



J A V A J A V A J A V A

# Java 程序设计

## 基础与应用

编写 / 技能型紧缺人才培养规划教材编写委员会  
主编 / 张 颍 王 琳



本书源代码下载

<http://www.wisbook.com/down/Java123.exe>



海洋出版社

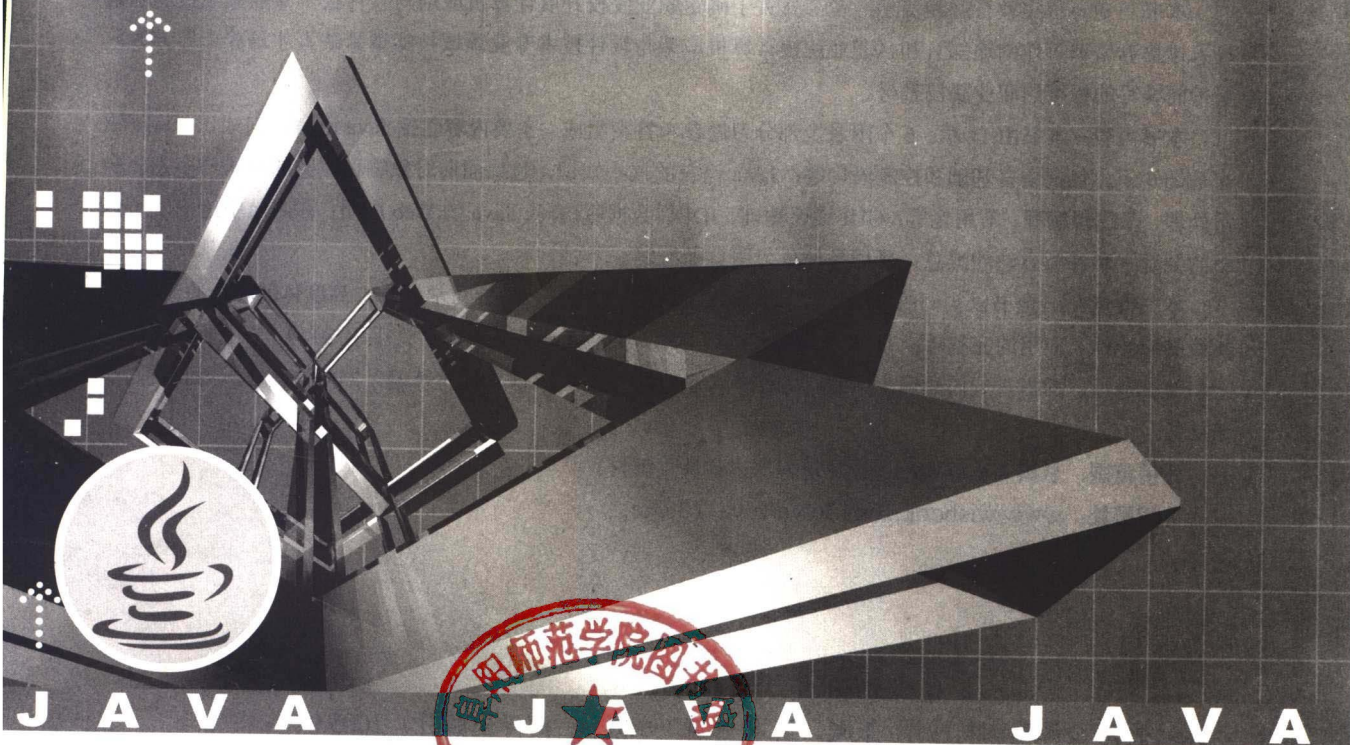


WISBOOK®  
智慧图书

教育部职业教育与成人教育司全国职业教育与成人教育教学用书规划教材

**21 世纪全国职业院校技能型紧缺人才培养规划教材**

中国计算机学会职业教育专业委员会专家组审定



# Java 程序设计 基础与应用

编写 / 技能型紧缺人才培养规划教材编写委员会

主编 / 张 颢 王 琳

海洋出版社

北京

## 内 容 简 介

本书是专为落实教育部和信息产业部《关于确定职业院校开展计算机应用与软件技术专业领域技能型紧缺人才培养培训工作的通知》和《职业院校计算机应用与软件技术专业领域技能型紧缺人才培养指导方案》精神而编写的标准的职业院校教材。

**本书内容：**本书由11章、6个附录及部分习题参考答案组成。主要包括Java语言入门知识；Java语言基础知识，包括语言基础和控制语句等；Java语言的核心知识，包括面向对象编程、类的继承与多态、异常处理、多线程编程、常用包等；GUI设计基础、JDBC数据库访问、Java的Web应用；在附录中给出了Java中的关键字和各类方法的汇总；最后是部分习题参考答案。

**本书特点：**1. 全书以实例的形式讲解，化抽象为具体，理解轻松，容易掌握；2. 高度体现理论的适度性，实践的指导性，应用的完整性；3. 语言精练，结构清晰，按应用由浅入深，手把手引导；4. 书中所有实例均上机调试通过，并提供专门的网上下载，方便读者学习和应用；5. 课后配有练习，方便检测、应用和巩固学习效果。

**适用范围：**全国高等职业院校计算机程序设计课程教材。

**下载网址：**[www.wisbook.com/down/Java123.exe](http://www.wisbook.com/down/Java123.exe)。

### 图书在版编目(CIP)数据

Java 程序设计基础与应用/张颢, 王琳主编. —北京: 海洋出版社, 2005.9

ISBN 7-5027-6365-1

I. J… II. ①张… ②王… III. Java 语言—程序设计—高等学校: 技术学校—教材  
IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 069231 号

总 策 划: WISBOOK

责任编辑: 王勇 钱晓彬

责任校对: 肖新民

责任印制: 肖新民 梁京生

排 版: 海洋计算机图书输出中心 晓阳

出版发行: 海洋出版社

地 址: 北京市海淀区大慧寺路 8 号 (716 房间) 印  
100081

技术支持: [www.wisbook.com/bbs](http://www.wisbook.com/bbs)

