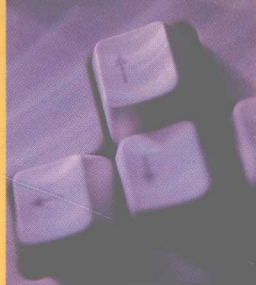




普通高等教育“十五”国家级规划教材



Java 编程及应用

杨武 刘贞 编



高等教育出版社

普通高等教育“十五”国家级规划教材

Java编程及应用

杨 武 刘 贞 编

高等教育出版社

内容提要

本书是普通高等教育“十五”国家级规划教材(高职高专教育),特点是:内容基础性强、图文并茂;在编写上尽量做到通俗易懂、方便学习,注重内容的编排和例题的选择。

本书重点强调 Java 的基础知识和应用实践以及用 Java 进行多媒体、图形界面、数据库和 Web 应用的程序设计,并注意对新、旧知识的融会与贯通,使读者能够尽快掌握 Java 的基础知识和应用。全书分为 14 章,比较全面系统地介绍了 Java 的运行环境及开发工具、基本编程方法、Java 面向对象程序设计的基本概念、继承与多态、异常处理、Java API、Web 服务器和 Applet 程序、用户图形界面设计、Java 多媒体程序设计、Java 数据库程序设计,Java Servlet 程序设计以及 Java Bean 组件等内容。

本书适用于高等职业学校、高等专科学校、成人高校、示范性软件职业技术学院、本科院校及举办的二级职业技术学院、继续教育学院以及民办高校使用,不仅可作为高等院校相关专业的教材,也可供从事 Java 开发、应用的人员学习参考。

图书在版编目(CIP)数据

Java 编程及应用/杨武,刘贞编. —北京:高等教育出版社,2004.2

ISBN 7-04-013903-0

I. J... II. ①杨... ②刘... 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(2003)第 123561 号

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.com.cn>

经 销 新华书店北京发行所
印 刷 北京市鑫霸印务有限公司

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

版 次 2004 年 2 月第 1 版
印 次 2004 年 2 月第 1 次印刷
定 价 18.50 元

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

为加强高职高专教育的教材建设工作,2000 年教育部高等教育司颁发了《关于加强高职高专教育教材建设的若干意见》(教高司[2000]19 号),提出了“力争经过 5 年的努力,编写、出版 500 本左右高职高专教育规划教材”的目标,并将高职高专教育规划教材的建设工作分为两步实施:先用 2 至 3 年时间,在继承原有教材建设成果的基础上,充分汲取近年来高职高专院校在探索培养高等技术应用性专门人才和教材建设方面取得的成功经验,解决好高职高专教育教材的有无问题;然后,再用 2 至 3 年的时间,在实施《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》立项研究的基础上,推出一批特色鲜明的高质量的高职高专教育教材。根据这一精神,有关院校和出版社从 2000 年秋季开始,积极组织编写和出版了一批“教育部高职高专规划教材”。这些高职高专规划教材是依据 1999 年教育部组织制定的《高职高专教育基础课程教学基本要求》(草案)和《高职高专教育专业人才培养目标及规格》(草案)编写的,随着这些教材的陆续出版,基本上解决了高职高专教材的有无问题,完成了教育部高职高专规划教材建设工作的第一步。

2002 年教育部确定了普通高等教育“十五”国家级教材规划选题,将高职高专教育规划教材纳入其中。“十五”国家级规划教材的建设将以“实施精品战略,抓好重点规划”为指导方针,重点抓好公共基础课、专业基础课和专业主干课教材的建设,特别要注意选择一部分原来基础较好的优秀教材进行修订使其逐步形成精品教材;同时还要扩大教材品种,实现教材系列配套,并处理好教材的统一性与多样化、基本教材与辅助教材、文字教材与软件教材的关系,在此基础上形成特色鲜明、一纲多本、优化配套的高职高专教育教材体系。

普通高等教育“十五”国家级规划教材(高职高专教育)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。

教育部高等教育司

2002 年 11 月 30 日

前言

Java 是目前 Internet 时代发展速度最快的软件开发工具之一，它采用面向对象思想和虚拟机技术，功能强大且简单易学，特别适用于 Internet 应用开发。由于 Java 内容丰富、庞杂，因此要完成一本涵盖 Java 基本内容和重点领域，且通俗易懂的 Java 应用程序设计教材绝不是一件容易的事。同时，如果不有所选择和限制的话，这些内容可能需要好几本书的篇幅来介绍。在教学实践中，我们发现国内一些较好的 Java 书籍，往往篇幅庞大，针对 Java 某一具体方面进行详细介绍，用于学习 Java 入门和常规应用程序设计存在一定困难；另有一些著名的外文翻译书籍，但在编排上又不太适应国内具体的教学情况。

那么，以怎样的方式编写一本能够让读者快速入门的 Java 应用程序设计教材呢？在充分考虑 IT 专业学生知识体系结构的基础上，确定了本书的基本框架和内容。本书重点强调 Java 的基础知识和应用实践，对一些其他课程介绍的内容不做过多的重复，但注意到了对新、旧知识的融会贯通，使读者能够尽快掌握 Java 的基础知识和应用方法。

本书除了介绍 Java 一般性知识外，还重点介绍了用 Java 进行网络+数据库的 Web 程序设计，这是 Web 应用的基础，也是 Java 的生命力所在。本书以简单易用的微软 SQL Server 为后台的网络数据库服务器，以 JDBC 为数据库连接接口，用 Java Servlet 进行 Web 服务器程序设计，使读者理解 Web 浏览器/Web 服务器/数据库服务器应用的一般模式，了解和掌握用 Java 开发 Web 应用的基本思路和方法。

