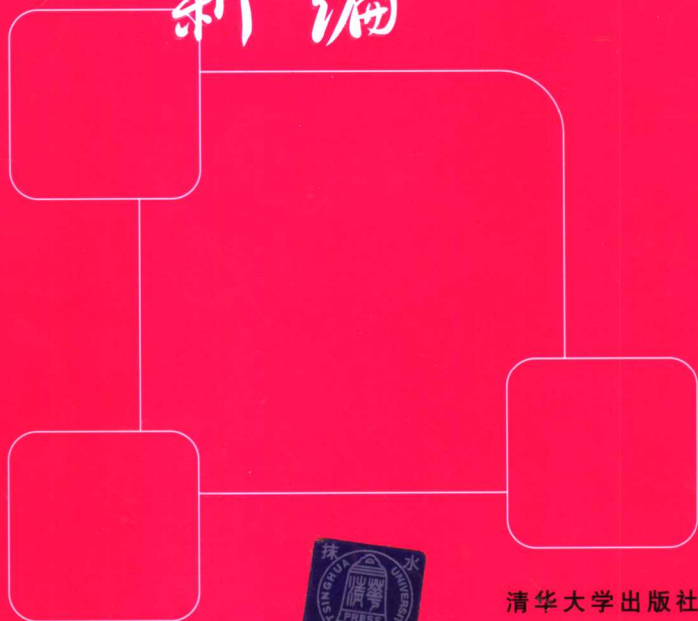


Java 基础教程 (第2版)

耿祥义 张跃平 编著

本书的配套辅导教材
《Java基础教程 (第2版) 实验练习与提高》已同期上市

新编



清华大学出版社

Java

基础教程 (第2版)

耿祥义 张跃平 编著

清华大学出版社
北京

内 容 简 介

Java 语言具有面向对象、与平台无关、安全、稳定和多线程等优良特性，是目前软件设计中极为强大的编程语言。Java 已成为网络时代最重要的语言之一。

本书以通俗易懂的语言，循序渐进地向读者介绍了 Java 语言编程的基础知识。针对较难理解的问题，所列举的例子都是由简到繁，便于读者掌握 Java 编程技巧。在第 2 版中加大了面向对象的知识容量，补充了 JDK 1.5 新增的一些重要功能。全书分为 15 章，分别讲解了基本数据类型、枚举和数组、运算符、表达式和语句、类、对象和接口、常用实用类、常用组件及事件处理、对话框、Java Applet 基础、Java 多线程机制、输入输出流、Java 网络的基本知识、泛型与集合框架以及图形与图像等内容。

本书适合初学编程或初学 Java 语言的读者使用，也可作为高等院校相关专业的教材。

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 13501256678 13801310933

图书在版编目 (CIP) 数据

Java 基础教程/耿祥义，张跃平编著.—2 版.—北京：清华大学出版社，2007.5
ISBN 978-7-302-14578-3

I. J… II. ①耿… ②张… III. JAVA 语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 010030 号

责任编辑：田在儒 王冰飞

责任校对：时翠兰

责任印制：孟凡玉

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

c-service@tup.tsinghua.edu.cn

社总机：010-62770175 邮购热线：010-62786544

投稿咨询：010-62772015 客户服务：010-62776969

印装者：北京鑫海金澳胶印有限公司

经 销：全国新华书店

开 本：185×260 印 张：23 字 数：557 千字

版 次：2007 年 5 月第 2 版 印 次：2007 年 5 月第 1 次印刷

印 数：1~5000

定 价：29.50 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103 产品编号：023241-01

导读

读者对象

本书适用于以下读者：

- 初学计算机编程语言的读者，想了解网络编程方面语言的读者；
- 有一定的其他编程语言学习基础，想进一步学习 Java 语言的读者；
- 想快速掌握 Java 语言开发技巧的读者。

写作环境

本书在写作时基于 Windows XP 操作系统平台，建议读者采用相同的系统环境学习本书。如果读者使用的是 Windows 98/2000/2003 操作系统，或者使用比本书所用版本低的软件，请参考相关的软件讲解。

学习提示

本书是《Java 基础教程》的第 2 版，在第 2 版中加大了面向对象的知识容量，补充了 JDK 1.5 新增的一些重要功能。全书分为 15 章，分别讲解了基本数据类型、枚举和数组、运算符、表达式和语句、类、对象和接口、常用实用类、常用组件及事件处理、对话框、Java Applet 基础、Java 多线程机制、输入输出流、Java 网络的基本知识、泛型与集合框架以及图形与图像等内容。本书同时配有《Java 基础教程（第 2 版）实验练习与提高》一书，它提供相应的上机实践和主教材的习题解答。

本书使用的 Java 语言开发环境 JDK 1.5，可登录到 <http://java.sun.com> 网站下载。需要本书实例源程序的读者可登录到“黑魔方”专题网站下载。

