



普通高等教育“十一五”国家级规划教材

中国高等学校信息管理与信息系统专业规划教材

Java程序设计基础 (第4版)

陈国君 主编

陈磊 邹林达 李梅生 等 编著



根据教育部管理科学与工程类学科专业教学指导委员会主持鉴定的《中国高等院校信息系统学科课程体系》组织编写



与美国ACM和IEEE/CS Computing Curricula 2005同步



清华大学出版社

014007124

TP312JA-43
195-4



普通高等教育“十一五”国家级规划教材

内容简介

中国高等学校信息管理与信息系统专业规划教材

Java程序设计基础 (第4版)

陈国君 主编

陈磊 邹林达 李梅生 等 编著



清华大学出版社
北京

TP 312JA-43

195-4

内 容 简 介

Java 是近年来最流行的计算机程序设计语言。本书全面系统地介绍 Java 语言的特点及应用技术,内容上以 Java 的基础程序设计、面向对象程序设计和事件处理为三大主线,利用浅显易懂的语言、丰富简单的实例,完整地介绍了 Java 面向对象程序设计的要点和难点。本书共分 17 章,其中第 1~5 章介绍程序设计基础;第 6~11 章介绍面向对象程序设计;第 12 章和第 13 章介绍界面设计和事件处理;第 14 章介绍绘图程序设计;第 15 章介绍 Applet 程序设计;第 16 章介绍 Java 数据库编程;第 17 章介绍 Java 网络编程。

本教材在取材上特别注意教材的体系,其特色是结构合理、概念清楚、思路清晰、循序渐进、通俗易懂。在结构上特别注重前后内容的连贯性,力求抓住关键、突出重点、分解难点,体现“理论性、实用性、技术性”三者相结合的编写特色。对每个知识点不但能告诉读者要怎么做,而且还要告诉读者这样做的原因和道理。

本书可以作为高等院校计算机、信息管理与信息系统及其相关专业的教学用书,也可作为各学校程序设计公共选修课的教材,同时也可作为职业教育的培训用书和 Java 初学者的入门教材,或为具有一定 Java 编程经验的开发人员学习使用。

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 程序设计基础 / 陈国君主编. -- 4 版. -- 北京:清华大学出版社, 2013

中国高等学校信息管理与信息系统专业规划教材

ISBN 978-7-302-33142-1

I. ①J… II. ①陈… III. ①JAVA 语言-程序设计-高等学校-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 159634 号

责任编辑:索梅李晔

封面设计:常雪影

责任校对:李建庄

责任印制:何芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 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

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市李旗庄少明印装厂

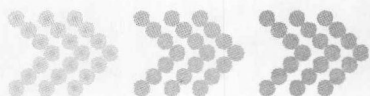
经 销:全国新华书店

开 本:185mm×260mm 印 张:26.5 字 数:657 千字

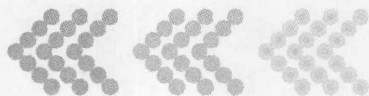
版 次:2006 年 1 月第 1 版 2013 年 11 月第 4 版 印 次:2013 年 11 月第 1 次印刷

印 数:1~2500

定 价:39.50 元



前言



第 4 版

该书自第 1 版面市以来,在广大教师和学生的使用过程中,一直受到教师和学生的好评,市场反映非常热烈。尤其是该教材的第 3 版被“中国书刊发行业协会”评为 2011 年度全行业优秀畅销教材后,虽已多次印刷,但均已售罄。为了能适应科学技术的发展和计算机教学的需要,清华大出版社和本书作者在征求广大读者意见和建议的基础上,决定修订再版,以便更好地满足广大读者的需求。该版在总结了以前版本的经验之上,并根据读者的建议,增加了数据库编程内容,使得该版教材在体系结构、内容组织、语言表达等方面都更加完善。正是由于该教材优化的知识体系、通俗易懂的讲解方式以及对知识点的透彻分析和灵活实用的举例而深受读者的欢迎,这也是促使该书再版的主要原因。由于计算机技术发展得很快,加之作者水平有限,书中难免有不足之处,欢迎广大读者批评指正。

书中所有例题全部在 JDK 7 环境下编译通过并运行。

在此本书全体作者对清华大学出版社的大力支持,尤其是对索梅编审的热心关注、建议与指导表示衷心的感谢!