本书在编写上尽量做到通俗易懂、方便学习，注意了内容的编排和例题的选择。书中对需要掌握的知识都举有实例程序，实例程序紧紧围绕学习过程中可能遇到的问题，并对执行结果和源程序进行了详细说明。每章都附有一定数量的练习题，帮助读者复习和巩固所学知识。为了便于读者阅读本书，在编写过程中，遵循了以下几项原则：

1. 每段理论知识后都有相应的详细解释，分为【语法说明】、【例题】、【运行结果】和【程序说明】等四部分。
2. 所有的例题都尽量与实际应用相结合，并且做到前后呼应，以便形成一个完整的知识理论体系。
3. 本书的所有例题均通过上机调试，运行正确无误。
4. 每个章节后面都有相关习题，这些习题是作者在 Java 题库的 1000 多道习题中精选出来的。建议读者在学习完相关章节后，选做相应习题并对上机题进行编程调试。

本书内容分为三部分，第一部分为 Java 入门，包括 1、2 章，主要介绍了 Java 的产生、历史、特点、应用背景、分类、应用范围、运行环境和开发工具。第二部分为 Java 语言基础，包括 3~7 章。3、4 章主要介绍了 Java 语言的基本特性及基本语法，包括 Java 语言概述、数据类型、运算符与表达式以及流程控制语句。5~7 章介绍了面向对象的程序设计技术，包括 Java 类与对象、继承与多态、接口与包、异常处理。通过该部分的学习，读者可以对面向对象思想和 Java 的面向对象程序设计特点有较为扎实的了解和掌握。第三部分即 Java 应用程序开发，包括 8~14

章,它是掌握好 Java 语言并进一步利用它解决实际问题的关键部分。其中:第 8 章介绍了 Java API,它是 Java 程序设计的基础;第 9 章介绍了 Web 服务器、HTML 表单和 Java Applet 小程序的基本知识,为进行 Web 应用提供准备;第 10、11 章介绍了 Java 用户图形界面和多媒体程序设计;第 12 章介绍了 Java 进行数据库应用的相关知识;第 13 章介绍了用 Java Servlet 进行一般 Web 服务器程序设计和 Web 数据库应用的方法;第 14 章简单介绍了 Java 组件程序设计 JavaBean。

采用本书教学,建议总学时数为 64,其中课堂教学为 40 学时,上机实验为 24 学时。具体建议学时数分配为:

章 节	建议课堂学时	建议实验学时	合 计
第 1 章	2		2
第 2 章	2	2	4
第 3 章	2	2	4
第 4 章	2	2	4
第 5 章	4	2	6
第 6 章	2	2	4
第 7 章	2	2	4
第 8 章	2	2	4
第 9 章	4	2	6
第 10 章	4	2	6
第 11 章	4	2	6
第 12 章	4	2	6
第 13 章	4	2	6
第 14 章	2		2
合 计	40	24	64

由于水平与时间的原因,书中不足之处在所难免,欢迎读者批评指正。

编 者

2003 年 9 月

目 录

第1章 Java 简介	1
1.1 Java 的发展历程	1
1.2 Java 语言的特点	2
1.3 Java 与 Internet	5
1.4 Java 程序的分类	6
1.5 Java 应用举例	8
本章小结	10
练习与思考	10
第2章 Java 运行环境及开发工具	
简介	11
2.1 Java 运行系统与开发流程	11
2.2 Java 的安装和配置	14
2.3 Java 开发工具包	15
2.4 Java 源文件编辑环境的选择与设置	17
2.5 编写并运行 Application 程序	18
2.6 编写并运行一个 Applet 小程序	21
本章小结	22
练习与思考	22
第3章 数据类型、变量与表达式	24
3.1 数据类型	24
3.2 标识符	26
3.3 关键字	27
3.4 常量	27
3.5 变量	28
3.6 运算符与表达式	29
3.6.1 赋值运算符与类型转换	29
3.6.2 算术运算符	30
3.6.3 关系运算符	32
3.6.4 逻辑运算符	33
3.6.5 位运算符	34
3.6.6 三目运算符和复杂运算符	34
3.6.7 运算符优先级	35
3.7 数组	36

3.7.1 一维数组	36
3.7.2 多维数组	38
本章小结	40
练习与思考	40
第4章 流程控制	42
4.1 选择(条件)控制	42
4.1.1 条件语句(if-else)	42
4.1.2 多分支语句(switch...case)	43
4.2 循环控制	46
4.2.1 for 语句	46
4.2.2 while 语句	46
4.2.3 do-while 语句	48
4.3 跳转控制	49
4.3.1 break 语句	49
4.3.2 continue 语句	50
4.3.3 return 语句	51
4.4 其他语句	52
本章小结	52
练习与思考	52
第5章 Java 面向对象程序设计的	
基本概念	53
5.1 Java 面向对象基础	53
5.1.1 类和对象	53
5.1.2 类的定义	54
5.1.3 类修饰符	55
5.2 对象创建和引用	57
5.2.1 对象的定义	57
5.2.2 对象成员变量的引用	58
5.2.3 对象方法的调用	58
5.3 成员变量	60
5.3.1 成员变量的定义	60
5.3.2 成员变量修饰符	61
5.4 方法	62