经 销: 新华书店

发 行 部: (010) 62132549 62112880-878、875  
62174379 (传真) 86607694 (小灵通)

网 址: [www.wisbook.com](http://www.wisbook.com)

承 印: 北京媛明印刷厂

版 次: 2005 年 9 月第 1 版

2005 年 9 月北京第 1 次印刷

开 本: 787mm×1092mm 1/16

张: 12.75

字 数: 300 千字

印 数: 1~5000 册

定 价: 20.00 元

本书如有印、装质量问题可与发行部调换

# 21 世纪全国高职高专计算机技能型紧缺人才培养规划教材

## 编 委 会

主 任：吴清平

副主任：程时兴 徐 敏 孙振业

委 员（排名不分先后）：

李燕萍 邓振杰 周国烛 果晓来 陈 亮

徐烈英 穆 平 陶晓欣 崔武子 李 红

张建军 朴仁淑 宫 谦 涂玉芬 向 隅

韩祖德 钱晓彬 朱国英 徐 明 乐新宇

周京艳 黄梅琪 蒋湘群 王 勇 王红卫

## 写在前面的话

当前我国正向现代化、信息化、工业化的国家大步迈进,人才资源自然是最为重要的资源。社会各行业、工业企业等部门人才短缺、特别是技能型人才严重短缺,在某种程度上已经影响和妨碍了现代化建设的发展。近年来,我国的职业教育已日益被经济建设所依赖,技能型人才需求存在巨大缺口,因此培养培训任务迫在眉睫。

为配合国家工业化进程和推进城镇化建设,积极培养培训大批适合国家发展和企业需要的有用人才,培养他们成为有一技之长的劳动者和实用型人才,培养的目的主要是面向就业。

本套教材就是面对目前全国职业院校学生的现状和职业需求而编写的、颇具特色的实用培养培训教材,以配合教育部、劳动与社会保障部、国防科工委、信息产业部、交通部、卫生部联合颁发的《教育部等六部门关于职业院校制造业和现代服务业技能型紧缺人才培养培训工程的通知》而行动。

根据以上精神和指导方案,海洋出版社计算机图书出版中心,特组织北京、大连、长春、唐山、武汉、广州、深圳、杭州等地主要职业院校负责人和一线教师,召开教材研讨会,相互交流经验,研究需求,共同策划和编写了本套《21世纪全国高职高专计算机技能型紧缺人才培养规划教材》,倾心奉献给全国广大的教师和学生,为满足社会巨大的人才培养需求做出应有的贡献!

### 整套书的编写宗旨

- 三符合:符合教育部教学大纲、符合市场技术潮流、符合职业院校专业课程需要。
- 技术新、任务明、步骤细致、实用性强,专为技能型紧缺人才量身定制。
- 软件功能与具体范例操作紧密结合,边讲解边动手,学习轻松,上手容易。
- 三适应:适应新的教学理念、适应学生水平现状、适应用人标准要求。

### 整套书的特色

- 理论精练够用、任务明确具体、技能实操落实,活学活用。

教材编委会

# 前 言

在信息技术飞速发展的今天，计算机已进入人们日常生活的各个方面，而 Internet 又是连接世界各地计算机的桥梁。面对 Internet 庞大的信息资源，如何选择一种较好的语言在网络传输信息中应付自如，是许多程序员都倍感费心的事。Java 语言的出现为我们提供了一种最佳的选择。Java 语言的面向对象性、平台无关性、安全性和健壮性及多线程和网络支持能力等诸多特点，使它很快成为网络世界的通用语言。学习掌握 Java 语言可以自如迎接信息时代的挑战。

Java 语言是一门面向对象的程序设计语言，但同时又是标准的、大众化的面向对象编程语言。Java 将面向对象编程技术中一些抽象的很难理解的内容，如继承、多态、异常、多线程、客户机/服务器等，在 Java 中都得到了简化。因此，Java 语言适合初学编程者学习。

本书是一本实用编程教材，学以致用是编写本书的指导思想。不讲深奥的原理，不涉及不常用的知识，只介绍学习编程最需要的内容。每章都有紧扣所讲内容的实例和习题，用以增强读者的理解、促进应用。

本课程的参考学时为 68 学时。全书共 11 章，第 1 章是 Java 语言入门，介绍了 Java 语言的概况，引导读者进入 Java 世界；第 2 章和第 3 章介绍了 Java 语言的基础知识，包括语言基础和控制语句等，这是必须掌握的内容；第 4 章至第 8 章介绍了 Java 语言的核心知识，包括面向对象编程、类的继承与多态、异常处理、多线程编程、常用包等；第 9 章至第 11 章包括 GUI 设计基础、JDBC 数据库访问、Java 的 Web 应用等内容。全书以实例讲解，逻辑清晰，易于初学者上手。为配合读者学习，书中各章例题程序源代码可在 [www.wisbook.com/dow/Javal23.exe](http://www.wisbook.com/dow/Javal23.exe) 下载。

本书第 1、2、3、4、5 章由张颢编写；第 6 章由张文静编写；第 7 章由王琳编写；第 8 章由王向军编写；第 9、10、11 章由王苏编写；本书的附录及中英文对照部分由张翼翼编写；张颢统编全稿。