作 者

2013 年 5 月



前言



第 3 版

本教材在广大教师和学生的使用过程中,由于教师和学生建议,这一版总结了以前版本的经验,同时又收集了广大读者的意见和建议,使得该版在内容组织、表达方式等方面都较以前的版本有较大的改进。该版中的所有例题完全采用 Swing GUI 组件重新编写,每个例题都突出一个编程的知识点。除保持了由浅入深、循序渐进的优点外,还对教学过程中学生和教师遇到的问题进行了详细讲解,彰显了突出重点、分解难点的特色,使学生对学习 Java 编程感兴趣,而兴趣又成了学习 Java 语言的动力,使学生在学习的乐趣中掌握了 Java 的基本编程技巧。这种良性循环都归功于教材对内容的精选和组织结构的合理性,衷心希望本教材能成为广大读者的良师益友。由于 Java 技术的内容庞大、结构复杂,所以从其中抽出基本的内容,并能以通俗的方式介绍给读者并非易事,所以本教材难免存在不尽如人意的地方,因此希望广大读者继续能对本教材提出合理化建议,使本教材更加完善。

本教材由陈国君、陈磊、刘洋、邹林达、李梅生、鲜征征、陆寄远共同修改完成。

该教材的再版,得到了清华大学出版社的大力支持,尤其是得到了索梅编审的热心关注与指导,作者在此表示衷心感谢。

作 者

2011 年 1 月



前言

第 2 版

自该书第一版面市以来，市场反映非常热烈。正是由于该书的循序渐进、由浅入深、突出重点、分解难点的讲解方式，深受读者的欢迎。许多读者在与作者沟通时，对本书给予了充分的肯定，并提出了一些合理化建议和意见。在广大读者的要求和关怀下，直接推动了本书的再版。为使本书的结构更为合理、重点进一步突出、难点进一步分解，本书第 2 版在各方面较之第 1 版做了相应的修改，并用最新的 JDK 6 版本修改和重写了部分章、节的内容，使之更加通俗易懂，同时更换了部分例题，使之更加具有代表性，书中所有例题全部在 JDK 6 环境下编译通过。同时在每章的后边还增加了对该章内容的小结，以此来帮助读者归纳和总结所学内容，以最大可能地满足读者要求。

本教材由陈国君、陈磊、陈锡祯、刘洋、邹林达共同完成修改。

本书的再版得到了清华大学出版社的大力支持，在此表示衷心的感谢！

作 者

2008 年 10 月



前言



第 1 版

随着计算机技术的日新月异及网络化发展的趋势, Java 语言已成为目前最具吸引力且功能强大的程序设计语言。Java 是完全面向对象的, 且具有容易学习、功能强大、程序的可读性好等优点, 是其他传统语言无可比拟的。

由于教材是体现教学内容和教学方法的知识载体, 是进行教学的基本工具, 也是深化教育教学改革, 全面推进素质教育、培养创新人才的重要保证, 因此本教材在内容的编排上做了精心的设置与选取, 注重基础知识的理解与基本技能的培养。全书内容思路清晰、结构严谨, 在内容的叙述上由浅入深、循序渐进、用语规范, 全面准确讲述基本语法和面向对象技术等理论内容, 完整地介绍了 Java 2 面向对象程序设计的要点和难点。在结构上特别注重前后内容的连贯性, 做到了抓住关键、突出重点、分解难点, 体现“理论性、实用性、技术性”三者相结合的编写特色。同时, 将实用性强的应用程序穿插在理论叙述中, 以实例体现和巩固理论基础知识, 并结合新技术的发展趋势, 介绍网络通信机制等。这些实例汇集了作者多年从事计算机教学和软件开发过程中的案例精品, 并全部在 JDK 1.4 环境下编译通过。

本书可以作为高等院校计算机及其相关专业的教学用书, 也可作为各学校程序设计公共选修课的教材, 同时还可用作职业教育的培训用书和 Java 2 初学者的入门教材或为具有一定 Java 语言编程经验的开发人员学习使用。

