

21 世纪高等学校计算机应用技术规划教材



# Java 语言程序 设计教程

邓 琨 吕志峰 主编



清华大学出版社

21 世纪高等学校计算机应用技术规划教材

# Java 语言程序设计教程

邓 琨 吕志峰 主编  
赵蕴龙 张振蕊 张中华 副主编

清华大学出版社  
北 京

## 内 容 简 介

随着 Java 的应用越来越广泛,Java 程序设计已越来越多地受到广大计算机教育工作者的重视。Java 作为一种面向对象的程序设计语言,同时具备了平台无关性、高可靠性、多线程等特性,是 Internet 上最流行的编程语言之一,正被越来越多的用户了解和使用。本书紧密结合实际,循序渐进,总结优化了 Java 编程经验,每章都配有大量习题、典型实例及编程项目,生动详细地讲解了 Java 编程思想和编程语法,特点是简单明了,重点突出,使读者能够边学边练,激发学习兴趣。

本书共分 11 章,主要包括 Java 编程环境、Java 基本程序结构、对象和类、继承和多态、异常处理、输入输出、多线程程序设计、网络编程、Applet 编程、使用 JDBC 创建数据库的连接、JSP 技术基础等内容。本书既可作为高等院校计算机及相关专业程序设计课程的入门教材,也可作为程序设计公共选修课的教材。本书还适合接受职业教育或从事实际软件开发的读者。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

## 图书在版编目(CIP)数据

Java 语言程序设计教程/邓琨,吕志峰主编. —北京:清华大学出版社,2010.9

(21 世纪高等学校计算机应用技术规划教材)

ISBN 978-7-302-23583-5

I. ①J… II. ①邓… ②吕… 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 158945 号

责任编辑:索 梅 薛 阳

责任校对:梁 毅

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者:北京市世界知识印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:185×260 印 张:20.5 字 数:493 千字

版 次:2010 年 9 月第 1 版 印 次:2010 年 9 月第 1 次印刷

印 数:1~3000

定 价:29.50 元

---

产品编号:036635-01

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生的知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程可以有针对性、不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机应用技术规划教材

联系人:魏江江 weijj@tup.tsinghua.edu.cn

# 前言

自从 1995 年 Sun 公司推出了 Java 语言以来,Java 因其具有纯面向对象、高性能、分布式及稳定性好等优点而迅速风靡世界,尤其是 Java 的跨平台性、安全性等特点使得 Java 非常便于开发 Internet/Intranet 应用程序。Sun 公司在推出 Java 的同时,也推出了 Java 的开发工具 JDK,其中包括 Java 编译器、Java 解释器等开发程序时所需要的工具。

虽然在程序执行效率上 Java 确实不如 C 和 C++,但是在网络日益普及的今天,Java 具有它独特的魅力,它的作用不再局限于程序的开发,更为重要的是它提供了一个网络解决方案,Java 的内涵也不再局限于语言本身,还包括 Java 软件技术、Java 虚拟机、Java 嵌入技术等从软件到硬件的一系列概念。同时 Java 技术的应用范围不仅仅局限于信息领域,还涉及电子、信息综合服务、科学计算、软件生产等很多领域,有着广阔的应用前景,受到了各类用户的肯定和支持。

本书介绍了 Java 编程的基础知识。书中采用分步骤的教学方法,安排了许多示例、编程项目。本书不需要读者具备编程经验,从最基本的基础知识,诸如如何编译并运行一个 Java 程序开始讲起。接下来讨论了 Java 的每一个关键字,还介绍了 Java 的一些最重要的高级功能,如多线程编程和创建 Applet。最终学习者将会牢固地掌握 Java 编程精髓。

本书结合了编者多年的 Java 教学经验,具有较好的可操作性、实验性和先进性。本书既可作为高等院校计算机及相关专业程序设计课程的入门教材,也可作为程序设计公共选修课的教材。本书还适合接受职业教育或从事实际软件开发的读者。本书通俗易懂,简单明了,重点突出,并用各种事例来阐明一些难以理解或者易混淆的概念。读者可以边学边练,逐步加深和完善对 Java 核心技术的理解。