5.4.1 方法声明	62	8.3 Java.lang.Math 类	96
5.4.2 方法调用	62	8.4 java.util 类	98
5.4.3 方法参数的传递	63	8.4.1 日期类 Date	98
5.4.4 方法修饰符	64	8.4.2 随机数类 Random	100
本章小结	65	8.5 Java 输入输出类	101
练习与思考	65	8.5.1 输入流与输出流	102
第 6 章 继承与多态	66	8.5.2 文件输入输出	102
6.1 继承	66	8.5.3 缓冲区输入输出	106
6.1.1 子类创建 (extends 关键字)	66	本章小结	108
6.1.2 this 与 super 关键字	67	练习与思考	108
6.1.3 多重继承与接口	69	第 9 章 Web 服务器和 Applet 程序	109
6.2 接口	70	9.1 客户机/服务器结构	109
6.2.1 接口定义	70	9.1.1 Web 服务器	110
6.2.2 接口实现	70	9.1.2 统一资源定位符	110
6.3 多态	72	9.1.3 超文本传输协议 HTTP	110
6.3.1 方法覆盖	72	9.1.4 Tomcat 服务器	111
6.3.2 方法重载	74	9.2 HTML 表单设计基础	114
6.3.3 构造函数	74	9.2.1 Form 语法结构	114
6.4 包	75	9.2.2 Form 属性	114
6.4.1 包的定义	76	9.2.3 Form 输入元素	117
6.4.2 包的引用	76	9.3 Applet 的执行过程	120
本章小结	77	9.4 在页面中加入 Applet	121
练习与思考	77	9.5 传递参数给 Applet	123
第 7 章 异常处理	78	本章小结	125
7.1 异常的概念	78	练习与思考	125
7.2 异常处理机制	80	第 10 章 图形用户界面程序设计	126
7.3 异常处理方式	81	10.1 Java 图形用户界面编程基础	126
7.4 捕获异常	82	10.1.1 Java.awt	126
7.5 抛出异常	84	10.1.2 AWT 常见控制组件	127
7.6 自定义异常	86	10.1.3 布局管理器	130
本章小结	87	10.2 Swing 简介	133
练习与思考	88	10.3 事件处理机制	141
第 8 章 Java API 简介	89	10.3.1 KeyEvent	141
8.1 Java API 中的包	89	10.3.2 TextEvent	143
8.2 java.lang 包	90	10.3.3 ItemEvent	145
8.2.1 java.lang.System 类	90	10.3.4 MouseEvent	147
8.2.2 java.lang.String 类	91	本章小结	150
8.2.3 java.lang.StringBuffer 类	95	练习与思考	150

第 11 章 Java 多媒体程序设计	152	12.5.3 Connection 接口	175
11.1 多媒体技术概述	152	12.5.4 Statement 接口	176
11.2 Java 图形处理	152	12.5.5 PreparedStatement 接口	178
11.2.1 图形坐标系	153	12.5.6 ResultSet 接口	181
11.2.2 Graphics 类	153	本章小结	182
11.2.3 Color 类	154	练习与思考	183
11.2.4 文本与字体	156	第 13 章 Java Servlet 程序设计	184
11.3 图像处理	157	13.1 Java Servlet 工作原理	184
11.3.1 加载图像	157	13.1.1 Servlet 主要功能	184
11.3.2 显示图像	158	13.1.2 Servlet 的执行过程	184
11.3.3 图像生成	159	13.1.3 Java Servlet 与 CGI 的比较	185
11.3.4 图像的简单处理	161	13.1.4 Servlet 的运行环境	185
11.4 声音播放	162	13.1.5 Servlet 程序的两种基本 结构	185
11.5 动画技术	163	13.1.6 Servlet 的生命周期与基本 方法	187
本章小结	166	13.2 Servlet API	189
练习与思考	166	13.2.1 与 Servlet 有关类、接口的 关系	189
第 12 章 Java 数据库程序设计	167	13.2.2 Servlet 接口	190
12.1 JDBC 原理	167	13.2.3 GenericServlet 类	190
12.1.1 JDBC 的功能	167	13.2.4 HttpServlet 类	191
12.1.2 JDBC 的驱动程序管理器	168	13.2.5 ServletRequest 接口	193
12.1.3 ODBC 和 JDBC 的比较	168	13.2.6 ServletResponse 接口	193
12.2 JDBC 两层结构和三层结构	168	13.2.7 HttpServletRequest 接口	194
12.3 JDBC 驱动程序类型	169	13.2.8 HttpServletResponse 接口	195
12.3.1 JDBC-ODBC 桥	169	13.3 Servlet 程序设计实例	195
12.3.2 Java to Native API	170	13.4 Servlet 会话	197
12.3.3 Net Protocol All-Java	170	13.5 Servlet 数据库程序设计举例	200
12.3.4 Pure JDBC Driver	170	本章小结	205
12.3.5 建立 Book_Shop 数据库和 ODBC 数据源	171	练习与思考	205
12.4 JDBC 数据库访问流程	172	第 14 章 Java 组件程序设计	207
12.4.1 创建数据库连接	173	14.1 JavaBeans 概述	207
12.4.2 执行 SQL 语句	173	14.1.1 JavaBeans 的基本概念	207
12.4.3 接收并处理 SQL 的返回 结果	174	14.1.2 JavaBeans 的特点	208
12.4.4 关闭创建的各个对象	174	14.1.3 JavaBeans 和 EJB	208
12.5 JDBC 应用程序接口	174	14.2 JavaBeans 的属性	209
12.5.1 JDBC API	174	14.3 JavaBean 的事件	211
12.5.2 DriverManager 类	175		

