



普通高等教育“十一五”国家级规划教材
高职高专计算机类专业教学改革规划教材

COMPUTER



Java进阶教程

第2版

刘志成 张杰 主编



机械工业出版社
CHINA MACHINE PRESS

赠电子课件

普通高等教育“十一五”国家级规划教材
高职高专计算机类专业教学改革规划教材

Java 进阶教程

第 2 版

主 编 刘志成 张 杰
副主编 于训全 李振军 谢树新
参 编 宁云智 龚 娟 薛志良
吴献文 贺新建 刘毅红



机械工业出版社

Java 语言是当前最流行的程序设计语言。本书以 Java 最新版本 Java SE6 和 Netbeans IDE 6.1 开发工具为载体,按照“语言基础”、“技术基础”和“高级应用”3个层次由浅入深地介绍了 Java SE6 的核心技术。主要内容包括:语言基础篇(Java 语言概述、数据类型与运算符、流程控制语句);技术基础篇(面向对象编程技术、GUI 编程技术、异常处理技术、输入/输出技术);高级应用篇(数据库编程、线程与多线程编程、网络编程)。

作者在多年开发与教学经验的基础上,根据 Java 桌面开发程序员的岗位能力要求和学生的认知规律精心组织了本书内容。书中通过 87 个典型的实例详细介绍了 Java 桌面开发相关技术,并精心设计了“教学导航”、“课堂模仿”、“课外拓展”和“单元实训”等环节,将知识讲解和技能训练有机结合,融“教、学、练”于一体,适合“项目驱动、案例教学、理论实践一体化”的教学方法。同时提供书中各章的电子课件和所有环节的源代码,供读者参考使用。

本书可作为高职高专软件技术专业、网络技术专业的教材,也可作为计算机爱好者和程序员自学教材和参考用书。

图书在版编目(CIP)数据

Java 进阶教程/刘志成,张杰主编. —2 版. —北京:机械工业出版社, 2008.11

普通高等教育“十一五”国家级规划教材
高职高专计算机类专业教学改革规划教材
ISBN 978-7-111-14863-0

I. J… II. ①刘…②张… 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(2008)第 186487 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
策划编辑:王玉鑫 责任编辑:王玉鑫 张 芳
版式设计:霍永明 责任校对:张 薇
封面设计:王伟光 责任印制:李 妍
北京蓝海印刷有限公司印刷
2009 年 2 月第 2 版第 1 次印刷
184mm × 260mm · 22.75 印张 · 565 千字
0001—4000 册
标准书号:ISBN 978-7-111-14863-0
定价:35.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换
销售服务热线电话:(010) 68326294
购书热线电话:(010) 88379639 88379641 88379643
编辑热线电话:(010) 88379543
封面无防伪标均为盗版

第2版前言

Java是当前最流行的程序设计语言之一，它的出现大大促进了软件产业和互联网的发展。从1995年Java诞生以来，它已从一种编程语言发展为一个平台、一个产业。Java作为一种优秀的面向对象程序设计语言，已成为软件开发领域中的主流技术，全球约有450万名程序员使用Java开发软件，14亿部设备上运行着Java编写的程序。

本书是作者在第1版的基础上，经过几年的教学实践，总结了多年的开发经验与成果编写的。全书通过87个典型的实例，按照“语言基础”、“技术基础”和“高级应用”3个层次由浅入深、由易到难地介绍了Java SE6的核心技术。通过本书的学习，读者可以快速、全面地掌握使用Java SE6技术开发桌面应用程序的方法。

与第1版相比，本书主要突出了以下特点：

(1) 知识技能阶梯式推进 根据高职学生的认知规律和Java桌面开发程序员岗位能力的要求，将Java SE6知识体系合理地划分为“语言基础篇”、“技术基础篇”和“高级应用篇”螺旋推进式的三大部分内容；每一章的内容通过“课堂模仿”和“课外拓展”等环节实现编程技能的逐步提高。

(2) 理论实践一体化 根据编者多年的Java语言教学经验，合理设计教师“教”和学生“学”的环节，将知识讲解和技能训练设计在同一教学单元和教学地点完成，融“教、学、练”于一体。同时，所使用的典型实例源于开发实践，体现了“学为所用，学以致用”的教学理念。

(3) 强化职业技能训练和素质训练 本书精心设置了3个实践环节：课堂模仿、课外拓展和单元实训。让读者通过不断的编程实践，实现编程技能的逐步提升和职业态度及职业素质的养成，最终具备较强的职业能力。

本书共分10章，具体如下：

第1章：Java语言概述。介绍了Java语言的特点、JDK的版本与选择、Java桌面开发环境的构建、Java程序编写/编译和运行的过程、Eclipse和NetBeans开发工具的使用。

