由一个问号分隔。在一个 URL 中不允许使用空格。

1.2.2 基于 Java 技术的 Web 应用体系结构

图 1.1 所示的简单 Web 应用结构, 只能提供静态信息资源。虽然通过图像、声音等多媒体的表现方法, 使得传输的信息资源更为友好和丰富, 但是在客户机与服务器的信息交互方

面却无法给出适当的解决方案。

图 1.3 所示的基于 Java 技术的 Web 应用体系结构,是目前解决上述问题的一种比较好的解决方案。

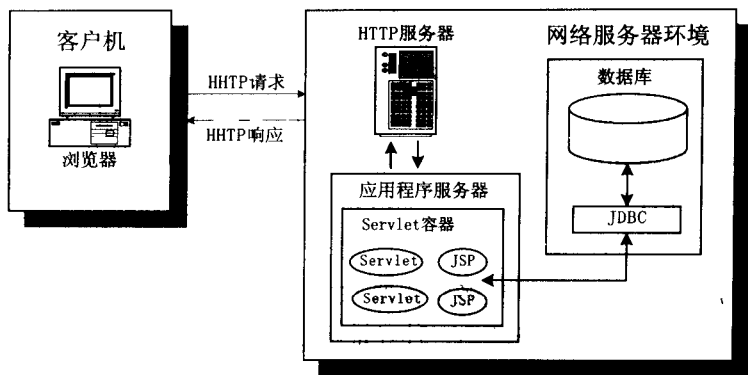


图 1.3 基于 Java 技术的 Web 应用体系结构

在图 1.3 中,增加了一个存放服务程序容器的应用程序服务器。当用户通过浏览器向 Web 服务器发出一个请求时,Web 服务器将接收请求,但并不马上返回一个文件,Web 服务器将通过应用程序服务器执行一个 Java 服务器端小程序 Servlet 并返回其输出,从而实现与客户机进行信息资源的交互。Servlet 是在一个服务程序容器或应用程序服务器下运行的 Java 程序。Apache 工程 (<http://www.apache.org>) 的 Tomcat 就是一个著名的 Java 服务程序应用服务器。我们将在第 7 章做详细的介绍。

在响应成功的情况下,客户机与服务器之间实现信息资源交换的步骤如下:

- (1) 连接——客户机建立或打开与目标服务器的一个连接。
- (2) 请求——客户机向 Web 服务器发送一个消息,请求信息资源或服务。
- (3) 调用——Web 服务器把该请求传递给应用程序服务器,后者接着调用 URL 标识的服务程序。

(4) 响应——Web 服务器向客户机发送返回的一个响应,其中包括应用程序所调用的服务程序的输出。

(5) 断开——关闭服务器和客户机之间的连接。

Web 服务器没有为每个请求生成一个新进程,而是把请求传递给已经在运行并等待的应用程序服务器,由其接着调用 URL 标识的服务程序的服务方法。也就是说,服务程序是随着应用程序服务器而持久存在。服务程序的初始化只有在应用程序服务器启动并调用服务程序初始化方法时才发生一次,这就使服务程序的服务方法可以集中用于快速生成响应的任务,而无需重新进行初始化。每次需要生成响应时,就在新线程里执行服务方法,大大提高了响应效率。

JSP (Java Server Pages, JSP) 也属于服务程序类型。JSP 允许 Java 代码嵌入到 HTML 文档中,通过使用服务器端 Java 小程序 Servlet 来提供 HTML 的动态生成。有关 JSP 的知识将在第 7、8、9 章做详细的介绍。

1.2.3 Java 应用设计的定义

Web 应用程序可以看成是:

Web 服务器+计算机网络+HTTP+浏览器

所组成的 Web 应用系统。

但是, Web 应用程序与通常的客户机/服务器系统有着显著的区别。其一是 Web 应用程序的部署, 部署 Web 应用程序是指建立网络的服务器构件, 而客户机不需要特别的软件或配置; 其二是客户机和服务器通信的本质, Web 应用程序的基本通信协议 HTTP 是一个无连接的协议, 它不是为最大的通信吞吐量而设计的, 是为强壮性和容错性而设计的。Web 应用程序中所有信息的传递都可以描述为 Web 页实体的请求和接收, 所以 Web 应用程序与 Web 网站的体系结构并没有太大的区别。

计算机辅助软件工程 (Computer Aided Software Engineering, CASE) 工具, 是 Web 应用系统开发的基础, 也是提高系统开发效率的一种有效辅助工具。数据流分析技术、数据库建模技术及工具、统一建模语言 (Unified Modeling Language, UML) 及设计工具, 是目前进行 Web 应用系统开发所采用的主要技术与工具。

Web 应用系统的设计模式, 就是系统开发的解决方案。设计模式是开发一个成功的 Web 应用系统的必要条件。

因此, 可以将“Java 应用设计”定义为:

Web 应用程序+CASE 工具+设计模式。

1.3 编译和运行 Java 程序

为了鼓励程序设计人员用 Java 语言开发应用软件, Java 语言的发明者 Sun 公司提供了免费的软件开发工具包 Java 2 SDK 1.3.1, 其中包括编译器、解释器、虚拟机、调试器等许多有用的开发工具。Microsoft Windows 版 Java 2 SDK 1.3.1 可以从 Sun 公司的站点 <http://java.sun.com/j2se1.3/> 上免费下载。本书所有的例题、习题均是在 Java 2 SDK 1.3.1 环境下编译执行的。

1. 安装和配置 Java 开发工具包

从 Sun 公司站点下载 Windows 版的 Java 2 SDK 1.3.1 软件包, 然后进行安装, 本书将它安装到 F:\jdk1.3.1 下。安装完 Java 2 SDK 1.3.1 软件包之后, 就会生成如图 1.4 所示的目录结构。

安装好之后, 还必须修改环境变量 PATH 和 CLASSPATH, 以便将其提供的开发工具所在的路径包含进去。在 Windows 95/98 操作系统下, 可以按以下方式修改 Autoexec.bat 文件中的环境变量 PATH 和 CLASSPATH:

```
PATH = %PATH%; F:\JDK1 3 1\bin
```

```
SET CLASSPATH=.; F\jdk1 3 1\lib\tools.jar; F\jdk1 3 1\lib\dt.jar; F\jdk1 3 1\user
```