本书由邓琨策划编写,张中华、赵蕴龙编写了第 1、2、7 章,邓琨编写了第 3、4 章,张振蕊编写了第 5、6、8 章,吕志峰编写了第 9、10、11 章,另外参加编写和整理的还有姚璐、孙鹤、孙海龙、李晓峰、翟霞等。全书最后由邓琨统稿,本书在编写过程中得到了哈尔滨工业大学华德学院高洪志副教授的大力支持,在此表示诚挚的感谢。

在本书的编写过程中尽管编者尽了最大努力来避免错误的发生,但限于水平和时间,书中疏漏之处,恳请各位专家、读者批评指正,意见和建议可发邮件至 [hithddk@126.com](mailto:hithddk@126.com)。

编 者

2010 年 7 月

|                                |           |
|--------------------------------|-----------|
| <b>第 1 章 引言 .....</b>          | <b>1</b>  |
| 1.1 Java 语言的起源 .....           | 1         |
| 1.1.1 Java 发展历程 .....          | 1         |
| 1.1.2 Java 版本划分 .....          | 2         |
| 1.2 Java 语言的特点 .....           | 2         |
| 1.2.1 简单性 .....                | 2         |
| 1.2.2 面向对象 .....               | 3         |
| 1.2.3 平台无关性 .....              | 3         |
| 1.2.4 分布式 .....                | 3         |
| 1.2.5 安全稳定 .....               | 4         |
| 1.2.6 支持多线程 .....              | 4         |
| 1.2.7 内存垃圾自动回收 .....           | 5         |
| 1.3 Java 的开发环境 .....           | 5         |
| 1.3.1 JDK .....                | 5         |
| 1.3.2 MyEclipse .....          | 8         |
| 1.3.3 Java 的编码规范 .....         | 14        |
| 1.4 Java 程序的结构 .....           | 14        |
| 1.5 本章小结 .....                 | 16        |
| 习题 1 .....                     | 16        |
| <b>第 2 章 Java 基本程序结构 .....</b> | <b>17</b> |
| 2.1 Java 符号集 .....             | 17        |
| 2.1.1 标识符及其命名规则 .....          | 17        |
| 2.1.2 关键字 .....                | 18        |
| 2.1.3 注释 .....                 | 18        |
| 2.1.4 分隔符 .....                | 19        |
| 2.2 数据类型 .....                 | 19        |
| 2.2.1 基本数据类型 .....             | 20        |
| 2.2.2 引用数据类型 .....             | 21        |
| 2.2.3 数据类型的转换 .....            | 21        |
| 2.3 变量和常量 .....                | 22        |
| 2.3.1 变量 .....                 | 22        |

|              |                 |           |
|--------------|-----------------|-----------|
| 2.3.2        | 常量              | 25        |
| 2.4          | 运算符与表达式         | 27        |
| 2.4.1        | 算术运算符           | 27        |
| 2.4.2        | 关系运算符           | 29        |
| 2.4.3        | 逻辑运算符           | 31        |
| 2.4.4        | 位运算符            | 32        |
| 2.4.5        | 赋值运算符           | 32        |
| 2.4.6        | 其他运算符           | 33        |
| 2.4.7        | 运算符的优先级和结合      | 34        |
| 2.5          | 数组和集合类          | 34        |
| 2.5.1        | 数组的声明           | 35        |
| 2.5.2        | 数组的初始化          | 35        |
| 2.5.3        | 数组的引用           | 37        |
| 2.5.4        | 多维数组            | 38        |
| 2.5.5        | 集合类             | 41        |
| 2.6          | 字符串             | 44        |
| 2.6.1        | String 类        | 45        |
| 2.6.2        | StringBuffer 类  | 50        |
| 2.6.3        | StringBuilder 类 | 52        |
| 2.7          | 流程控制            | 53        |
| 2.7.1        | 条件控制语句          | 53        |
| 2.7.2        | 循环控制语句          | 57        |
| 2.7.3        | 跳转语句            | 59        |
| 2.8          | 本章小结            | 61        |
| 习题 2         |                 | 61        |
| <b>第 3 章</b> | <b>对象和类</b>     | <b>65</b> |
| 3.1          | 面向对象的基本概念       | 65        |
| 3.1.1        | 对象和类            | 65        |
| 3.1.2        | 数据抽象和封装         | 66        |
| 3.1.3        | 继承性             | 67        |
| 3.1.4        | 多态性             | 68        |
| 3.2          | 类               | 68        |
| 3.2.1        | 类的创建            | 68        |
| 3.2.2        | 类变量             | 69        |
| 3.2.3        | 类方法             | 71        |
| 3.2.4        | 构造方法            | 73        |
| 3.2.5        | main 方法         | 75        |
| 3.2.6        | 设计类的原则          | 75        |