第2章：数据类型与运算符。介绍了Java符号、常量与变量、基本数据类型、数据类型转换、运算符与表达式、数组。

第3章：流程控制语句。介绍了if语句和switch语句、for循环、while循环和do-while循环、break语句、continue语句。

第4章：面向对象编程技术。介绍了面向对象程序设计的基本概念、Java语言中声明类的方法、由类创建对象并使用对象的方法、Java语言中继承的实现方法、在Java语言应用接口实现多态、Java常用类库。

第5章：GUI编程技术。介绍了Java GUI概述、常用容器、常用GUI组件、高级GUI组件、常用布局管理器、Java事件处理、在NetBeans中开发GUI程序。

第6章：异常处理技术。介绍了异常的定义和异常的类型、Java语言中的异常类层次结

构、Java 语言中的异常处理机制、自定义异常类的定义和使用以及与异常处理相关的知识内容。

第 7 章：输入/输出技术。介绍了 Java 输入输出的基本概念、File 类的应用、Random-AccessFile 类的应用、标准输入输出、字节流类的应用、字符流类的应用、对象序列化。

第 8 章：数据库编程。介绍了 JDBC 的概念、JDBC 应用程序开发流程、应用 JDBC 实现数据库记录的查询、应用 JDBC 实现数据库记录的插入/删除和修改操作、数据库元数据的操作。

第 9 章：线程与多线程编程。介绍了线程的概念、线程与进程的区别、线程的状态及改变线程状态的方法、使用 Thread 类创建线程、使用 Runnable 接口创建线程、线程的优先级和线程的调度。

第 10 章：网络编程。介绍了 C/S 网络编程模型、网络编程相关概念、使用 URL 进行网络编程、使用 Socket 进行网络编程。

本书由湖南铁道职业技术学院刘志成和张杰主编，湖南铁路科技职业技术学院的于训全、湖南网络工程职业学院的李振军、湖南铁道职业技术学院的谢树新任副主编，湖南铁道职业技术学院的宁云智、龚娟、薛志良、吴献文，株洲时菱交通设备有限公司的贺新建工程师和株洲时代电工厂的刘毅红工程师参编。

本书可作为高职高专软件技术专业、网络技术专业的教材，也可作为计算机爱好者和程序员自学教材和参考用书。

为方便教学，本书配备电子课件等教学资源。凡选用本书作为教材的教师均可登录机械工业出版社教材服务网 www.cmpedu.com 免费下载。如有问题请致信 cmpgaozhi@sina.com 或致电 010-88379375 咨询。

由于时间仓促以及编者水平有限，书中难免存在错误和疏漏之处，欢迎广大读者和同仁提出宝贵意见和建议。

编者

第1版前言

Java 是面向对象的编程语言，它是目前 Internet 最流行的编程语言。由于使用 Java 语言编写程序具有编程简单、功能强大、安全性高等特点，所以 Java 语言被广泛应用于 Internet 中。

本书的编写有别于一般的 Java 语言编程教材。由于大部分的院校是以 C 语言作为入门级语言，而 Java 语言与 C 语言在语法上有较大的类似性，因此本书并不是简单地重复介绍 Java 语言的基础知识，而是从 Java 的体系结构与各项编程技术的实现方法上进行介绍。对每一种编程技术，一步一步地详细介绍其实现方法，并配有实例，这样，读者在学完相关内容后就可以掌握相关的编程技术。本书意在通过丰富、实用的实例讲解，深入浅出地介绍 Java 语言编程的技术方法，重点培养学生解决实际问题的能力，真正起到提高学生编程能力的作用。

本书共 10 章，由湖南铁道职业技术学院张杰任主编，广东轻工职业技术学院古凌岚任副主编，广东轻工职业技术学院罗佳、河南济源职业技术学院王树森、湖南铁道职业技术学院唐吉参编。张杰负责第 7、9 章的编写，古凌岚负责第 5、6、10 章的编写，罗佳负责第 4、8 章的编写，王树森负责第 1、2 章的编写，唐吉负责第 3 章的编写。本书由彭孟良负责主审。

本书配有大量的实例，所有的实例都是经过作者实际测试通过的，可以作为学生上机实训的内容。

本书可以作为高职高专软件技术专业 Java 语言程序设计的教材，也可作为计算机相关专业的参考书。

由于时间仓促以及编著者水平有限，书中难免有错误和不妥之处，敬请读者批评指正。

编 者
2004 年 3 月

目 录

第2版前言

第1版前言

语言基础篇

第1章 Java语言概述	1
1.1 Java语言简介	2
1.1.1 Java语言的发展历程	2
1.1.2 Java的版本历史	2
1.1.3 Java语言与C/C++	4
1.1.4 Java的特点	5
1.2 JDK和Java开发环境及工作原理	6
1.2.1 JDK	6
1.2.2 Java开发环境	7
1.2.3 Java工作原理	10
1.3 第一个Java程序	11
1.3.1 Java Application程序	11
1.3.2 Java Applet程序	13
1.3.3 技术要点	15
1.4 Java开发工具简介	16
1.4.1 Eclipse	16
1.4.2 NetBeans	21
第2章 数据类型与运算符	28
2.1 Java符号	29
2.1.1 关键字和标识符	29
2.1.2 运算符和分隔符	30
2.1.3 注释	30
2.2 常量与变量	31
2.2.1 常量	31
2.2.2 变量	32
2.3 基本数据类型	34
2.3.1 整数类型	35
2.3.2 实数类型	35