由于编者水平有限，错误和不当之处在所难免，敬请广大读者不吝指正。

编 者

## 《Java 程序设计基础与应用》学时分配建议

**总学时：68**

| 章 序 号  | 主要内容                                  | 讲授 (学时) | 上机 (学时) | 合计 (学时) |
|--------|---------------------------------------|---------|---------|---------|
| 第 1 章  | Java 简介                               | 1       |         | 1       |
|        | 编程方法                                  | 1       |         | 1       |
|        | Java 程序的开发                            | 1       | 1       | 2       |
| 第 2 章  | 基本语言要素                                | 2       |         | 2       |
|        | 数据类型                                  | 1       | 1       | 2       |
|        | 运算符                                   | 1       | 1       | 2       |
|        | 数组                                    | 1       | 2       | 3       |
|        | 字符串                                   | 1       | 1       | 2       |
| 第 3 章  | 选择控制语句                                | 1       | 2       | 3       |
|        | 循环控制语句                                | 1       | 2       | 3       |
|        | 跳转控制语句                                | 1       | 2       | 3       |
| 第 4 章  | 面向对象中的基本概念                            | 1       | 1       | 2       |
| 第 5 章  | 类                                     | 2       | 1       | 3       |
|        | 对象                                    | 2       | 1       | 3       |
|        | 类的继承、多态与重载                            | 1       | 2       | 3       |
| 第 6 章  | 异常捕获、处理与抛出                            | 1       | 1       | 2       |
| 第 7 章  | 多线程机制与编程实例                            | 2       | 2       | 4       |
| 第 8 章  | 常用包和标准输入/输出                           | 1       | 1       | 2       |
| 第 9 章  | 框架、事件、布局管理器<br>文本框、密码框与对话框<br>菜单系统、表格 | 4       | 4       | 8       |
| 第 10 章 | 用 Java 访问数据库                          | 4       | 5       | 9       |
| 第 11 章 | Java 的 Web 应用                         | 4       | 4       | 8       |

# 目 录

|  |  |
|--|--|
| <b>第 1 章 Java 概述</b> .....1              |  |
| 1.1 引言.....1                             |  |
| 1.2 Java 简介.....1                        |  |
| 1.2.1 Java 的历史.....1                     |  |
| 1.2.2 Java 的发展历程.....1                   |  |
| 1.2.3 Java 的特点.....2                     |  |
| 1.3 编程方法.....4                           |  |
| 1.3.1 过程编程.....4                         |  |
| 1.3.2 结构化编程.....4                        |  |
| 1.3.3 面向对象的编程.....5                      |  |
| 1.4 Java 程序的开发.....5                     |  |
| 1.4.1 Java 运行环境及安装.....6                 |  |
| 1.4.2 编制第一个 Java Application<br>程序.....7 |  |
| 1.4.3 字节码的编译生成.....9                     |  |
| 1.4.4 字节码的解释与运行.....10                   |  |
| 1.4.5 另一种类型的 Java 程序.....10              |  |
| 1.5 小结.....11                            |  |
| 1.6 习题.....12                            |  |
| <b>第 2 章 Java 语言基础</b> .....13           |  |
| 2.1 基本语言要素.....13                        |  |
| 2.1.1 标识符.....13                         |  |
| 2.1.2 关键字.....14                         |  |
| 2.1.3 分隔符.....14                         |  |
| 2.1.4 注释.....14                          |  |
| 2.2 数据类型.....15                          |  |
| 2.2.1 基本数据类型.....15                      |  |
| 2.2.2 常量.....17                          |  |
| 2.2.3 变量.....18                          |  |
| 2.2.4 各种数据类型的转换.....19                   |  |
| 2.3 运算符.....20                           |  |
| 2.3.1 算术运算符.....20                       |  |
| 2.3.2 关系运算符.....21                       |  |
| 2.3.3 条件运算符.....22                       |  |
| 2.3.4 逻辑运算符.....22                       |  |
| 2.3.5 位运算符.....23                        |  |
| 2.3.6 其他运算符.....24                       |  |
| 2.3.7 运算符优先级.....25                      |  |
| 2.4 数组.....25                            |  |
| 2.4.1 一维数组.....25                        |  |
| 2.4.2 二维数组.....27                        |  |
| 2.5 字符串.....27                           |  |
| 2.5.1 创建字符串常量.....27                     |  |
| 2.5.2 字符串转换为相应类型数值.....28                |  |
| 2.5.3 数值转换为字符串.....29                    |  |
| 2.5.4 String 类中的方法.....29                |  |
| 2.6 小结.....32                            |  |
| 2.7 习题.....33                            |  |
| <b>第 3 章 Java 控制语句</b> .....35           |  |
| 3.1 选择控制语句.....35                        |  |
| 3.1.1 if 语句.....35                       |  |
| 3.1.2 switch 语句.....37                   |  |
| 3.2 循环控制语句.....38                        |  |
| 3.2.1 for 语句.....38                      |  |
| 3.2.2 while 语句.....39                    |  |
| 3.2.3 do-while 语句.....40                 |  |
| 3.3 跳转语句.....41                          |  |
| 3.3.1 break 语句.....41                    |  |
| 3.3.2 continue 语句.....42                 |  |
| 3.3.3 return 语句.....43                   |  |
| 3.4 小结.....44                            |  |
| 3.5 习题.....44                            |  |
| <b>第 4 章 面向对象编程</b> .....45              |  |
| 4.1 面向对象编程基础.....45                      |  |
| 4.1.1 面向对象概述.....45                      |  |
| 4.1.2 面向对象中的基本概念.....46                  |  |
| 4.2 类.....48                             |  |
| 4.2.1 创建新类型.....48                       |  |
| 4.2.2 为什么要创建新类型.....49                   |  |
| 4.2.3 类的声明.....49                        |  |
| 4.2.4 类的成员变量.....53                      |  |
| 4.2.5 类的成员方法.....56                      |  |