|                          |            |
|--------------------------|------------|
| 3.3 对象 .....             | 76         |
| 3.3.1 对象的创建 .....        | 76         |
| 3.3.2 对象的使用 .....        | 78         |
| 3.3.3 对象的回收 .....        | 82         |
| 3.4 方法重载 .....           | 83         |
| 3.4.1 普通方法的重载 .....      | 83         |
| 3.4.2 构造方法的重载 .....      | 84         |
| 3.4.3 向方法传递对象 .....      | 85         |
| 3.5 this 对象 .....        | 87         |
| 3.6 信息隐藏 .....           | 89         |
| 3.6.1 包 .....            | 89         |
| 3.6.2 访问控制 .....         | 91         |
| 3.7 static 关键字 .....     | 94         |
| 3.7.1 静态成员的使用 .....      | 94         |
| 3.7.2 静态成员的初始化 .....     | 95         |
| 3.8 嵌套类 .....            | 97         |
| 3.9 本章小结 .....           | 99         |
| 习题 3 .....               | 99         |
| <b>第 4 章 继承与多态 .....</b> | <b>101</b> |
| 4.1 继承性 .....            | 101        |
| 4.1.1 子类的创建 .....        | 101        |
| 4.1.2 成员变量与方法的继承 .....   | 103        |
| 4.1.3 super 关键字的使用 ..... | 105        |
| 4.1.4 多态性 .....          | 107        |
| 4.2 final 关键字 .....      | 109        |
| 4.2.1 final 数据 .....     | 109        |
| 4.2.2 final 方法 .....     | 112        |
| 4.2.3 final 类 .....      | 113        |
| 4.3 类对象之间的类型转换 .....     | 113        |
| 4.4 抽象类 .....            | 114        |
| 4.5 接口 .....             | 116        |
| 4.5.1 接口的定义 .....        | 117        |
| 4.5.2 接口的实现和使用 .....     | 117        |
| 4.5.3 接口中的变量 .....       | 120        |
| 4.5.4 接口与抽象类 .....       | 121        |
| 4.6 本章小结 .....           | 121        |
| 习题 4 .....               | 121        |



|                             |     |
|-----------------------------|-----|
| <b>第 5 章 异常处理</b>           | 125 |
| 5.1 Java 的异常处理机制            | 125 |
| 5.1.1 异常处理的概念               | 125 |
| 5.1.2 异常类的层次                | 127 |
| 5.2 异常处理的结构                 | 128 |
| 5.2.1 try-catch 语句块         | 128 |
| 5.2.2 try-catch-finally 语句块 | 131 |
| 5.3 引发异常的环境和方法              | 133 |
| 5.3.1 throw 语句的使用           | 133 |
| 5.3.2 throws 语句的使用          | 135 |
| 5.4 定制异常                    | 138 |
| 5.5 本章小结                    | 140 |
| 习题 5                        | 140 |
| <b>第 6 章 输入和输出</b>          | 141 |
| 6.1 概述                      | 141 |
| 6.1.1 流的概念                  | 141 |
| 6.1.2 I/O 流类概述              | 141 |
| 6.2 字节数据流                   | 142 |
| 6.2.1 字节输出流                 | 142 |
| 6.2.2 字节输入流                 | 144 |
| 6.3 字符数据流                   | 147 |
| 6.4 其他流类                    | 150 |
| 6.4.1 缓冲流                   | 150 |
| 6.4.2 随机文件读写                | 151 |
| 6.5 本章小结                    | 153 |
| 习题 6                        | 153 |
| <b>第 7 章 多线程程序设计</b>        | 155 |
| 7.1 理解线程                    | 155 |
| 7.1.1 什么是线程                 | 155 |
| 7.1.2 进程与线程                 | 155 |
| 7.1.3 线程的状态与生命周期            | 156 |
| 7.1.4 并发编程的优缺点              | 158 |
| 7.2 Java 的基本线程              | 159 |
| 7.2.1 Thread 的简介            | 159 |
| 7.2.2 创建线程(Thread)          | 160 |
| 7.2.3 创建线程(Runnable)        | 162 |