2.3.3 布尔类型	36
2.3.4 字符类型	36
2.4 数据类型转换	38
2.4.1 自动类型转换	38
2.4.2 强制类型转换	39
2.5 运算符	40
2.5.1 运算符与表达式	40
2.5.2 运算符的优先级	43
2.6 数组	45
2.6.1 一维数组	46
2.6.2 二维数组	48
第3章 流程控制语句	54
3.1 分支语句	54
3.1.1 if语句	54
3.1.2 switch语句	61
3.2 循环语句	64
3.2.1 for语句	64
3.2.2 while语句	66
3.2.3 do-while语句	67
3.2.4 循环语句嵌套	69
3.3 跳转语句	72
3.3.1 break语句	72
3.3.2 continue语句	75
单元实训一 数组排序	82

技术基础篇

第4章 面向对象编程技术	84
4.1 面向对象概述	84
4.1.1 面向对象的基本概念	84
4.1.2 面向对象的基本特性	86
4.2 Java语言中的类	88
4.2.1 类的定义	88
4.2.2 成员变量定义	91

4.2.3 成员方法定义	92	5.3.4 列表框和组合框	145
4.2.4 构造方法	94	5.3.5 菜单和工具栏	148
4.2.5 垃圾回收	96	5.4 高级 GUI 组件	154
4.3 对象	96	5.4.1 对话框	154
4.3.1 对象的特性	96	5.4.2 表格	155
4.3.2 创建对象	96	5.4.3 树	159
4.3.3 调用对象的成员	97	5.5 布局管理	162
4.4 继承	98	5.5.1 FlowLayout	162
4.4.1 类的继承	98	5.5.2 GridLayout	163
4.4.2 this 和 super	101	5.5.3 BorderLayout	163
4.4.3 方法重载与方法重写	104	5.5.4 CardLayout	163
4.4.4 抽象类与抽象方法	108	5.5.5 GridBagLayout	165
4.4.5 final 修饰符	111	5.5.6 空布局	169
4.5 接口	112	5.6 事件处理	171
4.5.1 接口定义	112	5.6.1 Java 事件模型	171
4.5.2 实现接口	114	5.6.2 事件类型	172
4.5.3 接口应用与多态	118	5.6.3 AWT 事件及其相应的 监听器接口	173
4.6 包与 Java 类库	119	5.6.4 Swing 事件及其相应的 监听器接口	175
4.6.1 定义包	119	5.6.5 典型事件处理程序	176
4.6.2 引入包	120	5.7 在 NetBeans 中开发 GUI 程序	184
4.6.3 Java 常用类库简介	121	5.7.1 新建 JFrame 窗体	184
第 5 章 GUI 编程技术	126	5.7.2 设计窗体	186
5.1 Java GUI 概述	126	5.7.3 添加事件处理	187
5.1.1 JFC 简介	127	5.7.4 编译运行程序	187
5.1.2 AWT 简介	127	第 6 章 异常处理技术	192
5.1.3 Swing 简介	128	6.1 异常概述	192
5.1.4 SWT 简介	129	6.1.1 异常的定义	193
5.1.5 Java 图形用户界面设计的 一般过程	129	6.1.2 异常类层次结构	194
5.2 常用容器	130	6.2 Java 语言中的异常处理	196
5.2.1 框架	130	6.2.1 异常处理机制	196
5.2.2 面板	131	6.2.2 声明异常	197
5.2.3 小程序	133	6.2.3 抛出异常	199
5.2.4 Swing 容器	134	6.2.4 捕获和处理异常	201
5.3 常用 GUI 组件	135	6.3 自定义异常	205
5.3.1 标签和按钮	135	6.4 异常处理拾遗	208
5.3.2 单行文本框和多行 文本框	138	6.4.1 错误/异常类型	208
5.3.3 复选框和单选按钮	144		