|                               |    |                                   |     |
|-------------------------------|----|-----------------------------------|-----|
| 4.2.6 类的成员的访问控制 .....         | 58 | <b>第 8 章 常用包和标准输入/输出 .....</b>    | 92  |
| 4.2.7 方法的参数传递 .....           | 59 | 8.1 Java 常用包 .....                | 92  |
| 4.3 对象 .....                  | 60 | 8.2 标准输入/输出 .....                 | 93  |
| 4.3.1 创建对象 .....              | 60 | 8.2.1 标准输入方法 .....                | 93  |
| 4.3.2 使用对象 .....              | 60 | 8.2.2 标准输出方法 .....                | 94  |
| 4.3.3 清除对象 .....              | 62 | 8.3 小结 .....                      | 95  |
| 4.4 小结 .....                  | 62 | 8.4 习题 .....                      | 95  |
| 4.5 习题 .....                  | 63 | <b>第 9 章 GUI 设计基础 .....</b>       | 96  |
| <b>第 5 章 类的继承与多态 .....</b>    | 64 | 9.1 应用程序框架 .....                  | 96  |
| 5.1 继承 .....                  | 64 | 9.1.1 框架类 JFrame .....            | 96  |
| 5.1.1 认识继承 .....              | 64 | 9.1.2 按钮类 JButton .....           | 97  |
| 5.1.2 成员变量的继承和隐藏 .....        | 67 | 9.1.3 标签类 JLabel .....            | 99  |
| 5.1.3 成员方法的覆盖 .....           | 69 | 9.1.4 AWT 包的字体类 Font .....        | 99  |
| 5.1.4 this 和 super .....      | 70 | 9.1.5 AWT 包的颜色类 Color .....       | 100 |
| 5.2 类的多态与重载 .....             | 72 | 9.2 Action 事件处理 .....             | 102 |
| 5.2.1 关于多态 .....              | 72 | 9.2.1 事件驱动 .....                  | 102 |
| 5.2.2 方法的重载 .....             | 72 | 9.2.2 Action 事件处理 .....           | 102 |
| 5.2.3 构造方法重载 .....            | 75 | 9.2.3 区别事件源 .....                 | 104 |
| 5.3 小结 .....                  | 77 | 9.3 布局管理器 .....                   | 106 |
| 5.4 习题 .....                  | 77 | 9.3.1 swing 容器 .....              | 106 |
| <b>第 6 章 异常处理 .....</b>       | 78 | 9.3.2 布局管理器的构造方法 .....            | 106 |
| 6.1 异常处理基础 .....              | 78 | 9.3.3 布局管理器的使用 .....              | 107 |
| 6.1.1 异常及其发生的原因 .....         | 78 | 9.4 文本框与密码框 .....                 | 108 |
| 6.1.2 异常的分类 .....             | 78 | 9.4.1 单行输入文本框类 JTextField .....   | 108 |
| 6.2 捕获及处理异常 .....             | 79 | 9.4.2 密码输入框类 JPasswordField ..... | 109 |
| 6.2.1 try 和 catch 的使用 .....   | 79 | 9.4.3 文本框与密码框的事件处理 .....          | 109 |
| 6.2.2 finally .....           | 80 | 9.5 对话框 .....                     | 110 |
| 6.3 异常抛出 .....                | 81 | 9.5.1 系统预定义对话框 .....              | 110 |
| 6.3.1 throw 和 throws 语句 ..... | 81 | 9.5.2 自定义对话框 .....                | 112 |
| 6.3.2 定义自己的异常类 .....          | 82 | 9.6 菜单系统 .....                    | 114 |
| 6.4 小结 .....                  | 83 | 9.6.1 菜单栏 JMenuBar 类 .....        | 115 |
| 6.5 习题 .....                  | 83 | 9.6.2 菜单 JMenu 类 .....            | 115 |
| <b>第 7 章 多线程编程 .....</b>      | 84 | 9.6.3 JMenuItem 类 .....           | 115 |
| 7.1 Java 多线程机制 .....          | 84 | 9.6.4 菜单项目的事件处理 .....             | 116 |
| 7.1.1 Java 线程的基本概念 .....      | 84 | 9.7 表格 .....                      | 117 |
| 7.1.2 多线程的实现 .....            | 85 | 9.7.1 简单表格 .....                  | 117 |
| 7.1.3 线程的同步 .....             | 87 | 9.7.2 默认的表格模型 .....               | 118 |
| 7.2 多线程编程实例 .....             | 88 | 9.7.3 表格的选择模式 .....               | 120 |
| 7.3 小结 .....                  | 91 | 9.8 小结 .....                      | 121 |
| 7.4 习题 .....                  | 91 | 9.9 习题 .....                      | 121 |