14.4	BDK 安装与配置	212
14.4.1	BeanBox 的启动	212
14.4.2	在 BeanBox 中应用已建好的 Bean 组件	212
14.5	Bean 组件的创建	213

14.6 在 Applet 中使用 JavaBeans	217
组件	217
本章小结	217
练习与思考	218
参考文献	219
第 15 章 Java 数据库程序设计	221
15.1 JDBC 原理	221
15.2 JDBC 的组成	221
15.3 JDBC 的驱动程序管理器	221
15.4 JDBC 和 JDBC 驱动程序	221
15.5 JDBC 的 API	221
15.6 JDBC 驱动程序	221
15.7 JDBC 驱动程序	221
15.8 JDBC 驱动程序	221
15.9 JDBC 驱动程序	221
15.10 JDBC 驱动程序	221
15.11 JDBC 驱动程序	221
15.12 JDBC 驱动程序	221
15.13 JDBC 驱动程序	221
15.14 JDBC 驱动程序	221
15.15 JDBC 驱动程序	221
15.16 JDBC 驱动程序	221
15.17 JDBC 驱动程序	221
15.18 JDBC 驱动程序	221
15.19 JDBC 驱动程序	221
15.20 JDBC 驱动程序	221
15.21 JDBC 驱动程序	221
15.22 JDBC 驱动程序	221
15.23 JDBC 驱动程序	221
15.24 JDBC 驱动程序	221
15.25 JDBC 驱动程序	221
15.26 JDBC 驱动程序	221
15.27 JDBC 驱动程序	221
15.28 JDBC 驱动程序	221
15.29 JDBC 驱动程序	221
15.30 JDBC 驱动程序	221
15.31 JDBC 驱动程序	221
15.32 JDBC 驱动程序	221
15.33 JDBC 驱动程序	221
15.34 JDBC 驱动程序	221
15.35 JDBC 驱动程序	221
15.36 JDBC 驱动程序	221
15.37 JDBC 驱动程序	221
15.38 JDBC 驱动程序	221
15.39 JDBC 驱动程序	221
15.40 JDBC 驱动程序	221
15.41 JDBC 驱动程序	221
15.42 JDBC 驱动程序	221
15.43 JDBC 驱动程序	221
15.44 JDBC 驱动程序	221
15.45 JDBC 驱动程序	221
15.46 JDBC 驱动程序	221
15.47 JDBC 驱动程序	221
15.48 JDBC 驱动程序	221
15.49 JDBC 驱动程序	221
15.50 JDBC 驱动程序	221
15.51 JDBC 驱动程序	221
15.52 JDBC 驱动程序	221
15.53 JDBC 驱动程序	221
15.54 JDBC 驱动程序	221
15.55 JDBC 驱动程序	221
15.56 JDBC 驱动程序	221
15.57 JDBC 驱动程序	221
15.58 JDBC 驱动程序	221
15.59 JDBC 驱动程序	221
15.60 JDBC 驱动程序	221
15.61 JDBC 驱动程序	221
15.62 JDBC 驱动程序	221
15.63 JDBC 驱动程序	221
15.64 JDBC 驱动程序	221
15.65 JDBC 驱动程序	221
15.66 JDBC 驱动程序	221
15.67 JDBC 驱动程序	221
15.68 JDBC 驱动程序	221
15.69 JDBC 驱动程序	221
15.70 JDBC 驱动程序	221
15.71 JDBC 驱动程序	221
15.72 JDBC 驱动程序	221
15.73 JDBC 驱动程序	221
15.74 JDBC 驱动程序	221
15.75 JDBC 驱动程序	221
15.76 JDBC 驱动程序	221
15.77 JDBC 驱动程序	221
15.78 JDBC 驱动程序	221
15.79 JDBC 驱动程序	221
15.80 JDBC 驱动程序	221
15.81 JDBC 驱动程序	221
15.82 JDBC 驱动程序	221
15.83 JDBC 驱动程序	221
15.84 JDBC 驱动程序	221
15.85 JDBC 驱动程序	221
15.86 JDBC 驱动程序	221
15.87 JDBC 驱动程序	221
15.88 JDBC 驱动程序	221
15.89 JDBC 驱动程序	221
15.90 JDBC 驱动程序	221
15.91 JDBC 驱动程序	221
15.92 JDBC 驱动程序	221
15.93 JDBC 驱动程序	221
15.94 JDBC 驱动程序	221
15.95 JDBC 驱动程序	221
15.96 JDBC 驱动程序	221
15.97 JDBC 驱动程序	221
15.98 JDBC 驱动程序	221
15.99 JDBC 驱动程序	221
16.00 JDBC 驱动程序	221

第 1 章 Java 简介

学习目标

- 了解 Java 的发展历程及应用
- 理解 Java 语言的特点
- 掌握 Java 程序的分类

Java 是目前 Internet 时代发展速度最快的软件开发工具之一，由于它具有面向对象、跨平台、分布式应用、健壮性、安全性、多线程、高度可移植性等诸多优秀特点，因此得到了广泛的使用。同时，由于 Java 与 Internet 的紧密结合，它也是当今主要的网络应用开发工具。