6.4.2 异常转型和异常链	209	7.6.5 PrintWriter 类	244
6.4.3 Java 异常处理的原则 和技巧	210	7.7 对象的序列化	246
第7章 输入/输出技术	214	单元实训二 简易记事本	251
7.1 Java 输入/输出概述	215	高级应用篇	
7.1.1 Java 输入/输出简介	215	第8章 数据库编程	255
7.1.2 Java 输入/输出类层次 结构	215	8.1 JDBC	256
7.2 File 类	217	8.1.1 JDBC 概述	256
7.2.1 File 类概述	217	8.1.2 JDBC 驱动程序	256
7.2.2 JFileChooser 类	218	8.1.3 创建示例数据库	259
7.2.3 File 类和 JFileChooser 类的 应用实例	219	8.1.4 创建 ODBC 数据源	261
7.3 RandomAccessFile 类	222	8.2 JDBC 编程的基本步骤	263
7.3.1 RandomAccessFile 类 概述	222	8.2.1 数据库 URL	263
7.3.2 RandomAccessFile 类 应用实例	224	8.2.2 加载驱动程序并建立 连接	263
7.4 标准输入/输出	227	8.2.3 执行 SQL 语句	268
7.4.1 标准输入/输出概述	227	8.3 数据库的基本操作	269
7.4.2 标准输入/输出应用实例 ..	227	8.3.1 数据查询	269
7.5 字节流类	228	8.3.2 数据添加/删除/修改	272
7.5.1 InputStream 类和 OutputStream 类	228	8.3.3 使用存储过程	277
7.5.2 FileInputStream 类和 FileOutputStream 类	229	8.4 数据库元数据操作	281
7.5.3 BufferedInputStream 类和 Buffered OutputStream 类	232	第9章 线程与多线程编程	286
7.5.4 ByteArrayInputStream 类和 ByteArrayOutputStream 类 ..	234	9.1 概述	286
7.5.5 PrintStream 类	236	9.1.1 线程基础	287
7.6 字符流类	238	9.1.2 线程的状态	288
7.6.1 Reader 类和 Writer 类	238	9.2 创建线程	289
7.6.2 FileReader 类和 FileWriter 类	239	9.2.1 扩展 Thread 类创建线程	289
7.6.3 BufferedReader 类和 BufferedWriter 类	240	9.2.2 实现 Runnable 接口创建 线程	292
7.6.4 InputStreamReader 类和 OutputStream Writer 类	242	9.3 实现多线程	296
		9.3.1 创建多线程	296
		9.3.2 使用 isAlive () 和 join () 方法	298
		9.3.3 线程的暂停和恢复	301
		9.3.4 线程的优先级	304
		9.4 线程的应用	306
		9.4.1 线程的同步	306
		9.4.2 线程的死锁	308
		9.4.3 线程在动画中的应用	309

第 10 章 网络编程	318	10.3.2 Socket 类和 ServerSocket 类	329
10.1 网络编程基础	318	10.3.3 基于 Socket 的 C/S 程序	331
10.1.1 C/S 模式和 B/S 模式	318	10.3.4 Server 和一个客户的 通信	331
10.1.2 TCP/IP	319	10.3.5 Server 和多个客户的 通信	334
10.1.3 IP 地址和 InetAddress 类	320	10.4 数据报编程	338
10.2 URL 编程	323	10.4.1 数据报通信概述	338
10.2.1 URL 的概念	323	10.4.2 简单聊天吧的实现	339
10.2.2 URL 的组成	323	单元实训三 快乐聊天室	345
10.2.3 从 URL 读取 WWW 网络资源	325	附录 本书实例一览表	349
10.2.4 通过 URLConnection 连接 WWW	326	参考文献	353
10.3 Socket 编程	329		
10.3.1 Socket 概述	329		

语言基础篇

第 1 章 Java 语言概述

学习目标

本章主要介绍 Java 语言的基本概况，主要包括 Java 语言的发展历程、Java 语言的特点、Java 的版本、JDK 的组成、Java 开发环境和第一个 Java 程序等。通过本章的学习，读者应能了解 Java 语言的基本特性、构建 Java 桌面程序环境和编写第一个 Java 程序。本章的学习要点包括：

- Java 语言的特点。
- JDK 的版本与选择。
- Java 桌面开发环境的构建。
- Java 程序编写、编译和运行的过程。
- Eclipse 和 NetBeans 开发工具的使用。

教学导航

Java 语言是由 Sun 公司于 1995 年 5 月 23 日正式推出的纯面向对象的程序设计语言，集安全性、简单性、易用性和平台无关性于一身，特别适合在网络环境中使用。本章主要内容及其在 Java 桌面开发技术中的位置如图 1-1 所示。