|                                  |     |   |     |
|----------------------------------|-----|---|-----|
| <b>第 10 章 JDBC 数据库访问</b> .....   | 123 | <b>第 11 章 Java 的 Web 应用</b> .....         | 145 |
| 10.1 使用 MySQL 数据库.....           | 123 | 11.1 客户端小应用程序 applet.....                 | 145 |
| 10.1.1 MySQL 数据库安装与<br>启动.....   | 124 | 11.1.1 JApplet 类的基本方法.....                | 146 |
| 10.1.2 SQL 语言简介——数据<br>定义语句..... | 125 | 11.1.2 使用 swing 组件的 applet.....           | 146 |
| 10.1.3 SQL 语言简介——数据<br>操作语句..... | 126 | 11.1.3 转换 swing 应用程序为<br>swing 小应用程序..... | 147 |
| 10.2 java 数据库连接.....             | 128 | 11.2 服务器端小应用程序 servlet.....               | 148 |
| 10.2.1 JDBC 简介.....              | 128 | 11.2.1 交互式网页.....                         | 148 |
| 10.2.2 JDBC 的数据库驱动程序.....        | 128 | 11.2.2 servlet 引擎: tomcat.....            | 148 |
| 10.2.3 JDBC 的基本功能.....           | 129 | 11.2.3 servlet 版的“Hello,<br>World!”.....  | 150 |
| 10.3 应用 JDBC.....                | 129 | 11.2.4 JDBC 应用.....                       | 153 |
| 10.3.1 加载 JDBC 驱动.....           | 129 | 11.3 java 服务器网页 jsp.....                  | 156 |
| 10.3.2 建立数据库的连接.....             | 129 | 11.3.1 jsp 简介.....                        | 156 |
| 10.3.3 发送 SQL 语句.....            | 131 | 11.3.2 jsp 快速上手.....                      | 157 |
| 10.3.4 访问结果集.....                | 132 | 11.3.3 在 jsp 中使用 javaBean.....            | 160 |
| 10.4 应用举例: 通讯录.....              | 134 | 11.4 小结.....                              | 162 |
| 10.4.1 封装数据库访问的基本<br>操作.....     | 134 | 11.5 习题.....                              | 163 |
| 10.4.2 自定义 JTable 类表格模型.....     | 136 | <b>附录 A Java 关键字</b> .....                | 164 |
| 10.4.3 为表格模型增加删除与插<br>入功能.....   | 139 | <b>附录 B Java 编程规则</b> .....               | 166 |
| 10.4.4 整合各个模块.....               | 142 | <b>附录 C String 类方法汇总</b> .....            | 169 |
| 10.5 小结.....                     | 144 | <b>附录 D Thread 类方法汇总</b> .....            | 174 |
| 10.6 习题.....                     | 144 | <b>附录 E InputStream 类方法汇总</b> .....       | 178 |
|                                  |     | <b>附录 F PrintStream 类方法汇总</b> .....       | 179 |
|                                  |     | <b>部分习题参考答案</b> .....                     | 181 |

# 第 1 章 Java 概述

## 本章要点

- ☑ 理解 Java 语言的平台无关性
- ☑ 了解编程方法的演变过程
- ☑ 熟悉简单的 Java 应用程序及小应用程序的编写

## 1.1 引言

众所周知, Internet 是当今的热点, 至于 Java 语言在 Internet 开发上的重要地位, 可能很多读者并不知道。Java 语言最初的用途并不是在 Internet 上, 甚至连 Java 这个名字也不是原名。读完本章后就会对 Java 语言有一个基本认识。

## 1.2 Java 简介

本节主要介绍 Java 语言的历史与发展以及 Java 语言的特点。

### 1.2.1 Java 的历史

1991 年, 美国 Sun Microsystems 公司成立了一个开发小组, 目的是为交互式电视机、烤面包箱等消费类家用电器开发一个分布式代码系统, 使用户能将电子邮件发给这些家用电器, 从而对它们进行控制, 与它们进行信息交流。这个项目被命名为 Green。

当时 Java 的设计者 James Gosling (詹姆斯·高斯灵) 率领开发小组准备采用已经成熟的面向对象语言 C++ 来完成, 但 C++ 太复杂, 安全性差, 于是用 C++ 开发了一种新的语言, 取名为 Oak (橡树)。这个名称来自开发小组的主持人 James Gosling 办公室外的一棵橡树。

Oak 先被用于家用电器的控制系统中。逐渐地 Oak 系统成熟起来, Sun 公司曾用此投标一个交互式电视项目, 但结果被 SGI 击败了。在 Oak 无路可走的情况下, 恰巧 Internet 迅速发展起来了, Sun 公司决策层认为 WWW 可以挽救 Oak。此时, 美国伊利诺斯大学开发的 Mosaic WWW 浏览器启发了项目组成员, 他们用 Oak 编制了 WebRunner 浏览器, 并得到了 Sun 决策层的支持。1995 年 1 月, Oak 被重新赋予了一个更响亮的名字——Java, WebRunner 则被命名为 HotJava。随后, Sun 公司将 Java 和 HotJava 在国际互联网上免费发布。

关于 Java 这个名字还有一个流传很广的故事。这个故事说, Java 是几个技术人员在一起品尝 Java 岛 (爪哇, 印尼的一个地名) 出产的咖啡时想起来的, Java 这个岛的咖啡很出名, 这些技术人员也许希望 Java 程序设计语言也同样出名吧。

### 1.2.2 Java 的发展历程

自从 1995 年 Java 语言被正式推出之后, Java 语言就以其独到的优势迅猛发展, 经过短短

8、9 年时间，成为迄今为止最为优秀的面向对象程序设计语言。Java 也从当初的一种语言逐渐形成一种产业，基于 Java 语言的 J2EE 架构已经成为微软.NET 平台的强大竞争对手。万维网 WWW 的创始人 Berners-Lee 曾经说过：计算机发展的下一个浪潮就是 Java，并且将会很快发生的。现在看来，这一预言已成为不争的事实。比尔·盖茨也不无感慨地说过：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 单机时代向着以网络为中心的计算机时代转移，而毫不犹豫地购买 Java 使用权则是其重大战略决策的实施。Java 最大的特点是跨平台性，将是未来网络世界中的“世界语”。有人预言，今后所有的用其他语言编写的软件统统都要用 Java 语言来改写。

当初，Java 语言的发布不亚于一场革命，但是它并不标志着 Java 快速革命时代的结束。与其他大多数编程语言、软件系统的小规模改进不同，Java 发布后就继续以爆炸式的步伐向前发展。在 Java1.0 发布后不久，Java 的设计者就已经制定出了 java1.1。从版本号看来只增加了 0.1，但 Java1.1 新增的特性远比普通意义上的版本修订有意义，内容也要丰富许多。Java1.1 增加了许多新的库元素，重新定义了小应用程序处理事件的方法，并重新设置了 1.0 版本中库的许多特性。同时，它也放弃了原来在 Java1.0 中所定义的若干过时的特征。