本章主要介绍 Java 语言的发展过程、Java 的主要特点、与 Internet 的关系以及 Java 的应用，使读者对 Java 语言的基本特征有所了解。

1.1 Java 的发展历程

Java 的产生可以追溯到 20 世纪 90 年代，Sun 公司的 James Gosling 负责开发一个称为 Green 的项目。这个项目的用途是为家用电子消费产品开发一个分布式系统，以便对电冰箱、电视机、烤面包箱等家用电器进行控制，从而在它们之间可以进行信息交流。由于该系统涉及日用家电产品，这些产品对电子产品的安全性、可靠性、高效性等要求较高，并且涉及多处理器进行处理的功能，因此，要求这种语言必须是简单的和面向对象的。在项目实施之初，采用 C++ 来写嵌入式软件，但在实现过程中发现用 C++ 来实现该项目有诸多弊端，如 C++ 常有系统失效性程序错误，尤其内存管理要求设计人员必须记录并管理内存资源，这都会成为安全性问题的隐患。在这种情况下，Gosling 最后决定在 C++ 语言的基础上开发一种新型语言 Oak。Oak 是一种用于网络的精巧而安全的语言，它在保留了大部分与 C++ 相似语法的同时，对 C++ 语言中存在的弊端进行了改进，比如我们上面所提到的内存管理。

Sun 公司曾经用 Oak 对一个交互式电视项目进行投标，但被 SGI 打败。这时 Web 浏览器 Mosaic 启发了 Oak 项目组成员，由于 Oak 独立于平台、可靠性高、安全性好等特性，非常适合 WWW，因此他们把 Oak 作为一种软件程序语言，编制了 HotJava 浏览器。后来，因为 Oak 这个商标已被注册，项目组成员把 Oak 重新命名 Java。在 SunWord95 中发布出来，并得到了 Sun 公司首席执行官 Scott McNealy 的支持，从此以后，Java 进军 Internet，并随着软件的快速发展而成为程序设计语言家族中的一颗璀璨明珠。

Java 的出现给软件业带来了巨大的冲击。1994 年下半年，Internet 的迅猛发展，WWW 的快速增长，促进了 Java 语言研制的进展，使得它逐渐成为 Internet 上最受欢迎的开发与编程语言之一。一些著名的计算机公司纷纷购买了 Java 语言的使用权，如 Microsoft、IBM、Netscape、Novell、Apple、DEC、SGI 等，因此，Java 语言被美国的著名杂志 PC Magazine 评为 1995 年十大优秀科

技产品之一，随之大量出现了用 Java 编写的软件产品，受到工业界的重视与好评，认为“Java 是 20 世纪 80 年代以来计算机界的一件大事”。微软总裁比尔·盖茨在悄悄地观察了一段时间后，不无感慨地说：“Java 是长时间以来最卓越的程序设计语言”，并确定微软整个软件开发的战略从 PC 单机时代转向以网络为中心的计算机时代。Java 的诞生对整个计算机产业产生了深远的影响，对传统的计算模型提出了新的挑战。Sun 公司总裁 Scott McNealy 认为 Java 为 Internet 和 WWW 开辟了一个崭新的时代。Microsoft 和 IBM 两大公司都计划在 Internet 上销售用 Java 编写的软件。Apple、HP、IBM、Microsoft、Novell、SGI、SCO、Tandem 等公司均计划将 Java 并入各自开发的操作系统，而负责开发并推广 Java 技术的 SunSoft 公司（这是 Sun 下属的一个子公司），将通过颁发许可证的办法来允许各家公司把 Java 虚拟机和 Java 的 Applets 类库嵌入他们开发的操作系统，这样各类开发人员就能更容易地选择多种平台来使用 Java 语言编程，不同的用户也就可以脱离 Web 浏览器来运行 Java 应用程序，这无疑是很受广大用户欢迎的，也为 Java 语言的应用开拓了极为广阔的前景。

Java 语法是从 C++ 继承过来的，它的对象模型源自 C++，因此它与 C 和 C++ 有着直接的联系。现在，Java 与 C 和 C++ 的这种联系对编程人员变得更加便利，因为许多熟悉 C/C++ 语法的编程人员，可以很容易对 Java 进行学习；从另一方面来看，熟悉 Java 的编程人员也可以轻松地学习 C/C++。这里特别说明一点，尽管 Java 受 C++ 的影响，但是它并不是 C++ 的增强版，它与 C++ 既不是向上兼容，也不向下兼容。Java 并不是用来取代 C++，它是为了解决一系列特定问题而设计，而 C++ 是为解决另外一系列问题而设计的，两者将并存很长一段时间，读者在今后的学习中将会逐渐体会到这一点。

1.2 Java 语言的特点

由于 Java 是一种定位于网络应用的软件开发工具，因此，Java 的特点也是紧紧围绕这一中心展开的。另外，Java 语言也充分利用了当代软件技术的最新成果。Java 性能优异，其主要特点集中在简单易学、面向对象、分布式、解释性、健壮性、安全性、平台无关性、可移植性、高性能、多线程、动态等多方面，下面简单介绍 Java 的主要特点。

1. 简单易学

Java 最初是为对家用电器进行集中控制而设计的，因此它必须简单明了。Java 的简单易学主要体现在以下三个方面：① Java 的风格类似于 C++，因而 C++ 程序员是非常熟悉的。从某种意义上讲，Java 语言是 C 及 C++ 语言的一个变种，因此，C/C++ 程序员可以很快掌握 Java 编程技术。② Java 摒弃了 C/C++ 中容易引发程序错误的、不易理解和掌握的部分，如指针操作、结构类型、运算符和内存管理。③ Java 语言对计算机的硬件环境要求很低。用 Java 编制的程序，可以在内存很小的计算机独立运行。