|              |                                 |            |
|--------------|---------------------------------|------------|
| 7.3          | 线程的基本操作 .....                   | 163        |
| 7.3.1        | 暂停线程 .....                      | 163        |
| 7.3.2        | 休眠线程 .....                      | 164        |
| 7.3.3        | 连接线程 .....                      | 166        |
| 7.3.4        | 守护线程 .....                      | 167        |
| 7.3.5        | 线程的优先级 .....                    | 169        |
| 7.4          | 线程的高级操作 .....                   | 171        |
| 7.4.1        | 线程间共享资源 .....                   | 171        |
| 7.4.2        | 死锁 .....                        | 175        |
| 7.4.3        | 停止线程的正确方法 .....                 | 179        |
| 7.4.4        | 打断受阻的线程 .....                   | 180        |
| 7.4.5        | 线程间协作 .....                     | 181        |
| 7.4.6        | 管道 .....                        | 185        |
| 7.5          | 本章小结 .....                      | 186        |
| 习题 7         | .....                           | 187        |
| <b>第 8 章</b> | <b>网络编程 .....</b>               | <b>188</b> |
| 8.1          | URL 通信 .....                    | 188        |
| 8.1.1        | URL 的概念 .....                   | 188        |
| 8.1.2        | URL 类 .....                     | 189        |
| 8.1.3        | 使用 URLConnection 类访问网络资源 .....  | 192        |
| 8.2          | Socket 通信 .....                 | 194        |
| 8.2.1        | Socket 通信基础 .....               | 194        |
| 8.2.2        | Socket 通信程序的开发 .....            | 194        |
| 8.2.3        | Socket 通信与 URL 通信的区别 .....      | 198        |
| 8.3          | UDP 通信 .....                    | 198        |
| 8.3.1        | UDP 通信基础 .....                  | 199        |
| 8.3.2        | UDP 通信程序的开发 .....               | 199        |
| 8.4          | 本章小结 .....                      | 203        |
| 习题 8         | .....                           | 203        |
| <b>第 9 章</b> | <b>GUI 编程与 Applet 小程序 .....</b> | <b>204</b> |
| 9.1          | 图形用户接口概述 .....                  | 204        |
| 9.1.1        | AWT 简介 .....                    | 204        |
| 9.1.2        | 组件 .....                        | 205        |
| 9.1.3        | 容器 .....                        | 205        |
| 9.1.4        | 布局管理 .....                      | 206        |
| 9.1.5        | GUI 程序的结构 .....                 | 207        |
| 9.2          | 事件处理 .....                      | 208        |