全书分为 16 章: 第 1 章 Java 语言概述、第 2 章 Java 语言开发环境、第 3 章 Java 语言基础、第 4 章 流程控制、第 5 章 数组与字符串、第 6 章 类与对象、第 7 章 Java 语言类的特性、第 8 章 继承、抽象类和接口、第 9 章 异常处理、第 10 章 Java 的输入输出与文件处理、第 11 章 多线程、第 12 章 图形界面设计、第 13 章 事件处理、第 14 章 绘图程序设计、第 15 章 小程序设计、第 16 章 Java 网络编程。

本教材由陈国君教授主编, 其中, 第 1~4 章与刘洋合编; 第 5~10 章与陈锡祯合编; 第 11~16 章与陈磊合编。本书中的所有程序都由陈磊调试通过。由于作者水平有限, 书中若有不妥之处, 恳请广大读者批评指正。

作 者

2005 年 8 月

目录

第 1 章 Java 语言概述	1	3.7.2 关系运算符	39
1.1 Java 语言的诞生与发展	1	3.7.3 逻辑运算符	40
1.2 Java 语言的特点	2	3.7.4 位运算符	41
1.3 Java 技术简介	4	3.7.5 赋值运算符	41
1.4 Java 虚拟机	5	3.7.6 条件运算符	42
1.5 Java 程序种类和结构	6	3.7.7 字符串运算符	43
本章小结	8	3.7.8 表达式及运算符的优先级、结合性	43
习题 1	9	本章小结	44
第 2 章 Java 语言开发环境	10	习题 3	45
2.1 Java 开发工具	10	第 4 章 流程控制	46
2.1.1 JDK 的下载与安装	11	4.1 语句与复合语句	46
2.1.2 设置 JDK 的操作环境	13	4.2 顺序结构	47
2.2 JDK 帮助文档下载与安装	15	4.3 分支结构	47
2.2.1 下载 JDK 帮助文档	15	4.3.1 if 条件语句	48
2.2.2 安装 JDK 帮助文档	16	4.3.2 switch 选择语句	50
2.3 JDK 的使用	17	4.4 循环结构	52
2.3.1 编译与运行 Java 应用程序	17	4.4.1 while 语句	53
2.3.2 编译与运行 Java 小程序	19	4.4.2 do-while 语句	55
本章小结	21	4.4.3 for 循环语句	58
习题 2	22	4.4.4 多重循环	59
第 3 章 Java 语言基础	23	4.5 循环中的跳转语句	60
3.1 数据类型	23	4.5.1 break 语句	60
3.2 关键字与标识符	26	4.5.2 continue 语句	61
3.3 常量	27	4.5.3 return 语句	61
3.4 变量	29	本章小结	61
3.5 数据类型转换	30	习题 4	62
3.6 从键盘输入数据	33	第 5 章 数组与字符串	63
3.7 运算符与表达式	38	5.1 数组的基本概念	63
3.7.1 算术运算符	38	5.2 一维数组	64
		5.2.1 一维数组的定义	64
		5.2.2 一维数组元素的访问	66



5.2.3 一维数组的初始化及应用	67	7.3.1 构造方法的作用与定义	100
5.3 foreach 语句与数组	70	7.3.2 默认的构造方法	102
5.4 多维数组	70	7.3.3 构造方法的重载	102
5.4.1 二维数组	71	7.3.4 从一个构造方法调用另一个构造方法	103
5.4.2 三维以上的多维数组	74	7.3.5 公共构造方法与私有构造方法	105
5.5 字符串	75	7.4 静态成员	106
5.5.1 字符串变量的创建	75	7.4.1 实例成员	106
5.5.2 String 类的常用方法	76	7.4.2 静态变量	107
本章小结	78	7.4.3 静态方法	109
习题 5	79	7.4.4 静态初始化器	110
第 6 章 类与对象	80	7.5 对象的应用	111
6.1 类的基本概念	80	7.5.1 对象的赋值与比较	111
6.2 定义类	81	7.5.2 引用变量作为方法的返回值	114
6.3 对象的创建与使用	84	7.5.3 类类型的数组	114
6.3.1 创建对象	84	7.5.4 以对象数组为参数进行方法调用	115
6.3.2 对象的使用	85	本章小结	116
6.3.3 在类定义内调用方法	88	习题 7	117
6.4 参数的传递	90	第 8 章 继承、抽象类和接口	118
6.4.1 以变量为参数调用方法	90	8.1 类的继承	118
6.4.2 以数组作为参数或返回值的方法调用	92	8.1.1 子类的创建	119
6.5 匿名对象	94	8.1.2 在子类中访问父类的成员	123
本章小结	94	8.1.3 覆盖	124
习题 6	95	8.1.4 不可被继承的成员与最终类	127
第 7 章 Java 语言类的特性	96	8.1.5 Object 类	128
7.1 类的私有成员与公共成员	96	8.2 抽象类	132
7.1.1 私有成员	96	8.2.1 抽象类与抽象方法	132
7.1.2 公共成员	97	8.2.2 抽象类的应用	133
7.1.3 缺省访问控制符	98	8.3 接口	134
7.2 方法的重载	99	8.3.1 接口的定义	135
7.3 构造方法	100	8.3.2 接口的实现与引用	135
		8.3.3 接口的继承	137
		8.3.4 利用接口实现类的多重继承	138
		8.4 内部类与匿名类	139