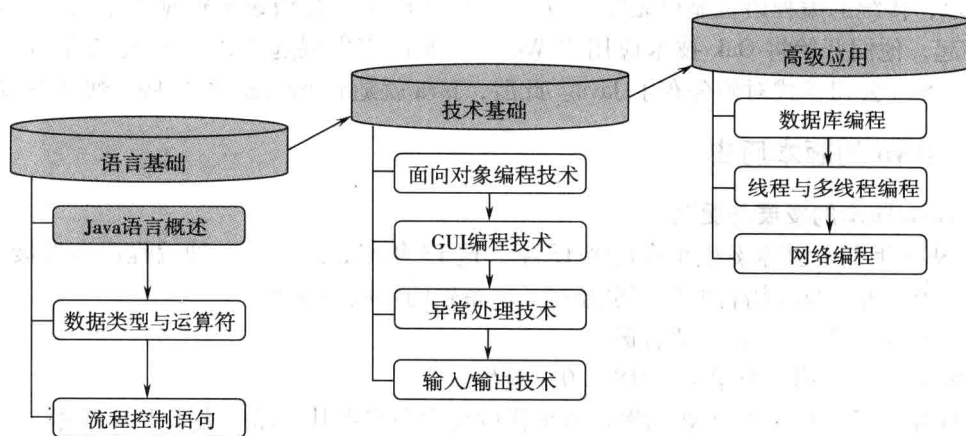


图 1-1 第 1 章学习导航

1.1 Java 语言简介

Java 是 Sun 公司推出的新一代面向对象程序设计语言，它的面向对象、跨平台和分布应用等特点给编程人员带来一种崭新的计算机概念，使 WWW 由最初的单纯提供静态信息发展到现在的提供各种各样的动态服务。Java 不仅能够编写嵌入网页中具有声音和动画功能的小应用程序，而且还能够编写大中型的应用程序，其强大的网络功能可以把整个 Internet 作为一个统一的运行平台，极大地拓展了传统单机计算模式和客户/服务器模式应用程序的外延和内涵。从 1995 年正式问世以来，Java 逐步从一种单纯的高级编程语言发展为一种重要的 Internet 开发平台，并进而引发和带动了 Java 产业的发展与壮大，成为当今计算机业界不可忽视的力量和最主要的发展潮流。

1.1.1 Java 语言的发展历程

1991 年，美国 Sun Microsystems 公司的某个研究小组为了能够在消费电子产品上开发应用程序，积极寻找合适的编程语言。消费电子产品种类繁多，包括 PDA、机顶盒、手机等，即使是同一类消费电子产品所采用的处理芯片和操作系统也不相同，存在着跨平台的问题。当时最流行的编程语言是 C 和 C++ 语言，Sun 公司的研究人员就考虑是否可以采用 C++ 语言来编写消费电子产品的应用程序，但是研究表明，对于消费电子产品而言 C++ 语言过于复杂和庞大，并不适用，安全性也并不令人满意。于是，由 Bill Joy 领导的研究小组就开始着手设计和开发出一种语言，将 C++ 语言进行简化，去掉其中的指针操作、运算符重载、多重继承等，得到了 Java 语言，并将它变为一种解释执行的语言，在每个芯片上装上一个 Java 语言虚拟机。刚开始，Java 语言被称之为 Oak 语言（橡树语言）。

Java 语言的发展得益于 WWW 的发展。在 Java 出现以前，Internet 上的信息内容都是一些乏味死板的 HTML 文档，这对于那些迷恋于 Web 浏览的人们来说简直不可容忍。他们迫切希望能在 Web 中看到一些交互式的内容，开发人员也极希望能够在 Web 上创建一类无需考虑软硬件平台就可以执行的应用程序，当然这些程序还要有极大的安全保障。对于用户的这种要求，传统的编程语言显得无能为力。Sun 公司的工程师敏锐地察觉到了这一需求，从 1994 年起，他们开始将 Oak 技术应用于 Web。Oak 语言发展起来以后来改名为 Java 语言。1995 年，Sun 公司正式对外公布了 Java。此后，Java 就随着 Internet 的发展而快速发展起来。

1.1.2 Java 的版本历史

1. Java 版本的发展与更新

从 1995 年 Java 技术诞生至今已经 13 年。这 13 年来，Java 从一种编程语言发展为一个平台、一个产业，Java 语言的版本也经过了一系列的发展与更新。

1995 年 5 月 23 日，Java 语言诞生。

1996 年 1 月，第一个 JDK（JDK1.0）诞生。

1996 年 4 月，10 个最主要的操作系统供应商申明将在其产品中嵌入 Java 技术。

1996 年 9 月，约 8.3 万个网页应用了 Java 技术来制作。

1997 年 2 月 18 日，JDK1.1 发布。

1997年4月2日, JavaOne 会议召开, 参与者逾 10,000 人, 创当时全球同类会议规模的纪录。

1997年9月, JavaDeveloperConnection 社区成员超过 10 万人。

1998年2月, JDK1.1 下载超过 2,000,000 次。

1998年12月8日, Java2 企业平台 J2EE 发布。

1999年6月, Sun 公司发布 Java 的 3 个版本: 标准版 (J2SE)、企业版 (J2EE) 和微型版 (J2ME)。

2000年5月8日, JDK1.3 发布。

2000年5月29日, JDK1.4 发布。

2001年6月5日, NOKIA 宣布, 到 2003 年将出售 1 亿部支持 Java 的手机。

2001年9月24日, J2EE1.3 发布。

2002年2月26日, J2SE1.4 发布, 自此 Java 的计算能力有了大幅提升。

2004年9月30日, J2SE1.5 发布, 成为 Java 语言发展史上的又一里程碑。为了表示该版本的重要性, J2SE1.5 更名为 Java SE 5.0。