|               |                          |            |
|---------------|--------------------------|------------|
| 9.2.1         | 事件处理模型 .....             | 208        |
| 9.2.2         | 事件类 .....                | 209        |
| 9.2.3         | 事件监听器 .....              | 210        |
| 9.2.4         | 事件适配器 .....              | 212        |
| 9.3           | 常用组件的使用 .....            | 214        |
| 9.3.1         | 按钮与标签 .....              | 214        |
| 9.3.2         | 文本框和文本区 .....            | 216        |
| 9.3.3         | 选择型组件 .....              | 219        |
| 9.3.4         | Java 窗口 .....            | 221        |
| 9.4           | Applet 编程 .....          | 222        |
| 9.4.1         | Applet 程序的基本结构 .....     | 222        |
| 9.4.2         | Applet 类 .....           | 224        |
| 9.4.3         | Applet 生命周期 .....        | 225        |
| 9.4.4         | HTML 语言的 applet 标记 ..... | 229        |
| 9.4.5         | Applet 参数的读取 .....       | 230        |
| 9.4.6         | Applet 应用编程实例 .....      | 231        |
| 9.5           | 本章小结 .....               | 236        |
| 习题 9          | .....                    | 237        |
| <b>第 10 章</b> | <b>JDBC 应用 .....</b>     | <b>238</b> |
| 10.1          | JDBC 概述 .....            | 238        |
| 10.2          | 创建数据库连接 .....            | 242        |
| 10.2.1        | 加载驱动程序 .....             | 242        |
| 10.2.2        | 建立连接 .....               | 242        |
| 10.2.3        | 关闭连接 .....               | 244        |
| 10.3          | 数据源操作程序设计 .....          | 244        |
| 10.3.1        | 查询操作 .....               | 245        |
| 10.3.2        | 更新操作 .....               | 251        |
| 10.4          | 访问数据库元信息 .....           | 255        |
| 10.5          | JDBC 操作其他类型数据库 .....     | 260        |
| 10.6          | 本章小结 .....               | 261        |
| 习题 10         | .....                    | 261        |
| <b>第 11 章</b> | <b>JSP 技术基础 .....</b>    | <b>262</b> |
| 11.1          | JSP 技术概述 .....           | 262        |
| 11.1.1        | JSP 简介 .....             | 263        |
| 11.1.2        | JSP 的运行环境配置 .....        | 263        |
| 11.1.3        | JSP 文件基本结构 .....         | 267        |
| 11.2          | JSP 语法 .....             | 268        |

|        |                        |     |
|--------|------------------------|-----|
| 11.2.1 | JSP 声明语法 .....         | 268 |
| 11.2.2 | 程序段 .....              | 272 |
| 11.2.3 | 表达式 .....              | 274 |
| 11.2.4 | JSP 中的注释 .....         | 274 |
| 11.2.5 | JSP 指令标签 .....         | 275 |
| 11.3   | JSP 内置对象 .....         | 279 |
| 11.3.1 | request 对象 .....       | 279 |
| 11.3.2 | response 对象 .....      | 283 |
| 11.3.3 | out 对象 .....           | 284 |
| 11.3.4 | session 对象 .....       | 286 |
| 11.3.5 | application 对象 .....   | 288 |
| 11.3.6 | 其他内置对象 .....           | 289 |
| 11.4   | JSP 标准动作 .....         | 291 |
| 11.4.1 | include 动作标签 .....     | 291 |
| 11.4.2 | forward 动作标签 .....     | 292 |
| 11.4.3 | useBean 动作标签 .....     | 294 |
| 11.5   | 本章小结 .....             | 300 |
| 习题 11  | .....                  | 300 |
| 附录A    | Java 编程规则 .....        | 301 |
| A.1    | 应用范围 .....             | 301 |
| A.2    | 设计类和方法 .....           | 301 |
| A.2.1  | 创建具有很强内聚力的类 .....      | 301 |
| A.2.2  | 创建松散连接和高度专用的方法 .....   | 301 |
| A.2.3  | 编程原则 .....             | 302 |
| A.3    | 命名约定 .....             | 303 |
| A.3.1  | 包、类及方法命名 .....         | 303 |
| A.3.2  | 其他 .....               | 304 |
| A.4    | 使用常量 .....             | 305 |
| A.5    | 变量 .....               | 305 |
| A.5.1  | 定义有焦点的变量 .....         | 305 |
| A.5.2  | 只对常用变量名和长变量名进行缩写 ..... | 305 |
| 参考文献   | .....                  | 309 |