8.4.1 内部类	139	10.2.1 基本的输入输出流	175
8.4.2 匿名内部类	141	10.2.2 输入输出流的应用	176
8.5 包	143	10.3 使用 Reader 和 Writer 流类	185
8.5.1 包的概念	143	10.3.1 使用 FileReader 类读取文件	186
8.5.2 使用 package 语句创建包	143	10.3.2 使用 FileWriter 类写入文件	187
8.5.3 Java 语言中的常用包	144	10.3.3 使用 BufferedReader 类读取文件	188
8.5.4 Java 语言中几个常用的类	146	10.3.4 使用 BufferedWriter 类写入文件	189
8.5.5 利用 import 语句引用 Java 定义的包	149	10.4 文件的处理与随机访问	191
8.5.6 Java 程序结构	150	10.4.1 Java 语言对文件与文件夹的管理	191
8.6 Java 语言的垃圾回收	150	10.4.2 对文件的随机访问	193
本章小结	151	本章小结	196
习题 8	153	习题 10	197
第 9 章 异常处理	154	第 11 章 多线程	198
9.1 异常处理的基本概念	154	11.1 线程的概念	198
9.1.1 错误与异常	154	11.1.1 程序、进程、多任务与线程	199
9.1.2 Java 语言的异常处理机制	155	11.1.2 线程的状态与生命周期	200
9.2 异常处理类	156	11.1.3 线程的调度与优先级	202
9.3 异常的处理	158	11.2 Java 的 Thread 线程类与 Runnable 接口	203
9.4 抛出异常	162	11.2.1 利用 Thread 类的子类来创建线程	203
9.5 自定义异常类	168	11.2.2 用 Runnable 接口来创建线程	206
本章小结	170	11.2.3 线程间的数据共享	208
习题 9	170	11.3 多线程的同步控制	211
第 10 章 Java 语言的输入输出与文件处理	171	11.4 线程之间的通信	216
10.1 Java 语言的输入输出类库	171	本章小结	218
10.1.1 流的概念	171	习题 11	220
10.1.2 输入输出流类库	173	第 12 章 图形界面设计	221
10.2 使用 InputStream 和 OutputStream 流类	174	12.1 图形用户界面概述	221
		12.2 图形用户界面工具包——Swing	222
		12.2.1 Swing 组件分类	222
		12.2.2 颜色类 Color、字体类 Font 与	