2005年6月, JavaOne 大会召开, Sun 公司公开 Java SE 6。此时, Java 的各种版本已经更名, 取消了原有版本中的数字“2”: J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

2006年12月, Sun 公司发布 JRE6.0。

2. 几个重要的名词

(1) Java SE (Java Standard Edition) Java SE 是 Java 各应用平台的基础, 主要用于桌面开发和低端商务应用开发。Java SE 也正是本书主要介绍的内容, 可分为 4 个主要部分, 即 Java 虚拟机 (JVM)、Java 运行环境 (JRE)、开发工具及其 API、Java 语言等。

Java 虚拟机 (JVM) 向 Java 程序提供运行环境。JVM 包括在 Java 运行环境 JRE 中, 所以要运行 Java 程序, 必须先取得 JRE 并安装。Java 开发工具包 JDK 除包含 JRE 的所有内容外, 还提供了 Java 运用程序的开发工具, 如 Javac、Java、Appletviewer 等。因此, 开发 Java 运用程序, 必须先安装 JDK。除此之外, Java 最重要的作用就是提供了庞大且功能强大的 API 类库, 如字符串处理、数据输入/输出、网络组件、使用者窗口接口等功能。可以使用这些 API 作为基础来进行程序开发, 而无须重复开发功能相同的组件。事实上, 在 Java 的学习过程中, 更多的是要学习 Java 提供了哪些 API, 及如何使用 API 构造自己的程序。

(2) Java EE (Java Enterprise Edition) Java EE 以 Java SE 为基础, 主要用于企业级应用开发, 提供面向分布式、多层式、组件式和 Web 应用程序的开发, 定义了一系列的服务、API、协议。整个 Java EE 的体系是相当庞大的, 比较为人所熟悉的技术有 JSP、Servlet、Enterprise JavaBeans (EJB)、Java Remote Method Invocation (RMI) 等。

(3) Java ME (Java Micro Edition) Java ME 是面向小型数字设备 (如手机、PDA、股票机等) 的开发及部署应用程序平台。目前, 已有越来越多的手持设备支持 Java ME 所开发出来的程序, 如 Java 游戏、股票相关程序、记事程序、月历程序等。

随着 Java 技术的不断进步, Java 已由一个程序设计语言变成了一种开发软件的平台, 成为一种开发软件的标准与架构的统称。事实上, 语言在整个 Java 的蓝图中只是极小的一部分, 学习 Java 本身也不仅仅在于学习如何使用 Java 语言的语法, 更多的是要学习如何应

用 Java 所提供的资源与各种标准，从而开发出架构更好、更容易维护的软件。

1.1.3 Java 语言与 C/C++

Java 的前身 Oak 语言是在 C++ 的基础上开发的，而 C++ 是在 C 语言的基础上开发的。因此，Java 和 C/C++ 语言具有许多相似之处。但是，如果仔细研究 Java 语言的许多细节，就会发现 Java 取消了不少 C 及 C++ 的特性，并且加入了一些新的特性。这些差异包括以下内容。

1. 不再有指针 (Pointer) 的概念

这是 Java 和 C/C++ 语言在语法上的主要区别之一。在 C/C++ 语言中，指针的灵活运用将会给程序设计带来极大的便利，但是其灵活性也成为了导致程序不稳定的一个主要因素。配合 C/C++ 语言的内存管理策略，编程人员必须亲自跟踪自己向系统申请到的内存，最后确认交还给系统。并且在使用指针时，要随时注意是否超过合法的内存空间，以免造成 Segmentation Fault (段错误) 或 General Protection Fault (保护错误) 这样的问题。

Java 提供了一种引用 (Reference) 类型用来替代指针，通过引用去访问申请到的内存空间，可以确保不会访问不属于自己的内存空间。同时，程序的执行系统也可以动态地进行内存垃圾回收工作，将那些被取消引用的内存空间回收给系统使用。这种动态内存分配机制，通过牺牲一定的灵活性，保证了内存分配在一定程度上的安全性。

2. 不再有函数 (Function) 的概念

在 Java 编程语言中，取消了结构化语言中最重要的部分——函数。在面向对象程序设计的概念中，对象的数据才是真正的主题，而处理对象数据的方法则必须依附在对象内才有意义。因此，函数概念完全是不必要的，这强化了面向对象的开发策略。

3. 不再使用 structure、union 和 typedef

事实上，在 C++ 语言中就可以去掉 C 语言中的 structure 和 union 等对复杂数据的自定义结构类型了，因为类 (Class) 的定义方式完全可以实现这项功能。而 typedef 的功能也可以由类来实现。虽然 C++ 语言这样设计是为了和 C 语言兼容，但这些毕竟是多余的语言特点，因此在 Java 语言中便被抛弃了。

4. 不再有类的多重继承 (Multiple Inheritance)