本书作者

耿祥义，1995 年获中国科学技术大学理学博士学位。1997 年从中山大学博士后流动站出站，现任大连交通大学教授。已编写出版《Java 2 实用教程》、《Java 课程设计》、《JSP 实用教程》、《JSP 基础教程》、《XML 基础教程》等 10 余本教材。

张跃平，现任大连交通大学讲师。编写和参编出版了《Visual FoxPro 课程设计》、《Java 2 实用教程》、《JSP 实用教程》3 本教材。

技术支持

如果您在阅读本书的过程中有什么困难，可以登录到“黑魔方”专题网站，网址是 <http://www.heimofang.com>。这是大家共同交流的平台，在那里会有很多的作者、老师、读者、编辑一起交流，在相关的栏目中发求助帖子，您的问题会很快得到解答。除上述方法外，也可以使用下面的方式寻求技术支持（但可能会慢一些）：

- ✉ 发电子邮件到 tianzr@tup.tsinghua.edu.cn
- ☎ 打电话给 010-62773950 或发传真给 010-62771155
- ✉ 发信到北京清华大学出版社《黑魔方丛书》编委会 收（邮编 100084）

目录

第1章 Java语言简介/1

- 1.1 Java的诞生/2
- 1.2 Java的特点/2
 - 1.2.1 简单/2
 - 1.2.2 面向对象/3
 - 1.2.3 与平台无关/3
 - 1.2.4 多线程/4
 - 1.2.5 安全/4
 - 1.2.6 动态/4
- 1.3 安装SUN公司的SDK/4
 - 1.3.1 三种平台简介/4
 - 1.3.2 安装J2SE平台/5
- 1.4 一个Java程序的开发过程/7
- 1.5 一个简单的Java应用程序/7
 - 1.5.1 编写源文件/7
 - 1.5.2 编译/8
 - 1.5.3 运行/9
- 1.6 一个简单的Java Applet程序/11
 - 1.6.1 源程序/11
 - 1.6.2 编译/12
 - 1.6.3 运行/12
- 1.7 什么是JSP/13
- 1.8 小结/13
- 习题1/13

第2章 基本类型、数组和枚举类型/15

- 2.1 标识符和关键字/16
 - 2.1.1 标识符/16
 - 2.1.2 关键字/16
- 2.2 Java语言的基本数据类型/16
 - 2.2.1 逻辑类型/16

- 2.2.2 整数类型/17
- 2.2.3 字符类型/18
- 2.2.4 浮点类型/19
- 2.2.5 基本数据类型的转换/20
- 2.3 从命令行输入、输出数据/21
- 2.4 数组/22
- 2.5 枚举类型/26
- 2.6 小结/27
- 习题2/27

第3章 运算符、表达式和语句/28

- 3.1 运算符与表达式/29
 - 3.1.1 算术运算符与算术表达式/29
 - 3.1.2 自增、自减运算符/29
 - 3.1.3 算术混合运算的精度/29
 - 3.1.4 关系运算符与关系表达式/30
 - 3.1.5 逻辑运算符与逻辑表达式/30
 - 3.1.6 赋值运算符与赋值表达式/31
 - 3.1.7 位运算符/31
 - 3.1.8 instanceof运算符/32
 - 3.1.9 运算符综述/33
- 3.2 语句概述/33
- 3.3 控制语句/34
 - 3.3.1 条件控制语句/34
 - 3.3.2 switch开关语句/36
- 3.4 循环语句/38
 - 3.4.1 for循环/38
 - 3.4.2 while 循环/38
 - 3.4.3 do-while循环/39
- 3.5 break和continue语句/39

- 3.6 数组与for语句/40
- 3.7 枚举类型与for、switch语句/41
- 3.8 小结/42
- 习题3/43

第4章 类与对象/44

- 4.1 编程语言的发展阶段/45
 - 4.1.1 机器语言/45
 - 4.1.2 过程语言/45
 - 4.1.3 面向对象编程/46
- 4.2 类/47
 - 4.2.1 类声明/48
 - 4.2.2 类体/48
 - 4.2.3 成员变量和局部变量/48
 - 4.2.4 方法/51
 - 4.2.5 方法重载/52
 - 4.2.6 构造方法/53
 - 4.2.7 类方法和实例方法/54
 - 4.2.8 两个值得注意的问题/54
- 4.3 对象/55
 - 4.3.1 创建对象/56
 - 4.3.2 使用对象/58
 - 4.3.3 对象的引用和实体/60
- 4.4 参数/61
 - 4.4.1 参数传值/61
 - 4.4.2 可变参数/63
- 4.5 static关键字/64
 - 4.5.1 实例变量和类变量的区别/64
 - 4.5.2 实例方法和类方法的区别/66
- 4.6 this关键字/66
- 4.7 包/68
 - 4.7.1 包语句/68
 - 4.7.2 import语句/70
 - 4.7.3 静态导入/74
- 4.8 访问权限/75