2. 面向对象

面向对象可以说是 Java 最重要的特性。面向对象技术的核心是，以更接近于人类思维的方式建立解决问题的逻辑模型，它利用类和对象机制将数据及其操作封装在一起，并通过统一的接口与外界交互，使反映现实世界实体的各个类在程序中能够独立及继承。通常我们所说的面向对象的特性一般要包括封装性、多态性、继承性、动态性。我们所使用的面向对象的语言，

很多都不是完全面向对象的，只是部分地用到了面向对象的技术。Java 是完全面向对象的，它不支持类似 C 语言那样的面向过程的程序设计技术，并且做到了 C++ 所不具备的动态链接功能。Java 支持静态和动态风格的代码继承及重用。单从面向对象的特性来看，Java 类似于 SmallTalk，但其他特性、尤其是适用于分布式计算环境的特性远远超越了 SmallTalk。编写 Java 程序的过程就是设计、实现类，定义类的属性、行为的过程。

3. 分布式

分布式计算指的是几部计算机通过网络同时协同工作。Java 在网络程序设计上极为优异，用 Java 来写网络程序就好像只是从一个文件调用或存入数据。Java 包含一个支持 HTTP、FTP 等基于 TCP/IP 协议的子库。因此，Java 应用程序可凭借 URL 打开并访问网络上的对象，其访问方式与访问本地文件系统几乎完全相同。在分布式环境中尤其是在 Internet 上实现动态内容无疑是一项非常宏伟的任务，但 Java 的语言特性却使我们很容易地实现这项目标。

4. 健壮性

Java 虚拟机检查程序在编译和运行时的错误，类型检查会帮助查出许多编程时产生的错误。Java 自己操纵内存减少了内存出错的可能性。Java 还实现了真数组，避免了覆盖数据的可能，这种功能特征大大缩短了开发 Java 应用程序的周期。

5. 平台无关性

为实现平台无关性，Java 采用了半编译、半解释的策略，使用 Java 虚拟机（Java Virtual Machine）技术。Java 将它的程序编译成一种结构中立的中间文件格式，只要安装有 Java 虚拟机的机器都能执行这种中间代码。Java 语言的平台无关性的体现如图 1.1 所示。目前，Java 运行系统可运行在 Windows98/NT/2000、Linux、Solaris2.4 (SPARC)、Apple Macintosh 等平台上。Java 源程序被编译成一种高层次的与具体机器无关的字节码（Byte Code），这种字节码在具体机器的虚拟机上解释、运行。

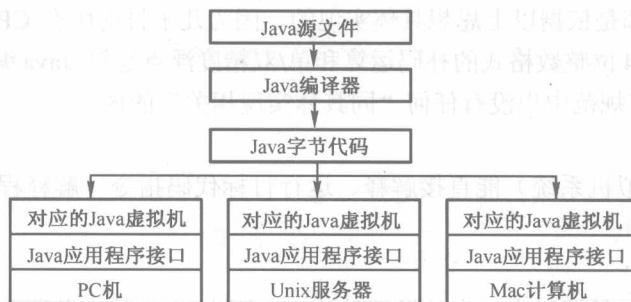


图 1.1 Java 语言的平台无关性

6. 安全性

Java 的安全性在 Java 设计时就得到了考虑。首先，在 Java 中，像 C/C++ 中的指针和内存释放等功能被取消，从根本上避免了程序产生的非法内存操作。另外，当 Java 程序嵌入在网页内时，Java 程序的功能受到严格限制，以确保整个系统的安全。Java 程序在执行前，要经过三次检查，分别是 Java 本身的代码检查、对字节码（Byte Code）的检查以及程序执行系统也即 Java 解释器的检查，如图 1.2 所示。

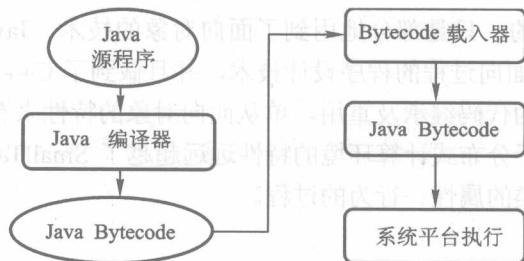


图 1.2 Java 对安全维护的三次检查

(1) 代码检查包括：检查代码段的格式，检查指针操作，检查是否试图改变一个对象的类型等。

(2) 字节码检查包括：检查代码有无引起堆栈溢出，检查所有操作代码参数类型是否都是正确的，检查是否发生非法数据转换（如将整数转换成指针），检查访问对象操作是否合法。

(3) 程序执行系统的检查：类装载（ClassLoader）通过将本机类与网络资源类的名称分开，来保持安全性。因为调入类时总要经过检查，这样避免了特洛伊木马现象的出现。从网络上下载类被调进一个与源相关的私有的名字域。当一个私有类访问另一个类时，本机类（build-in）首先被检查，然后检查相关的类，这样就避免了本机类被破坏的情况出现。

7. 可移植性

同具体平台无关的特性使得 Java 应用程序可以在安装了 Java 解释器和运行环境的任何计算机系统上运行，这成为 Java 应用软件移植的良好基础。但仅仅如此还不够。如果基本数据类型设计依赖于具体实现，也将为程序的移植带来很大不便。例如在 Windows98 中整数为 32 bit，而在 DEC Alpha 中整数为 64 bit。通过定义独立于平台的基本数据类型及其运算，Java 数据可以在任何硬件平台上保持一致。