在 Windows 2000 操作系统下, 可以在“控制面板 (Control Panel) → 系统控制台 → 高级 → 环境变量”中进行上述完全相同的修改。

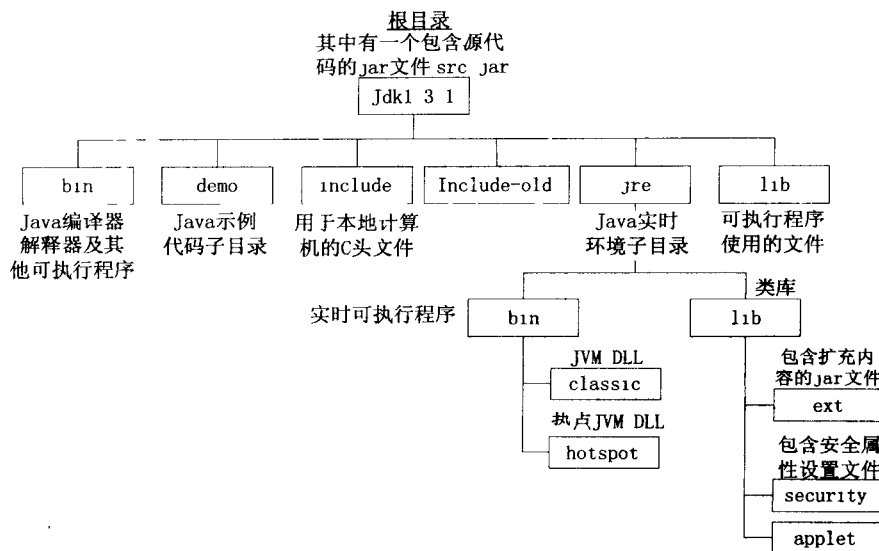


图 1.4 Java 2 SDK 1.3.1 的目录结构

重新启动 Windows 操作系统，在 MS-DOS 命令窗口中输入以下命令：

```
C\> PATH
```

如果环境变量 PATH 已经被修改，应该看到如下显示：

```
PATH=C\WINDOWS; C\WINDOWS\COMMAND; F:\JDK1.3.1\bin
```

在 MS-DOS 命令提示符下输入下列命令：

```
C\> java -version
```

如果出现下列提示信息，就说明已经成功地安装并配置好 Java 2 SDK 1.3.1 了。也就是说，我们可以编译和执行 Java 程序了。

```
java version "1.3.1"
```

```
Java(TM) 2 Runtime Environment, Standard Edition (build 1.3.1-b24)
```

```
Java HotSpot(TM) Client VM (build 1.3.1-b24, mixed mode)
```

其中，F:\jdk1.3.1 是作者建立的目录，本书全部实例的 Java 源程序的存放、编译、执行都是在这个目录下进行。读者如果需要建立自己的用于存放、编译、执行 Java 源程序的目录，也要在 CLASSPATH 环境变量中进行类似的设置。

2. 编辑 Java 源程序

Java 程序分为两种类型：Java Application 和 Java Applet。Java Application 必须在所处的操作系统下执行，而 Java Applet 必须与超文本标记语言 HTML 配合使用，由客户端的浏览器所包含的 Java 解释器来解释执行。

Java 编程可以分为编辑源程序、编译生成字节码和解释执行字节码几个步骤。Java 源程序是以 .java 为后缀的文本文件，可以使用各种 Java 集成开发环境中的源代码编辑器来编写，也可以使用 Windows 95/98/2000/NT 中的记事本或 MS-DOS 中的 EDIT 软件等编写。

3. 编译生成字节码文件

计算机高级语言程序从源代码到目标代码的生成过程称为编译。Java 源程序代码经过编

译所得到的目标代码称为字节码。字节码是二进制文件，由 Java 解释器来解释执行字节码。Java 2 SDK 1.3.1 提供的 Java 编译器程序为 `javac.exe`，Java 解释器程序为 `java.exe`。Java 程序在浏览器上下载及执行过程如图 1.5 所示。

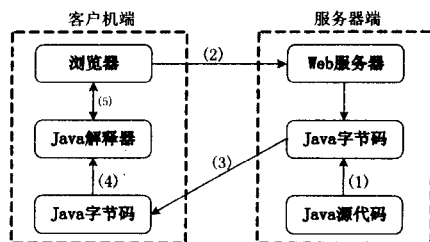


图 1.5 Java 程序的下载及执行过程

- (1) Java 编译器将 Java 源程序编译成字节码；
- (2) 客户机的浏览器与服务器进行连接，要求下载 Java 字节码文件；
- (3) 服务器将字节码文件通过网络传送给客户机；
- (4) 客户机上的 Java 解释器解释执行字节码；
- (5) 在浏览器上显示并与用户进行交互。

例如，要编译 Java 源程序 `MyClass1_1.java`，就要在 MS-DOS 窗口中键入如下命令：

```
F:\jdk1.3.1\user> javac MyClass1_1.java
```

这个命令的功能是调用 Java 2 SDK 1.3.1 软件包中 Java 编译器程序 `javac.exe`，检查源代码文件 `MyClass1_1.java` 中是否有语法错误并生成相应的字节码文件。如果有语法错误，Java 编译器就会在屏幕上输出这些错误所在的源代码的行号和错误的主要信息；否则编译成功并生成字节码文件 `MyClass1_1.class`。