# 第1章

## 引言

任何一种语言都有它的起源、发展历程、特点以及开发工具,Java 语言也不例外。本章首先介绍 Java 的起源和 Java 的特点,详细讲解 Java 开发工具(Java Develop Kit,JDK)的环境变量配置和使用方法,接着介绍一款集成开发工具——Eclipse,并讲解怎样使用 Eclipse 工具来开发 Java 应用程序,最后简单介绍编写 Java 程序要遵循的 Java 编码约定和 Java 程序基本结构。经过对本章的学习,读者将会对 Java 语言有一个全面的了解。

### 1.1 Java 语言的起源

Java 起源于 20 世纪 90 年代初 Sun 公司的一个叫 Green 的项目,该项目的目标是开发嵌入家用电器的分布式软件系统,使电器更加智能化。项目小组成员皆为 C++ 的高手,便采用 C++ 进行系统开发,但在开发过程中发现了很多问题,如 C++ 语言过于复杂、安全性差等。于是项目小组只好另辟蹊径,在“简单的、可靠的、紧凑的并易于移植的”框架内开发了小型的计算机语言,用于编写消费者的电子产品中的程序,当时这款语言命名为 Oak(橡树的意思)。至于这 Oak 怎么变成了 Java,请看下面的 Java 发展历程。

#### 1.1.1 Java 发展历程

1991 年,Sun 公司成立 Green 项目,Oak 语言在此项目中诞生;18 个月后,项目的第一个成果是可触摸控制的手持家庭娱乐设备控制器,其名为 \* 7(star seven),随后 Sun 公司为 Green 项目成立 First Person 公司;1993 年,因为当时 WWW 开始席卷全球,First Person 公司重新定位 Oak 语言,将其转向网络应用领域;1994 年,Oak 技术包括源代码全部免费公开,使其成为开源技术,并用 Oak 语言开发出网络浏览器 Webrunner,后更名为 HotJava。而且 Oak 编写的 Applet 也让网页由静态转成动态;1995 年,First Person 公司对 Oak 注册登记时,发现该名已被占用,于是将 Oak 更名为 Java。随后 First Person 公司解散。1995 年 3 月 23 日是 Java 的里程碑,Java 被评为十大优秀科技产品之一,Java 开发工具 JDK 1.0a2 版本正式对外发布;1996 年,Sun 公司成立 Javasoft 分公司来发展 Java,接着 Netscape 浏览器开始支持 Java,IBM、Symantec、Inprise、Microsoft 公司的 IE 浏览器也开始支持 Java,众多第三方的 Java 编译器也被推出,例如 VJ++、Builder 等;1997 年,JDK 1.1 版本发布;1998 年,JDK 1.2 发布,JDK 更名为 J2SDK,其全称为 Java 2 Software Development Kit,J2SDK 1.2 又名 Java 2,并将 Java 的应用进行细分;2000 年,J2SDK 1.3

发布; 2002 年, J2SDK 1.4 发布; 2004 年, J2SDK 1.5 发布; 2005 年, J2SDK 1.6 发布。

### 1.1.2 Java 版本划分

Java 的版本演进到 Java 2, 根据不同层面的应用进行了细化, Java 2 平台被分为三种版本。

- J2EE Java 2 Enterprise Edition——企业版, 适用于服务器, 目前已成为企业运算、电子商务等领域的热门技术。
- J2SE Java 2 Standard Edition——标准版, 适用于一般的计算机, 开发 PC 上的应用软件。
- J2ME Java 2 Micro Edition——微型版, 适用于手持设备, 进行应用开发, 如手机游戏、名片管理等。

本书以 J2SE 展开, 讨论 Java 语言程序设计。J2SDK 1.6 的标准版又名 Java SE 6。

## 1.2 Java 语言的特点

Java 是定位于网络计算的计算机语言, 它几乎所有的特点都是围绕这一中心展开并为之服务的, 这些特点使得 Java 语言特别适合用来开发网络上的应用程序。

Java 从 C++ 发展而来, 对于当今世界上众多的 C++ 程序员来说, Java 显得并不陌生。Java 是一种面向对象的编程语言, 甚至比 C++ 更为“面向对象”。目前面向对象技术已经取代早期的结构化程序设计方法而成为 IT 界的标准技术, 因为事实证明面向对象技术处理复杂问题的优势远非其他方法所能及。