Java 的数据类型都是依据以上思想具体实现的。因为几乎目前所有 CPU 都能支持常用的数据类型和运算，如 8~64 位整数格式的补码运算和单/双精度浮点运算。Java 编译器本身就是用 Java 语言编写的，Java 语言规范中也没有任何“同具体实现相关”的内容。

8. 解释性

Java 解释器（虚拟机系统）能直接解释、运行目标代码指令，解释程序通常比编译程序所需资源少。

9. 高性能

运行效率低是 Java 长期以来一直被批评的缺点，因为 Java 的字节码需经 Java 虚拟机(JVM)来转换为本地代码，所以速度较慢。但是随着 JVM 技术上的进步，加入 JIT(Just-in-time)实时编译技术，Java 可以在运行时直接将目标代码翻译成机器指令。目前翻译目标代码的速度与 C/C++ 几乎已经没什么区别。

10. 多线程

线程有时也称小进程，是一个大进程里分出来的小的独立的进程。拥有多线程的程序可以充分利用系统资源，改善 CPU 的闲置状态，缩短 CPU 闲置时间，对同一时间运行多个应用程序提供了最佳处理方式。多线程带来的更大的好处是更好的交互性能和实时控制性能。Java 的多

线程功能使得在一个程序里可同时执行多个小任务。在 Java 里, 你可用一个单线程来调一幅图片, 同时你可以访问 HTML 里的其他信息而不必等它。

11. 动态性

Java 的动态特性是其面向对象特性的扩展。它允许程序在运行过程中动态地装入所需要的类, 这是 C++ 无法实现的。在 C++ 程序设计过程中, 每当在类中增加一个实例变量或一种成员函数后, 引用该类的所有子类都必须重新编译, 否则将导致程序崩溃。Java 从几方面采取措施来实现动态性。Java 编译器不是将对实例变量和成员函数的引用编译为数值引用, 而是将符号引用信息在字节码中保存后传递给解释器, 再由解释器完成动态连接类, 最后将符号引用信息转换为数值偏移量。这样, 一个在存储器中生成的对象不是在编译过程中决定, 而是延迟到运行时由解释器确定, 因而在对类中的变量和方法进行更新时就不至于影响现存的代码。当解释、执行字节码时, 这种符号信息的查找和转换过程仅在一个新的名字出现时才进行一次, 随后代码便可以全速执行。在运行时确定引用的好处是可以使用已被更新的类, 而不必担心会影响原有的代码。如果程序连接了网络中另一系统中的某一个类, 该类的所有者也可以自由地对该类进行更新, 而不会使任何引用该类的程序崩溃。

1.3 Java 与 Internet

很少有软件开发工具能够像 Java 那样对软件技术产生了如此深远的影响, 现在的 Java 已不仅是一种程序设计工具, 它还提出了一种全新的“网络计算”计算模式。

传统的 Web 服务器端程序设计通常采用 CGI 的方式, 也就是通用网关接口(Common Gateway Interface)的方式。CGI 程序是用 C 或 Perl 编写的服务器端可执行程序, 专门用来处理客户端浏览器传送过来的 HTTP (Hyper Text Transfer Protocol, 超文本传输协议) 请求中的参数, CGI 程序执行完毕后, 将处理的结果传送给 Web 服务器, 最后由 Web 服务器传给客户浏览器。其执行过程如图 1.3 所示。

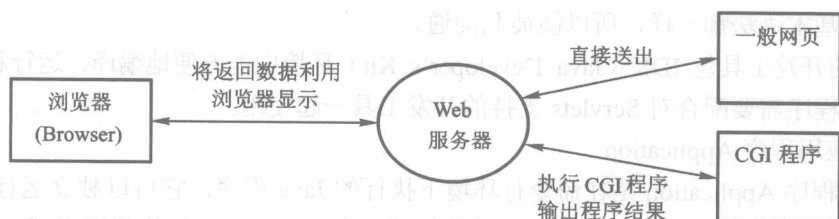


图 1.3 CGI 程序执行流程

CGI 虽然应用比较广, 但存在效率问题。因为 CGI 程序在服务器端以进程的形式运行, 每个客户请求都要产生一个 CGI 程序响应, 当客户请求数量增加时, 服务器的负载将会大大加重, 造成整个 Web 服务器系统运行效率低下, 这对重要系统特别是电子商务系统的影响尤为严重。

Java 的 Servlet 技术可以从一定程度上避免上述问题出现, Servlet 采用单进程多线程的方式, 即每个请求由一个 Java 线程(Thread)响应, 而不是一个操作系统进程, 整个 Web 服务器的开销大为减小, 系统效率得到提高。

另外由于 Java 跨平台的特点, 使 Java 程序可以不加修改就在多种操作系统平台上运行, 这是因为 Java 编译器产生的字节码 (Byte Code) 是独立于具体平台的。从网络上下载的 Java 程序都可在支持 Java 的环境中直接运行。例如可以下载包含了 Java 程序的 Web 网页, 只要客户端的浏览器如 Microsoft Internet Explorer 支持 Java, 它们就可在浏览器中解释、执行。