4. 解释与运行字节码文件

Java 源代码编译生成的字节码文件不能直接运行在操作系统平台上，而必须运行在一个称为“Java 虚拟机 (Java Virtual Machine, JVM)”的、在操作系统之外的软件平台上。所谓“Java 虚拟机”就是负责解释并且执行字节码的程序，如 Java 2 中的 `java.exe` 程序。

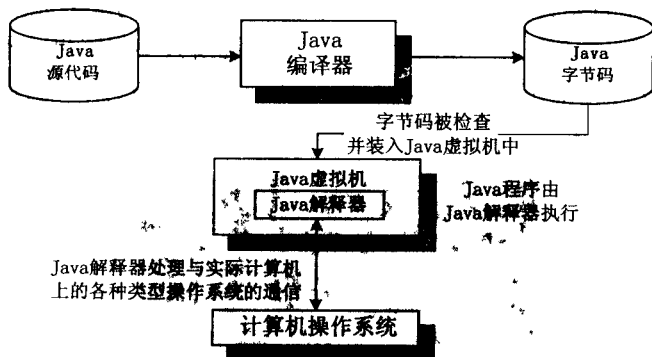


图 1.6 Java 语言的执行机制

图 1.6 所示的是 Java 语言的执行机制。在运行 Java 程序时，首先应该启动这个虚拟机，然后由它来负责解释执行 Java 的字节码。这样，利用 Java 虚拟机就可以把 Java 字节码程序

跟具体的软硬件平台分隔开来，只要在不同的计算机上安装上针对其特定平台特点的 Java 虚拟机，就可以把这种不同软硬件平台的具体差别隐藏起来，使得 Java 字节码程序在不同的计算机上能够面对相同的 Java 虚拟机，而不必考虑具体的平台差异，从而实现真正的二进制代码的跨平台可移植性。

当前 Internet 上使用的主要浏览器，在其内部都内嵌有 Java 虚拟机，因此，每当由 HTTP 服务器下载了 Java 字节码后，便将字节码的内容转交给虚拟机执行，然后将程序执行结果显示在网页上。无论是 Sun 公司的 Java 2 SDK 1.3.1 开发工具，还是网页浏览器，都有许多不同平台的版本，这样 Java 的跨平台能力才得以完全的实现。

1.4 Java Application 程序

Java Application 是完整的程序，需要独立的 Java 解释器来解释执行。在 Windows 95/98/2000/NT 操作系统下必须进入 MS-DOS 状态下才能执行。

【例 1.1】编译并运行下列 Java Application 源程序。

```

1: // MyClass1_1.java
2: import java.io.*;
3: public class MyClass1_1 {
4:     public static void main(String args[] ) {
5:         String s="Java 语言!";
6:         System.out.println("欢迎学习"+s);
7:     }
8: }
```

在包含 MyApplication.java 的目录中，执行以下命令：

```
javac MyClass1_1.java
```

如果编译成功，Java 编译器将 MyClass1_1.java 源代码翻译成字节码，并存储在一个新的名为 MyClass1_1.class 的类文件中。Java 源程序的文件名必须与类名完全一致。

需要注意的是，Java 语言是区分大小写的，不要混淆大小写字母。不改变当前目录执行 java MyClass1_1 命令，此时在 MS-DOS 命令窗口显示这样的文字：“欢迎学习 Java 语言！”，图 1.7 显示了这一执行结果。

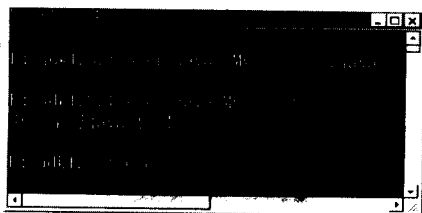


图 1.7 【例 1.1】的运行结果

■ 学习要点

(1) Java 源代码中的每一行前面的行号是为方便解释而附加的，在真正的 Java 程序中