图像图标类 ImageIcon	231	13.9 按键事件类	283
12.3 创建组件	233	13.10 鼠标事件类	286
12.3.1 标签 JLabel	233	13.11 列表框及相应的事件处理	289
12.3.2 命令按钮 JButton、复选框 JCheckBox 和单选按钮 JRadioButton	236	13.12 组合框及相应的事件处理	291
12.3.3 文本编辑组件 JTextField、 JPasswordField、JTextArea 与 滚动窗格 JScrollPane	239	13.13 菜单设计	294
12.3.4 选项卡窗格 JTabbedPane	243	13.13.1 窗口菜单	295
12.4 布局管理器	245	13.13.2 弹出式菜单	301
12.4.1 流式布局管理器 FlowLayout	246	13.14 工具栏设计	304
12.4.2 边界式布局管理器 BorderLayout	248	13.15 滑动条设计及相应的事件处理	306
12.4.3 网格式布局管理器 GridLayout	249	13.16 文件选择对话框	308
12.4.4 卡片式布局管理器 CardLayout	251	13.17 颜色选择窗格	313
12.4.5 网格包布局管理器 GridBagLayout	253	13.18 定时器	315
12.4.6 盒式布局管理器 BoxLayout	256	本章小结	317
12.4.7 重叠布局管理器 OverlayLayout 和弹簧布局管理器 SpringLayout 简介	258	习题 13	318
本章小结	258	第 14 章 绘图程序设计	319
习题 12	259	14.1 图形坐标系与绘图类	319
第 13 章 事件处理	260	14.2 绘图程序设计	322
13.1 Java 语言的事件处理机制 ——委托事件模型	260	本章小结	328
13.2 Java 语言的事件类	266	习题 14	329
13.3 适配器类	271	第 15 章 小程序设计	330
13.4 命令按钮及相应的事件处理	271	15.1 小程序的基本工作原理	330
13.5 复选框、单选按钮及相应的事件 处理	273	15.2 JApplet 类	331
13.6 文本组件及相应的事件处理	275	15.3 Java 小程序编程实例	333
13.7 窗口组件及窗口事件处理	277	15.4 将应用程序转换成小程序及 小程序的安全性	337
13.8 对话框设计及相应的事件处理	279	15.5 图像文件处理	338
		15.6 播放音乐	339
		15.7 动画程序设计	341
		本章小结	347



习题 15	347	第 17 章 Java 网络编程	387
第 16 章 Java 数据库程序设计	349	17.1 网络基础	387
16.1 关系数据库系统	349	17.1.1 TCP/IP 协议	387
16.1.1 数据库与数据库表	350	17.1.2 通信端口	388
16.1.2 完整性约束	351	17.1.3 URL 概念	388
16.2 SQL	352	17.1.4 Java 语言的网路编程	389
16.2.1 创建数据库	352	17.2 URL 编程	390
16.2.2 表操作	353	17.2.1 创建 URL 对象	390
16.2.3 表数据操作	354	17.2.2 使用 URL 类访问网络资源	391
16.2.4 数据查询	355	17.3 用 Java 语言实现底层网络通信	392
16.3 JDBC	358	17.3.1 InetAddress 程序设计	392
16.3.1 JDBC 概述	358	17.3.2 基于连接的 Socket 通信程序设计	394
16.3.2 JDBC 类型	359	17.3.3 无连接的数据报通信程序设计	402
16.3.3 使用 JDBC 开发数据库应用程序	360	本章小结	406
16.3.4 数据库的进一步操作	368	习题 17	407
16.3.5 获取元数据	376	参考文献	408
16.3.6 事务操作	380		
16.3.7 通过 Java JApplet 访问数据库	383		
本章小结	385		
习题 16	386		



第 1 章 Java 语言概述

本章主要内容:

- Java 语言的特点。
- Java 源文件 (.java) 与 Java 字节码文件 (.class)。
- Java 应用程序和 Java 小程序的主类。
- Java 虚拟机。
- Java 程序的种类和结构。
- Java 应用程序和 Java 小程序的差异。

Java 语言是一种简单易用、完全面向对象、与平台无关、安全可靠、主要面向 Internet 的开发工具。

1.1 Java 语言的诞生与发展

Java 语言诞生于 20 世纪 90 年代初期,自问世以来,它的快速发展已经让整个 Web 世界发生了翻天覆地的变化。

Java 语言的前身是 Sun Microsystems 公司 (Sun 公司于 2009 年 4 月被 Oracle 公司收购) 开发的一种用于智能化家电的名为 Oak (橡树) 的语言,它的基础是当时最为流行的 C 和 C++ 语言。但是,由于一些非技术上的原因, Oak 语言并没有得到迅速推广。直到 1993 年, WWW (万维网) 迅速发展, Sun 公司发现可以利用 Oak 语言的技术来创造含有动态内容的 WWW 网页,于是受人冷落了 Oak 语言又被重新开发和改造,并将改造后的 Oak 语言改名为 Java 语言 (Java 是太平洋上的一个盛产咖啡的岛屿的名字)。终于,在 1995 年, Java 这个被定位于网络应用的程序设计语言被正式推出。