Java 并不仅仅是 C++ 语言的一个变种, 而在某些本质问题上与 C++ 有根本的不同。

(1) Java 比起 C++ 来说, 程序可靠性更高。有人曾估计每 50 行 C++ 程序中至少有一个 bug。姑且不去讨论这个数字是否夸张, 但是任何一个 C++ 程序员都不得不承认 C++ 语言在提供强大的功能的同时也提高了程序含 bug 的可能性。Java 语言通过改变语言的特性大大提高了程序的可靠性。

(2) Java 语言不需要程序对内存进行分配和回收。在 Java 语言中, 内存的分配和回收都是自动进行的, 不用程序员考虑内存碎片的问题。

(3) Java 语言中没有“指针”的概念, 引入了真正的数组。不同于 C++ 中利用指针实现的“伪数组”, Java 引入了真正的数组, 同时将容易造成麻烦的指针从语言中去掉, 这将有利于防止在 C++ 程序中常见的因为数组操作越界等指针操作而对系统数据进行非法读写带来的不安全问题的发生。

(4) Java 用接口 (Interface) 技术取代 C++ 程序中的多继承性。接口与多继承有同样的功能, 但是省却了多继承在实现和维护上的复杂性。

### 1.2.1 简单性

Java 与 C++ 极为相似, 但却简单得多。Java 的语法实际上可以看作是 C++ 语法的一个净化版。高级编程语言的所有特性中, 不是绝对需要的都已删去了。例如, Java 没有运算

符重载、头文件、预处理、指针运算、结构、联合、模板及隐式类型变换。但是对于 C++ 中其他的一些并不太好的语法特色,Java 并没有作改进,如 switch 语句。总的来说,如果懂得一点 C、C++ 或 Pascal,很快就会驾驭 Java,从 C++ 程序员转变为 Java 的程序员并不太难。

同时 Java 还有一个特点,就是它的基本语法与 C 语言几乎一模一样,如果懂得 C 语言,学 Java 语言将变得非常简单。

### 1.2.2 面向对象

面向对象技术较好地适应了当今软件的开发,克服了传统的面向过程语言编程所存在的问题,适应了软件开发规模扩大、升级加快以及维护量增大等要求,是一种迅速成熟并被推广的软件开发方法。

面向对象技术的核心是以更接近于人类思维的方式建立计算机逻辑模型,它利用类和对象的机制,将数据和操作封装在一起,通过统一的接口和外界交互,反映出现实世界的各个类在程序中能够独立、自治和继承。面向对象技术有利于提高程序的可维护性和可重用性,大大提高开发效率和程序的可管理性。

Java 是面向对象的编程语言。Java 是从 C++ 语言发展而来的,而 C++ 是从 C 语言发展而来的,C++ 语言兼容 C 语言,那么 C++ 不是完全面向对象编程的语言,可以面向过程编程,而 Java 则基本上做到了完全面向对象,在 Java 中,除了基本的数据类型以外,如整型、浮点型等,任何一个基本元素都可以看作一个对象。因此 Java 语言的出现,大大提高了程序的可维护性和可重用性。

### 1.2.3 平台无关性

所谓平台无关性,是指应用程序与计算机系统结构的无关性,更直观地说就是用 Java 编写的应用程序不用修改就可在不同的软硬件平台同样地运行,如在 Windows 下编写的程序,可以在 UNIX 下运行,这就称为平台无关性,亦称为跨平台。也可用一句话概括,就是一次编写,到处运行。