在 C++ 语言中，多重继承是一项很强大的功能，但也是较难掌握的。取消多重继承降低了 Java 语言的功能，但它使 Java 看起来更为简洁。同时，Java 也提供了接口 (Interface) 的方式，可以实现部分多重继承的作用。和多重继承的不同之处在于接口并不会定义类方法的内容以及类中的数据。

5. 不再有操作符重载 (Operator Overloading)

操作符重载被看作是 C++ 语言的一个特色。在 C++ 语言中，利用操作符重载，程序设计者通过对现有运算符赋予自己的定义，可以使程序看起来更为自然。但是，如果使用不当，也会使整个程序的可读性大受影响。同时，这一功能的存在也并非必要，程序设计者可以通过定义类中的方法来达到同样的目的。

6. 取消了自动类型转换

Java 是一种强类型检查的编程语言，对于诸如浮点型变量赋值给整型变量这样的情况而言，在 C++ 语言的语法中是允许的，最多只是在编译时给出警告信息。但是，在 Java 中，除非写明强制类型转换，否则编译无法通过。

7. 不再有预处理器 (Preprocessor) 功能

利用 C/C++ 语言中提供的 `#define`、`#include` 等预处理指令, 熟练的程序设计人员可以自己开发一套只有自己才看得懂的宏指令集。从软件工程的角度来看, 这对团队开发软件和整个软件的维护都是十分不利的。同时, 预编译后的程序代码和程序设计者看到的程序代码是不同的。如果宏指令集有错, 编译器产生的错误信息将不会是编程人员所预期的, 这增加了程序调试的难度。

8. 取消了 goto 语句

在 C++ 语言中, `goto` 语句已经不被建议使用, 之所以仍然保留 `goto`, 只是为了和 C 语法兼容。在 Java 中, `goto` 语句被完全抛弃了, 与此同时, Java 又扩大了 `break` 语句和 `continue` 语句的功能。通过使用 `break` 和 `continue` 语句, 程序流程被允许在多层循环中跳转。

虽然 Java 是在 C++ 语言的基础上开发的, 但并不是 C++ 语言的增强版, 不是用来取代 C++ 语言。Java 语言与 C++ 语言既不向上兼容, 也不向下兼容, 两者将长时间共存。Java 在理论和实践上都与 C++ 语言有许多不同点。Java 是独立于平台、面向 Internet 的分布式编程语言, 它对 Internet 编程的影响就如同 C/C++ 语言对系统编程的影响一样。Java 的出现改变了编程方式, 但 Java 并不是孤立存在的一种语言, 而是计算机语言多年来演变的结果。

1.1.4 Java 的特点

1. 简单性

Java 语言是一种面向对象的语言, 它通过提供最基本的方法来完成指定的任务, 只需理解一些基本的概念, 就可以用它编写出适合于各种情况的应用程序。Java 略去了运算符重载、多重继承等模糊的概念, 并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。另外, Java 也适合于在小型设备上运行, 它的基本解释器及类的支持只有 40KB 左右, 加上标准类库和线程的支持也只有 215KB 左右。

2. 面向对象

Java 语言的设计集中于对象及其接口, 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以及相应的方法, 实现了模块化和信息隐藏 (参阅第 4 章); 而类则提供了一类对象的原型, 并且通过继承机制, 子类可以使用父类所提供的方法, 实现了代码的复用。

3. 分布性

Java 是面向网络的语言, 通过它提供的类库可以处理 TCP/IP, 用户可以通过 URL 地址在网络上很方便地访问其他对象 (参阅第 10 章)。

4. 健壮性

Java 在编译和运行程序时, 都要对可能出现的问题进行检查, 以消除错误的产生。它提供自动垃圾收集功能来进行内存管理, 防止出现程序员在管理内存时容易产生的错误。通过集成的面向对象的异常处理机制 (参阅第 6 章), 帮助程序员正确处理编译时可能出现的异常, 以防止系统的崩溃。另外, Java 在编译时还可捕获类型声明中的许多常见错误, 防止动态运行时不匹配问题的出现。

5. 安全性

用于网络、分布环境中的 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, 这样就防止程序员使用“特洛伊”木马等欺骗

手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

6. 体系结构中立

Java 解释器生成与体系结构无关的字节码指令，只要安装了 Java 运行时环境（JRE），Java 程序就可在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示，Java 解释器得到字节码后，对它进行转换，使之能够在不同的平台运行。

7. 可移植性

与平台无关的特性使 Java 程序可以方便地被移植到网络中的不同计算机上。同时，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 运行时系统由标准 C 语言实现，这使得 Java 系统本身也具有可移植性。

8. 解释执行

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时的信息，使得连接过程更加简单。

9. 高性能

和其他解释执行的语言（如 Basic）不同，Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码，从而得到较高的性能。

10. 多线程

多线程机制使应用程序能够并行执行，而且同步机制保证了对共享数据的正确操作。通过使用多线程，程序设计者可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制，这样就能很容易地实现网络中的实时交互行为（参阅第 9 章）。