由于 Java 语言功能强大,其问世后不久,即被业界广泛接受,于是 IBM、Apple、DEC、Adobe、HP、Oracle、Toshiba、Netscape 和 Microsoft 等大公司均购买了 Java 语言的许可证。Microsoft 还从其 Web 浏览器 Internet Explorer 3.0 版起开始增加了对 Java 语言的支持。同时,众多的软件开发商也开发了许多支持 Java 的产品。在目前以网络为中心的计算机时代,不支持 HTML 和 Java 语言,就意味着应用程序的应用范围只能限于同质的环境。

随着 Java Servlet 的推出, Java 语言极大地推动了电子商务的发展。Java Server Page (JSP)

技术的推出,更是让 Java 语言成为基于 Web 应用程序的首选开发工具。Internet 的普及和迅猛发展,以及 Web 技术的不断渗透,使得 Java 语言在现代社会的经济发展和科学研究中,占据了越来越重要的地位。

1.2 Java 语言的特点

Java 语言是一种跨平台、适合于分布式计算环境的面向对象编程语言。它具有的特点很多,如简单性、面向对象、分布式、解释型、可靠性、安全性、平台无关性、可移植性、高性能、多线程、动态性等。下面介绍 Java 语言的几个重要特性。

1. 简单易学

Java 语言虽然衍生自 C++,与 C++相比 Java 是一个完全面向对象的编程语言。出于安全性和稳定性的考虑,Java 去掉了 C/C++支持的三个不易理解和掌握的数据类型:指针(pointer)、联合体(union)和结构体(struct)。这样做的目的使用户不能通过 Java 程序直接访问内存地址,从而保证了程序更高的安全性。而 C/C++中联合体和结构体的功能,完全可以在 Java 中用类及类的属性等面向对象的方法来实现,这不但更加合理规范,而且还降低了学习难度。

2. 面向对象

Java 语言最吸引人之处就在于它是一种以对象为中心、以消息为驱动的面向对象的编程语言。面向对象的语言都支持三个概念:封装、继承和多态,Java 语言也是如此。

1) 封装

所谓封装就是指利用抽象数据类型将数据和基于数据的操作封装在一起,数据被保护在抽象数据类型的内部,系统的其他部分只有通过封装在数据外面的得到授权的操作,才能够与这个抽象数据类型交互。简单地说,即用一个自主式框架把对象的数据和方法连在一起,形成一个整体。所以说,对象是支持封装的基本单位,是支持封装的手段。Java 语言的封装性很强,它没有全局变量。在 Java 语言中,除了简单的数值类型、字符类型和逻辑类型外,大部分的成员都是对象。而对于这些类型,Java 语言也提供了相应的对象类型以便与其他对象交互操作。

2) 继承

继承是指一个对象直接使用另一个对象的属性和方法。Java 语言给用户提供了一系列的类,并且 Java 语言的类很有层次结构,子类可以继承父类的属性和方法。但是,与其他一些面向对象的编程语言所不同的是,Java 语言只支持单一继承,这样就大大地降低了复杂度,但在 Java 语言中,可以通过接口来实现多重继承。

3) 多态

多态是指一个程序中同名的多个不同方法共存的情况,即一个对外接口,多个内在实现方法。面向对象的程序中多态的情况有多种,可以通过子类对父类方法的覆盖实现多态,也可以利用重载在同一个类中定义多个同名的不同方法来实现多态。多态的特点大大提高了程序的抽象程度和简洁性,同时,它最大限度地降低了类和程序模块之间的耦合性,使得它们不需了解对方的具体细节,就可以很好地共同工作。这个优点,对程序的设计、开发和维护



都有很大的好处。

3. 平台无关性

Java 是与平台无关的语言，这是指使用 Java 语言编写的应用程序不用修改就可在不同的软硬件平台上运行。

平台无关体现在两个方面：源代码级和目标代码级。C 和 C++语言具有一定程度的源代码级平台无关，即用 C 和 C++语言编写的应用程序不用修改只需重新编译就可以在不同平台上运行。Java 语言是靠 Java 虚拟机 (JVM) 在目标代码级实现平台无关性的，可以说，JVM 是 Java 平台无关的基础（关于 JVM 的使用，将在 1.4 节介绍）。