Java 语言的第二个主要版本是 Java 2。Java 2 是一个分水岭，它标志这个快速演变语言时代的开始。Java 2 第一版的版本号是 1.2，Java 2 增加了很多对新特性的支持，例如 Swing 和类集框架，并且提高了 Java 虚拟机和各种编程工具的性能。Java 2 也摒弃了一些不赞成继续使用的内容，主要是不赞成线程类中 Suspend()、Resume()和 Stop()等方法的使用。

Java 的最新版本是 Java 2.1.4 版。该版本是对 Java 2 原有版本的一次重要升级。这个版本增强了 Java 大部分现有的功能，去除了部分过时的属性。目前来看，版本 1.2 至 1.4 的程序源代码都是兼容的。尽管版本 1.4 作了一些改变，主要是在类库的扩充上。本书介绍的内容、程序代码对这些版本基本都是适用的。

### 1.2.3 Java 的特点

Java 作为目前被广泛使用的面向对象编程语言，它具有多方面的特点。按照设计者的解释，Java 是一个面向对象、编写简单、脱离机器结构、适用网络、健壮性、可移植性、安全性、多线程、解释执行具有很高的性能。另外，还具有动态特点。

#### 1. 面向对象

Java 被设计成为一种真正的面向对象编程语言，尽管受到前辈语言的影响，Java 并没有兼容其他语言原代码的程序。这使得 Java 可以更直接、更易用、更实际地接近对象、描述对象、操作对象。

Java 的设计集中于对象及接口，它提供了简单的类机制以及动态的接口模型。对象封装了它的状态变量和相应的方法，实现了模块化及信息隐藏；而类则提供了一类对象的原形，并通过继承机制，子类可以使用父类所提供的方法，实现了代码重用。

#### 2. 简单性

这是 Java 最重要的特点。Java 语言的设计者虽然是基于 C++来设计 Java 的，但却删除了 C++中几乎不用或很难用好的一些面向对象的特征。如：Java 不支持 struct、union 和 pointer 数据类型；不支持 typedef 和 #define；不支持运算符的重载和多重继承性。Java 内部有一个自

动分配和释放内存的系统，使程序员从繁琐的事务中解放出来，在编程中少犯错误。

简单性还体现在小上，Java 解释器、系统模块和运行模块都比较小，适合在小型机上运行，从网上下载也很容易。

### 3. 结构中立

Java 设计者考虑的一个主要问题是程序代码的常久性和可移植性，而程序员面临的一个主要问题则是：不能保证今天编写的程序明天能否在同一台机器上顺利运行。操作系统的更新升级、处理器的主频提高以及核心系统资源的变化，都有可能导致程序无法继续生存。Java 虚拟机 (JVM) 就很好地解决了这个问题，基本上实现了只要写一次程序，在任何地方、任何时间该程序永远都能够运行。

结构中立实现的方法是：Java 解释器生成与体系结构无关的字节码 (bytecode)，这些字节码只对应于 Java 虚拟机中的表示，Java 解释器得到字节码后，对它进行转换，它和计算机平台无关，只要安装了 Java 运行环境，Java 程序就可以在任何处理器上运行。

### 4. 适用网络

Java 提供了大量的系统模块支持基于 TCP/IP 协议的编程，用 Java 建立网络连接要比 C/C++ 容易。Java 程序通过 URL 访问网络资源和存取本地文件系统一样简单。

### 5. 健壮性

Java 在编译和运行时都进行严格的检查，以杜绝类型匹配错误及其他潜在问题。Java 取消了一些容易引发问题的特征 (如指针)，而且内存释放是完全自动的，避免了程序员像在 C/C++ 中手工分配和释放内存所带来的问题。

### 6. 可移植性

平台无关的特性使 Java 程序可以方便地移植到网络上的不同机器。Java 的类库中也实现了与不同平台的接口，使这些类库可以移植。另外，Java 编译器是由 Java 语言实现的，Java 解释器是由标准 C 实现，这使得 Java 系统本身也具有可移植性。

### 7. 安全性

用于网络、分布环境下的 Java 必须要防止病毒的入侵。Java 不支持指针，一切对内存的访问都必须通过对象的实例变量来实现，关死了病毒发起攻击的大门。

### 8. 多线程

设计 Java 的目的之一是满足人们对创建交互式网上程序的需要。为此，Java 支持多线程编程。所谓多线程，是指多个线程并发执行的能力。用 Java 编写的应用程序可以同时执行多个任务。Java 解释器在多线程同步方面具有成熟的解决方案，可以使程序员创建出健壮的多线程程序。Java 的多线程机制使用起来非常方便，只需关注程序细节的编写，而不用担心后台的多任务系统。

### 9. 高性能

虽然字节码翻译执行的速度已相当快，但有些情况下还是要求程序有更高的执行性能。程序运行时，字节码被快速翻译成当前 CPU 的指令，Sun 用直接解释器进行的一秒钟内 300 000 个方法调用的实验，证明解释型字节码翻译成机器码的速度和 C/C++ 几乎没有区别。

## 10. 动态性

Java 被设计成能适应环境变化的语言。程序带有多种运行时类型信息，用于在运行时校验和解决对象访问问题。这使得 Java 可以在运行中加载，Java 的类库可以自由添加方法和属性而不会影响到用户程序。因此，Java 的动态性可以更好地适应不断变化的执行环境。