11. 动态性

Java 的设计使它适合于一个不断发展的环境，它允许程序动态地装入运行过程中所需要的类。在类中可以自由地加入新的方法和实例变量而不会影响用户程序的执行。并且 Java 通过接口来支持多重继承，使之比严格的类继承具有更灵活的方式和扩展性。

1.2 JDK 和 Java 开发环境及工作原理

1.2.1 JDK

JDK（Java 开发工具）是许多 Java 初学者使用的开发环境，由一个标准类库和一组测试及建立文档的 Java 实用程序组成，见表 1-1。

表 1-1 Java 实用程序

程序名称	程序功能
Javac	Java 编译器，将 Java 源程序转换成字节码
Java	Java 解释器，将 Java 字节码文件（类文件）解释为二进制代码执行
Appletviewer	小程序浏览器，一种执行 HTML 文件中的 Java 小程序的 Java 浏览器
Javadoc	根据 Java 源程序及说明语句生成 HTML 格式的标准的帮助文档
Jdb	Java 调试器，可以逐行执行程序，设置断点和检查变量
Javah	产生可以调用 Java 过程的 C 过程，或建立能被 Java 程序调用的 C 过程的头文件
Javap	Java 反汇编器，显示编译类文件中的可访问功能和数据，同时显示字节码含义

1.2.2 Java 开发环境

1. 下载 JDK

JDK 可以到 Sun 公司网站中提供的下载地址进行免费下载, 本书使用的 Java SE 6 的下载地址是 <http://java.sun.com/javase/downloads/ea.jsp>。

2. 安装 JDK

下载完毕后, 直接运行所下载的 Java jdk1.6.0.exe, 按照提示进行安装。在安装过程中, 可以更改默认的安装路径 (如 D:\Java\jdk1.6)。

3. 设置 Path 与 Classpath

安装完成后需要配置 JDK 环境变量, 从而可以对编辑的 Java 程序进行编译。操作步骤如下:

1) 在 Windows 2000/XP/2003 的桌面上右键单击“我的电脑”, 在弹出的快捷菜单中选择“属性”→“高级”→“环境变量”, 如图 1-2 所示。

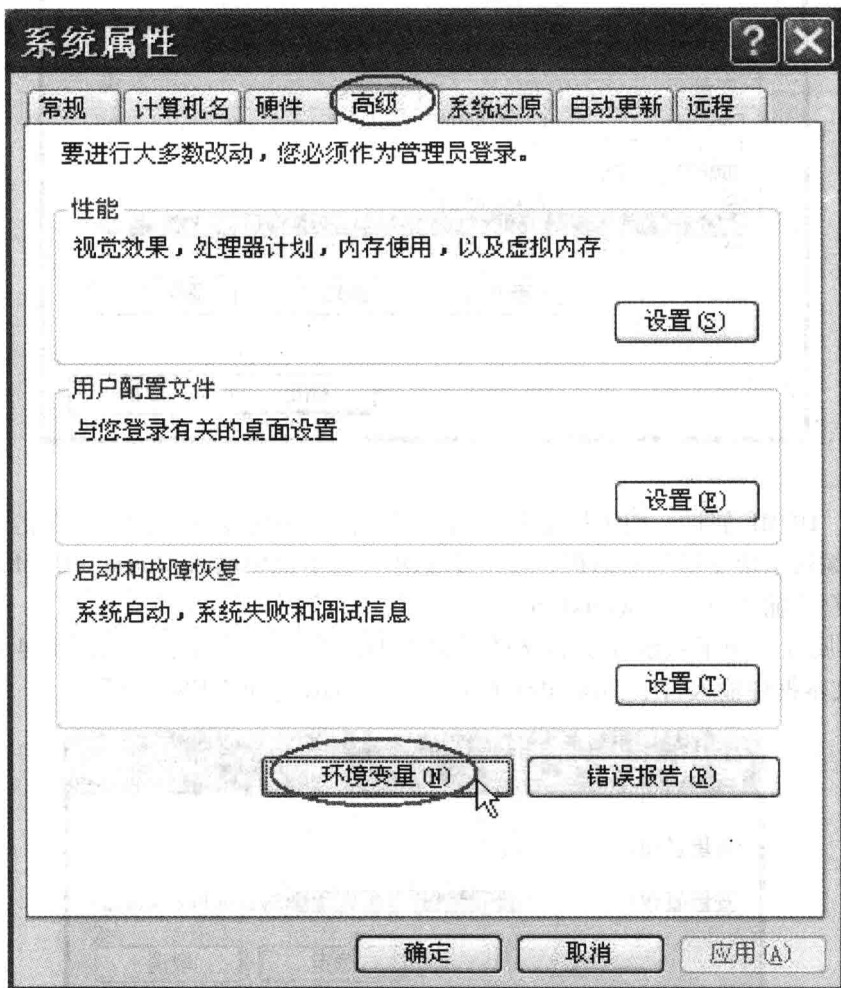


图 1-2 “系统属性”对话框