Java 语言的平台无关性意义深远。它使得编程人员只开发一次软件，就能够在任意平台上运行，大大加快和促进了软件产品的开发。

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的主机上处理。Java 语言支持 WWW 客户机/服务器计算模式，因此，它支持这两种分布性。对于数据分布，Java 语言提供了一个称作 URL 的对象，利用这个对象，可以打开并访问 URL 地址上的对象，访问方式与访问本地文件系统相同。对于操作分布，Java 的小程序 (Applet) 可以从服务器下载到客户端，将部分计算在客户端进行，提高系统执行效率。同时，Java 语言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 语言的分布式特性。

5. 可靠性

Java 语言虽源于 C++语言，但是它消除了 C++语言中的许多不可靠因素，防止了很多编程错误，可以说 Java 语言具有很高的可靠性。首先，Java 语言是强类型的语言，要求有显式的方法说明，这就保证了编译器可以发现方法的调用错误，保证了程序更加可靠；其次，Java 语言不支持指针，这就避免了对内存的非法访问；第三，Java 语言的自动单元回收功能防止了内存丢失等动态内存分配导致的问题；第四，Java 解释器运行时实施检查，可以发现数组和字符串访问的越界；最后，Java 语言提供了异常处理机制，可以把一组错误的代码放在一个地方，这样可以简化错误处理任务，便于恢复。

6. 安全性

Java 是一种主要用于网络应用程序开发的语言，因此，对安全性要有较高的要求。如果没有安全保证，用户从网络上下载程序执行就会非常的危险。

Java 语言具有较高的安全性，它通过自己的安全机制防止了病毒程序的产生和下载程序对本地系统的威胁破坏。当 Java 字节码进入解释器时，首先必须经过字节码校验器的检查；其次，Java 解释器将决定程序中类的内存布局；再次，类装载器负责把来自网络的类装载到单独的内存区域，避免应用程序之间相互干扰破坏；最后，客户端用户还可以限制从网络上装载的类只能访问某些文件系统。综合了上述几种机制，使得 Java 成为安全的编程语言。

7. 支持多线程

线程是比进程更小的可并发执行的单位。C++语言没有内置的多线程机制，因此必须调用操作系统的多线程功能来进行多线程程序设计。而 Java 语言却提供了多线程支持。Java 语言在两方面支持多线程：一方面，Java 环境本身就是多线程的，若干个系统线程运行，负责必要的无用单元回收、系统维护等系统级操作；另一方面，Java 语言内置多线程机制，可以

大大简化多线程应用程序开发。同时, Java 语言的线程还包括一组同步原语, 这些原语负责对线程实行并发控制。利用 Java 语言的多线程编程接口, 开发人员可以方便地写出支持多线程的应用程序, 提高程序执行效率。但是, 需要注意的是, Java 语言的多线程在一定程度上会受到运行时支持平台的限制。

8. 支持网络编程

Java 语言通过它所提供的类库可以处理 TCP/IP 协议, 用户可以通过 URL 地址在网络上很方便地访问其他对象。Java 的小程序(Applet)是动态、安全、跨平台的网络应用程序。Java 的小程序嵌入在 HTML 文档中, 通过主页发布到 Internet。网络用户访问服务器的小程序时, 这些小程序从网络上进行传输, 然后在支持 Java 的浏览器中运行。由于 Java 语言的安全机制, 用户一旦载入小程序, 就可以放心地生成多媒体的用户界面或完成复杂的计算而不必担心病毒的入侵。虽然小程序可以和图像、音频、视频、动画等一样从网络上下载, 但它并不同于这些多媒体的文件格式, 它可以接收用户的输入, 动态地进行改变, 而不仅仅是动画的显示以及音频、视频的播放。