浏览器下载和执行包含了 Java 程序的 Web 网页的过程, 如图 1.4 所示。

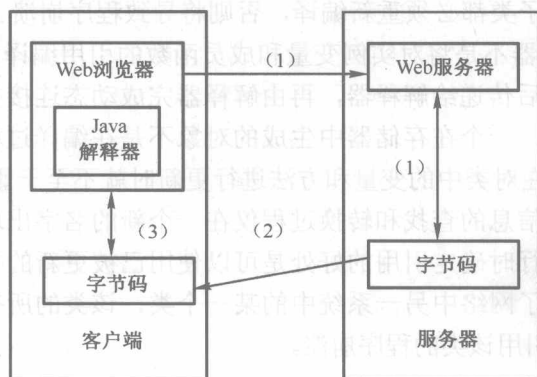


图 1.4 Java 程序在浏览器上的执行过程

- (1) 客户机浏览器与服务器进行连接, 要求下载整个 Web 网页 (包括 Java 字节码文件);
- (2) 客户机浏览器内的 Java 解释器解释、执行下载的字节码;
- (3) 在浏览器上显示 Java 程序的执行结果。

1.4 Java 程序的分类

Java 能开发可独立解释执行的本地应用程序 (Application)、包含在 Web 网页 HTML 文件中依靠浏览器解释并执行的小程序 (Applet) 和后端 Web 服务器程序 (Servlets)。虽然三者的结构不同, 但基本语法都一样, 所以能彼此沟通。

在 Java 的开发工具包 JDK (Java Developer's Kit) 环境中能方便地编译、运行和调试前二种程序, 第三种程序需要配合对 Servlets 支持的开发工具一起使用。

1. Java 应用程序 Application

Java 应用程序 Application 是在命令行环境下执行的 Java 程序, 它可以独立运行在 Java 虚拟机上。所谓独立, 是相对于 Applet 对浏览器的依赖而言, 实际上它的执行离不开 JDK 中的编译器 javac 和解释器 java。

Application 先由 Java 编译器编译成为独立于平台的字节码 (Byte Code), 然后由 Java 解释器 java 来运行。例如一个源程序叫 `app1.java`, 用 `Javac` 编译器编译后将会生成 `app1.class`, 而在命令行状态下输入 `java app1` 就可以运行此程序。

2. Java 小程序 Applet

Java 小程序 Applet, 它是一种嵌入在网页文件中的 Java 字节码程序。由于在网络上传输, Applet 程序往往很短小, 其源文件后缀仍为 `.java`, 编译后后缀也是 `.class`。

由于 Applet 没有自己的程序入口,不能直接在 java 虚拟机上运行,故执行过程与 Application 稍有不同。其具体过程是,首先由 Java 编译器将 Java 程序编译为字节码文件,并把这个字节码文件嵌入到 Web 页面中;客户端将字节码文件下载后,由支持 Java 的 Web 浏览器来解释、执行。

Applet 是提高网页生动性和交互性的有用的手段,因为 Java 提供了大量控制页面外观和处理交互事件的便利方式。图 1.5 是一个简单的图表显示,它是 JDK 工具中自带的一个 Applet 实例。

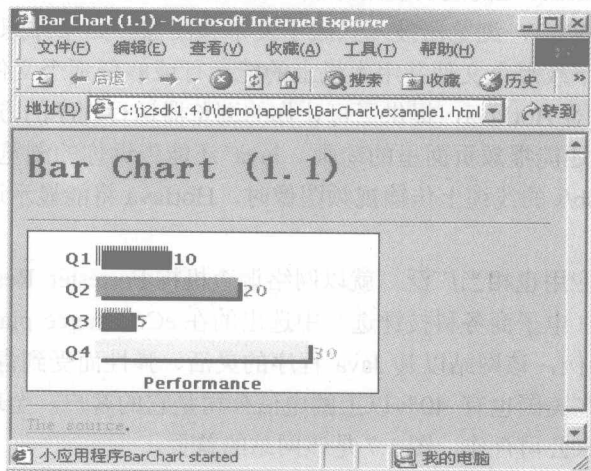


图 1.5 一个简单的图表显示实例

3. Java 服务器端程序 Servlet

Java 服务器端程序 Servlet 在服务器端执行,提供各种处理功能。Servlet 是一种采用 Java 技术来实现 CGI 功能的一种技术。虽然 Servlets 和 CGI 都是在 Web 服务器上运行,并生成 Web 页面,但传统的 CGI 或其他 CGI 类似替代技术相比,Java Servlets 具有效率更高,使用更方便,功能更强大,更小巧也更便宜等特点,例如数据查询及交互操作响应等。图 1.6 是对请求信息的响应。

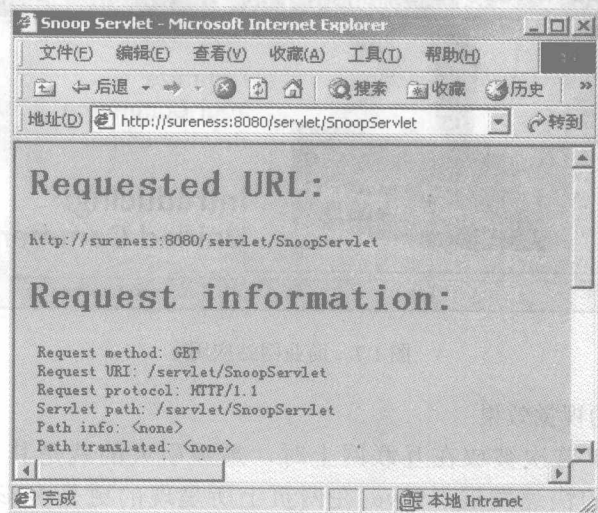


图 1.6 对请求的响应