## 1.3 编程方法

要想成为一名合格的程序员，了解编程方法的演变大有益处。

### 1.3.1 过程编程

我们知道，所有的计算机程序均由两类元素组成：代码和数据。最初，程序的构筑一般围绕“正在发生什么”而编写代码，这种方法被称作为面向过程的编程。程序被看作是一系列处理数据的过程，一个过程或函数是指一组特定的一个接一个执行的指令。数据和过程是分离的，编程的技巧主要是跟踪哪些函数调用了另一些函数以及哪些数据被改变了。面向过程的编程模型可以认为是代码作用于数据，像 Pascal、C 这样的过程式语言采用此模型是相当成功的。然而，使用面向过程的编程方法对小程序的编写可能有效，但当程序变得非常大且更为复杂时，就会出现种种问题，直至失去对代码的有效控制。由此对软件工程中的编程方法问题提出了新的要求，结构化编程方法应运而生。

### 1.3.2 结构化编程

结构化编程的主要思想是“分而治之，各个歼灭”。一个计算机程序可以看作是由一系列任务组成的。任何一项过于复杂无法简单描述的任务都将分解为一系列较小的子任务，直至每个任务都很小，很容易解决。

例如，要计算某公司中每一位员工的平均工资，可以先将其分解成下面几个子任务：

- (1) 确定每人挣多少。
- (2) 求出共有多少人。
- (3) 计算总工资额。
- (4) 用总工资除以总人数。

计算总工资额还可以分成以下几步：

- (1) 读取每个员工的记录。
- (2) 读取工资额。
- (3) 将工资额添加到汇总函数。
- (4) 读取下一个员工的记录。

依次类推，获取每个员工的记录又可分为以下几步：

- (1) 打开员工文件。
- (2) 找到正确的记录。
- (3) 从磁盘读取数据。

结构化编程仍然是处理复杂问题的一种非常成功的方法。但是，到了 20 世纪 80 年代后期，它的许多不足逐渐暴露出来。首先，人们直观的愿望是将数据（例如员工记录）及其操作（排序、编辑等）看作一个整体。可惜，结构化编程通常将数据结构从函数中分离出来，因此，在

结构化编程中没有将数据连接到函数的直观方式。结构化编程通常也被称为过程化编程，因为其着重于过程而不是对象。

### 1.3.3 面向对象的编程

上面提到的面向过程编程，遵循的是面向过程的问题求解方法，中心思想是用计算机能够理解的逻辑来描述和表达待解决的问题及其具体的解决流程。数据结构和算法是面向过程问题求解的核心所在。面向对象技术则代表了一种全新的程序设计思路，其观察、表述、处理问题的方法，与传统的面向过程的编程方法不同。面向对象的程序设计和问题求解力求符合人们日常自然的思维习惯，尽量分解、降低问题的难度和复杂性，从而提高整个求解过程的可监测性、可控制性及可维护性，以此达到以较小的代价和较高效率获得较满意的效果。

面向对象编程的一个实质性要素是抽象。人们通过抽象（Abstraction）来处理编程过程中遇到的复杂性问题。例如，当看到一辆汽车，一般不会把它想象成由几个互相独立的零件所组成的一套动力装置，而是把整个汽车看成一个具有自己独立行为（停止、启动、运行、加速、减速、转向等）的对象。这种抽象使人们很容易地将一辆汽车开到目的地，而不会因为组成汽车各部分零件过于复杂而不知所措。他们可以忽略发动机的工作原理，可以忽略引擎、传动及刹车系统的工作细节，而将汽车作为一个整体来使用。

使用层级划分是管理抽象的一个有效方法，它允许根据物理上的意义将复杂的系统分解为更多更容易处理的小块。例如，汽车是一个独立的对象。一旦到了内部，会看到汽车由若干子系统组成：驾驶系统、制动系统、音响系统、安全保障系统、供暖制冷系统等。再进一步细分，这些子系统由更多的专用元件组成。例如，音响系统由一台收音机、一台 CD 播放机、或许还有一台磁带放音机组成。从中得到的重要启发是，我们要通过层级抽象对复杂的汽车进行管理和控制。

毫无疑问，复杂系统的分层抽象也能被运用于计算机程序设计。在传统的面向对象程序中，数据经过抽象可以用若干个组成对象表示，程序中的过程步骤则可看成是在这些对象之间进行消息收集。这样，每一个对象都有它自己的独特行为特征。用户可以把这些对象当作具体的实体，让它们知道，对告诉它们做什么事的消息做出反映。这便是面向对象编程的本质。

Java 与面向对象编程密不可分，因为它是纯面向对象的编程语言，哪怕是编写最简单的 Java 程序，也必须理解面向对象编程的基本特征和原则。有关面向对象编程的具体问题在本书第 4 章再作介绍。

## 1.4 Java 程序的开发

开始编写 Java 程序前，需要在计算机系统中安装 Java 平台。Java 平台可从 Sun 公司的 [java.sun.com](http://java.sun.com) 网站免费下载，该网站上提供了免费的 Java 开发工具集（Java Developers Kits，简称 JDK），可以在这里下载适合于自己计算机操作系统的 JDK，利用这些工具来开发 Java 程序。

打开浏览器窗口，在地址栏输入 [java.sun.com](http://java.sun.com)，主页中显示了 Java 2 平台的三种不同版本：Enterprise Edition (J2EE, Java 2 企业版)、Standard Edition (J2SE, Java 2 标准版) 和 Micro Edition (J2ME, Java 2 微型版)，如图 1-1 所示。本书使用的开发平台是 J2SE (Java 2 标准版)，版本是 J2SE 1.4.2 SDK。鉴于目前大多数读者使用的是 Windows 操作系统，本书的程序调试与运行都在 Windows 操作系统下进行。