Java 主要靠 Java 虚拟机(JVM)在目标码级实现平台无关性。JVM 是一种抽象机器,它附着在具体操作系统之上,本身具有一套虚拟机指令,并有自己的栈、寄存器组等。但 JVM 通常是在软件上而不是在硬件上实现的。JVM 是 Java 平台无关的基础,在 JVM 上,有一个 Java 解释器用来解释 Java 编译器编译后的程序。Java 编程人员在编写完软件后,通过 Java 编译器将 Java 源程序编译为 JVM 的字节代码。任何一台机器只要配备了 Java 解释器,就可以运行这个程序,而不管这种字节码是在何种平台上生成的。另外,Java 采用的是基于 IEEE 标准的数据类型。JVM 保证了数据类型的一致性,也确保了 Java 的平台无关性。

### 1.2.4 分布式

分布式包括数据分布和操作分布。数据分布是指数据可以分散在网络的不同主机上,操作分布是指把一个计算分散在不同主机上处理。

Java 支持 WWW 客户机/服务器计算模式,因此,它支持这两种分布性。对于前者,



Java 提供了一个叫做 URL 的对象,利用这个对象,可以打开并访问具有相同 URL 地址上的对象,访问方式与访问本地文件系统相同。对于后者,Java 的 Applet 小应用可以从服务器下载到客户端,即部分计算在客户端进行,提高系统执行效率。

Java 有着丰富的类库,可以很方便地编写 TCP/IP 协议上的应用程序。Java 应用程序可以由网络通过 URL 调用远程对象,与操作本地文件系统没有区别。

Java 提供了一整套网络类库,开发人员可以利用类库进行网络程序设计,方便地实现 Java 的分布式特性。Java 语言的网路功能非常强大,而且很简单,如果曾使用过其他的语言编写过网络应用程序的话,就会惊讶于用 Java 生成一个 Socket 竟然这样简单。运用 Java 写 CGI 脚本也很简单。

### 1.2.5 安全稳定

对于编程语言,尤其是网络编程语言,与跨平台性相随而来的还有另一大特性:安全性。在网络上,如果网络语言不十分可靠,极有可能被人有意或无意地形成灾难:也许文件会在硬盘上奇怪地消失;也许发出去的文件又莫名其妙地存在自己的服务器上;也许内存会突然不足,以致运行的程序也越来越慢,甚至突然中断;也许秘密被人在不知不觉中窃取;更有甚者,系统无端地崩溃。

Java 被设计成易于在网络/分布环境下运行,因此在安全性上作了很多的考虑。Java 可以避免计算机病毒的侵害,也能够防止对系统的意外损害。

作为网络上的应用程序,安全性是至关重要的。为防止利用 Java 程序攻击其他人的系统,Java 主要在编译和运行时在系统中采取了防护措施:编译程序保证源代码不违反安全规则;运行时系统校验字节代码,类装入程序保证类不侵犯名字空间和访问限制的范围,运行时进行内存分配以使潜在的破坏者无法预知内存的布局,对文件的访问加以限制(当从服务器传过来的 Java 程序被像 Netscape Navigator 这样的客户端程序激活时,它不能对本地的文件进行读或写)。这其中,字节代码校验是一个关键部分。Java 字节码的一个特性是它的可解析性能,能够对其进行分析并对其以后的行为做出推理,这是 Java 安全系统的基础之一。任何代码在执行以前,都要经过一系列严格而复杂的测试。通过测试,可以保证校验字节代码不非法地避免进行类型转换(如伪造指针)、不以非法形式访问对象、操作数堆栈不上溢或下溢、错误地使用参数、变量和类型等。总之,Java 的编译和运行环境为 Java 代码筑起了两道坚实的防护屏障。

### 1.2.6 支持多线程

Java 的另一大特性就是支持多线程式机制。多线程是当今软件技术的又一个重要成果,已经成功应用在操作系统、应用程序开发等多个领域。当今的操作系统都支持多线程,把线程作为自己基本的执行单位。

在 Java 编程语言设计时,已将多线程的功能考虑进去了,语言定义就提供了多线程的功能。Java 编程语言中的基本方法库中已定义了 Thread 这个基本类,内置了一组方法,使程序设计者编写多线程程序时,只要继承这个类,就可以利用原来编写的方法,生成一个新的线程,执行一个线程,终止一个线程,或者查看执行状态。除了扩充 Thread 类,Java 中还