9. 编译与解释并存

用 Java 语言编写的程序称为源文件(扩展名为.java 的文件), 源文件是不能被计算机执行的。要想使程序得以运行, 必须利用编译器(不同的计算机语言有不同的编译器)对源文件进行编译, 编译器将源文件编译(即翻译)成计算机能懂的语言。对于 C 语言来说, 是针对特定的 CPU 芯片对源文件进行编译, 将源文件编译成二进制码(.exe 文件, 也被称为机器码), 这样计算机就可以读懂它, 它就可以按照人们的意愿去实现相应的功能。但是 C 语言的这种编译方式生成的目标程序就与特定的计算机有关, 一旦运行环境有所变化就可能需要重新修改源程序并针对新的运行环境重新编译, 生成新的目标程序。而 Java 语言不针对特定的 CPU 芯片进行编译, Java 提供的编译器并不是把源文件编译成二进制码, 而是将其编译成一种独立于机器平台的中间代码, 这种中间代码被称为字节码(即扩展名为.class 的文件)。字节码可以被 Java 解释器所执行, 由解释器将字节码再翻译成二进制码, 使程序得以运行。字节码非常类似于机器指令, 但字节码与具体机器是无关的, 并不能在具体的平台上执行, 而要通过 Java 运行系统中的解释器来解释执行。也就是说, Java 程序的运行要经过两个步骤来完成: 首先是由编译器将 Java 源程序编译成字节码文件, 然后再由 Java 运行系统解释执行字节码文件。这就是所谓的编译与解释并存。当然从本质上说, Java 语言属于解释型的高级程序设计语言, 但 Java 语言通过字节码的方式, 又在一定程度上克服了传统解释型语言的执行效率低, 同时又保留了解释型语言可移植的特点。所以 Java 程序的运行效率比较高。而且, 由于字节码并不专对一种特定的机器, 因此, Java 程序无须重新编译便可在多种不同的计算机上运行。

1.3 Java 技术简介

目前 Java 技术主要包括三个方面的内容。

(1) Java SE (Java Platform Standard Edition): 以前的版本称为 J2SE, 是 Java 平台的标准版, 是用于工作站、PC 机的 Java 标准平台。它体现了 Sun 公司的开放精神, 被称为是“互

联网上的世界语”。

(2) Java ME (Java Platform Micro Edition): 以前的版本称为 J2ME, 是 Java 平台的精简版, 是致力于消费产品和嵌入式设备的最佳解决方案。Java ME 是移动商务最佳的应用典范, 不论是进行无线通信, 还是手机、PDA 等小型电子装置, 均可采用 Java ME 作为开发工具及应用平台。它提供了 HTTP 等高级 Internet 协议, 可以使移动电话能以 Client/Server 方式直接访问 Internet 的全部信息, 不同的 Client 访问不同的文件, 此外还能访问本地存储区, 提供最高效率的无线交流。

(3) Java EE (Java Platform Enterprise Edition): 以前的版本称为 J2EE, 是 Java 平台的企业版, 它是以企业为环境而开发应用程序的解决方案。它提供了企业 e-Business 架构及 Web Services 服务, 其优越的跨平台能力与开放的标准, 深受广大企业用户的喜爱。目前它已经成为开发商创建电子商务应用的事实标准。

1.4 Java 虚拟机

大部分的计算机语言程序都必须先经过编译 (compile) 或解释 (interpret) 的操作后, 才能在计算机上运行, 例如 C/C++ 等是属于编译型的语言, 而 Basic 与 Lisp 等则是属于解释型的语言。然而, Java 程序 (.java 文件) 却比较特殊, 它必须先经过编译的过程, 然后再利用解释的方式来运行。通过编译器 (compiler), Java 程序会被转成与平台无关 (platform-independent) 的机器码, Java 称之为“字节码” (byte-codes), 字节码文件的扩展名为 .class。通过 Java 的解释器 (interpreter) 便可解释并运行 Java 的字节码。图 1.1 说明了 Java 程序的执行过程。



图 1.1 Java 程序的运行过程: 先编译, 后解释

字节码是 Java 虚拟机 (Java Virtual Machine, JVM) 的指令组, 和 CPU 上的微指令码很相像。它的形式为“<操作码><操作数>”, 其中操作码就是指令码。Java 语言编译成字节码后文件尺寸较小, 便于网络传输。

字节码最大的好处是可跨平台运行, 即 Java 的字节码可以编写一次, 到处运行。用户使用任何一种 Java 编译器将 Java 源程序 (.java) 编译成字节码文件 (.class) 后, 无论使用哪种操作系统, 都可以在含有 JVM 的平台上运行。这种跨越平台的特性, 也是让 Java 语言急速普及的原因之一。

任何一种可以运行 Java 字节码的软件均可看成是 Java 的“虚拟机”, 如浏览器与 Java 的开发工具等皆可视为一个 JVM。很自然的, 可以把 Java 的字节码看成是 JVM 上所运行的机器码 (machine code), 即 JVM 中的解释器负责将字节码解释成本地的机器码。所以从底层上