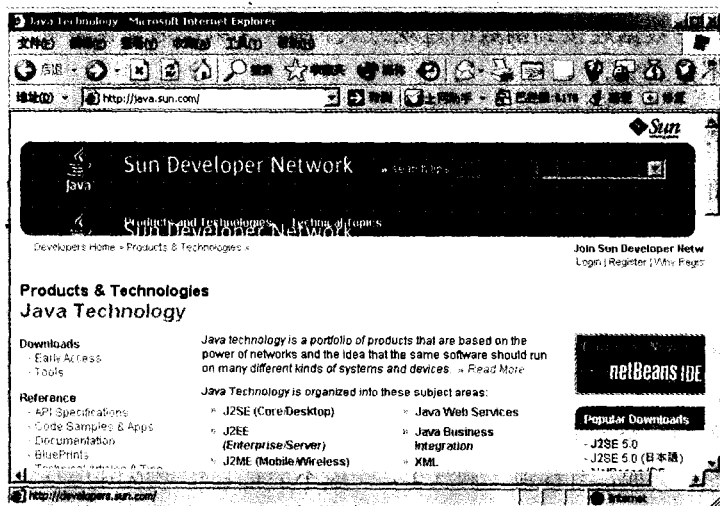


图 1-1 Sun 公司网站主页

## 1.4.1 Java 运行环境及安装

### 1. 运行环境

Java 平台由 Java 应用编程接口 (Application Programming Interface, 简称 API) 和 Java 虚拟机 (Java virtual machine, 简称 JVM) 构成。

Java 应用编程接口是已编译的可在程序中使用的代码库。它们使编程者能够添加现成的可定制的功能, 以节约编程时间。

Java 虚拟机是建立在实际的处理器基础上的假想的计算机。Java 虚拟机 (JVM) 被定义为: 在真实机器中用软件模拟实现的一种想象机器。Java 虚拟机代码被存储在 .class 文件中; 每个文件都包含最多一个 public 类。

Java 程序并不是在本机操作系统上直接运行, 而是通过 Java 虚拟机向本机操作系统进行解释来运行。这就是说, 任何安装有 Java 虚拟机的计算机系统都可以运行 Java 程序, 而不论最初开发应用程序的是何种计算机系统。

### 2. J2SDK 1.4.2 的安装

首先获取软件包, 然后进行安装, 可以像安装其他软件那样安装 JDK, 使用默认文件夹 c:\j2sdk1.4.2, 本书统一使用 c:\j2sdk1.4.2 作为 J2SE 的安装文件夹。安装完后浏览文件夹中的内容, 其中的 bin 文件夹中存放的是 Java 的可执行文件, 建议读者在 autoexec.bat 文件中加入下面两行:

```
set classpath=.;c:\j2sdk1.4.2\lib
path=c:\j2sdk1.4.2\bin;
```

设置 JDK 路径后, 需要重新启动计算机, 以便让 autoexec.bat 文件设置生效。以后在任何文件夹下都可以执行 Java 源代码了。

### 3. J2SDK 1.4.2 介绍

J2SDK 1.4.2 软件包中常用的工具包括:

- Javac: Java 语言编译器。



本书各章的程序都是在命令窗口中通过执行“javac 文件名”来编译 Java 程序的。

Java 源文件名必须以.java 为后缀，编译后生成以.class 为后缀的字节码文件。

- Java: 字节码解释器。

用于执行 Java 程序。本书各章的程序大都通过在命令窗口输入“java 字节码文件名”来运行由 javac 编译好的程序。输入命令时，字节码文件的后缀不用输入。

- Javap: Java 字节码分解程序。

可显示字节码文件的包、标记为 public 及 protected 的变量和方法等信息。

- Javaprof: 资源分析工具。

用于分析 Java 程序在运行过程中调用了哪些资源，包括类和方法的调用次数和时间，以及各数据类型的内存使用情况等。

- Javah: C 代码处理工具。

用于从 Java 类调用 C++ 代码。

- Appletviewer: Applet (小应用程序) 浏览工具。

在没有浏览器时，可用来运行和调试 Java Applet 程序。

- Jdb: Java 调试工具。

- Javadoc: 用于生成 API 文档。

在编写程序时，把注释语句写在“/\*”和“\*/”之间，其内容可被 javadoc 识别。执行“java\*.java”，自动生成 API (Application Programming Interface, 应用编程接口) 文档。

- Jar: 管理 jar 文件。

使用该工具可以将 Java 程序打包成为一个 (\*.jar)，.jar 文件比 Java 源文件小，处理起来更方便。

以上常用的工具命令是：Java 语言编译器和字节码解释器。

#### 4. Java 程序的编译

Java 程序的编译程序是 javac.exe。javac 命令将 Java 程序编译成字节码，然后用 java 命令解释执行这些 Java 字节码。

Java 程序源码必须存放在后缀为.java 的文件里。对于 Java 程序里的每一个类，javac 都将生成与类名称相同但后缀为.class 的文件。编译器自动把.class 文件放在与.java 命令相同的文件夹里，除非用了-d 选项。

在引用自己定义的类时，必须指明它们存放的文件夹，这时就需要使用环境变量参数 classpath。环境变量 classpath 由一个被分号隔开的路径名组成。如果传递给 javac 编译器的源文件里引用了自定义类，在本文件和传递的其他文件中找不到时，那么编译器会按照 classpath 指定的路径来搜索它。如设置路径：“set classpath=.;c:\j2sdk1.4.2\lib”，则编译器先搜索当前文件夹，如果没找到，则继续搜索“c:\j2sdk1.4.2\lib”文件夹。

### 1.4.2 编制第一个 Java Application 程序

编辑源程序，可以不采用任何集成开发环境，只是采用文本编辑器（例如：Windows 下的“记事本”）就可以编写简短的 Java 程序。

**【例 1.1】** 我的第一个 Java Application 程序。

```
//MyFirstApp.java
```