- 4.8.1 私有变量和私有方法/75
- 4.8.2 共有变量和共有方法/76
- 4.8.3 友好变量和友好方法/77
- 4.8.4 受保护的成员变量和方法/78
- 4.8.5 public类与友好类/79
- 4.9 基本类型的类包装/79
 - 4.9.1 Double和Float类/79
 - 4.9.2 Byte、Integer、Short、Long类/80
 - 4.9.3 Character类/80
 - 4.9.4 自动装箱与拆箱/81
- 4.10 反编译和文档生成器/82
- 4.11 JAR文件/82
- 4.12 小结/84
- 习题4/84

第5章 继承与接口/87

- 5.1 子类与父类/88
- 5.2 子类的继承性/88
 - 5.2.1 子类 and 父类在同一包中的继承性/88
 - 5.2.2 子类 and 父类不在同一包中的继承性/90
 - 5.2.3 protected的进一步说明/90
- 5.3 子类对象的构造过程/90
- 5.4 成员变量的隐藏和方法的重写/92
- 5.5 abstract类和abstract方法/93
- 5.6 final关键字/95
 - 5.6.1 final类/95
 - 5.6.2 final方法/96
 - 5.6.3 常量/96
- 5.7 对象的上转型对象/96
- 5.8 继承与多态/98
 - 5.8.1 多态的体现/98
 - 5.8.2 多态的使用/99

5.9	super关键字/102	7.1.3	创建字符串/129
5.9.1	使用super调用父类的构造方法/102	7.1.4	引用字符串常量对象/129
5.9.2	使用super操作被隐藏的成员变量和方法/103	7.1.5	String类的常用方法/130
5.10	接口/105	7.1.6	字符串与基本数据的相互转化/133
5.10.1	接口的声明与使用/105	7.1.7	对象的字符串表示/135
5.10.2	理解接口/107	7.1.8	字符串与字符、字节数组/136
5.10.3	接口回调/109	7.2	StringBuffer类/138
5.10.4	接口与多态/109	7.3	StringTokenizer类/141
5.10.5	接口做参数/112	7.4	Date类/142
5.10.6	abstract类与接口的比较/113	7.5	Calendar类/144
5.11	小结/113	7.6	Math类/147
	习题5/113	7.7	BigInteger类/148
		7.8	模式类/150
		7.8.1	正则表达式/150
		7.8.2	模式匹配/152
		7.9	小结/154
			习题7/154
第6章	内部类与异常类/116	第8章	常用组件及事件处理/157
6.1	内部类/117	8.1	窗口/159
6.2	匿名类/117	8.1.1	JFrame常用方法/159
6.2.1	和类有关的匿名类/117	8.1.2	菜单条、菜单、菜单项/159
6.2.2	和接口有关的匿名类/119	8.2	布局/161
6.3	异常类/120	8.2.1	FlowLayout布局/162
6.3.1	try~catch语句/120	8.2.2	BorderLayout布局/163
6.3.2	自定义异常类/121	8.2.3	CardLayout 布局/164
6.4	断言/122	8.2.4	GridLayout布局/165
6.5	Class类/123	8.2.5	BoxLayout布局/165
6.5.1	获取类的有关信息/123	8.2.6	null布局/167
6.5.2	使用Class实例化一个对象/125	8.3	文本框/167
6.6	小结/126	8.3.1	JTextField类的主要方法/167
	习题6/126	8.3.2	文本框上的ActionEvent事件/168
		8.3.3	文本框上的文档类型/173
第7章	常用实用类/128		
7.1	String类/129		
7.1.1	字符串常量/129		
7.1.2	声明字符串/129		

8.4	密码框/174	接口处理鼠标事件/203
8.5	按钮与标签/175	8.14.3 移动组件/203
	8.5.1 按钮/175	8.15 焦点事件/205
	8.5.2 标签/178	8.16 键盘事件/205
8.6	菜单项/178	8.16.1 使用KeyListener
8.7	文本区/179	接口处理键盘事件/205
	8.7.1 JTextArea类的 主要方法/179	8.16.2 处理复合键/207
	8.7.2 文本区上的 DocumentEvent事件/181	8.17 MVC/209
8.8	选择性组件/183	8.18 发布应用程序/212
	8.8.1 选择框/183	8.19 小结/212
	8.8.2 JRadioButton 单选按钮/184	习题8/212
	8.8.3 下拉列表/185	第9章 对话框/214
8.9	容器/187	9.1 JDialog类/215
	8.9.1 JPanel 面板/187	9.1.1 JDialog类的主要方法/215
	8.9.2 滚动窗格JScrollPane/187	9.1.2 对话框的模式/215
	8.9.3 拆分窗格JSplitPane/188	9.2 文件对话框/217
	8.9.4 JLayeredPane 分层窗格/188	9.3 消息对话框/219
	8.9.5 内部窗体 JInternalFrame/188	9.4 确认对话框/221
		9.5 颜色对话框/223
8.10	表格/190	9.6 小结/224
8.11	树/192	习题9/224
8.12	Component类的常用方法/196	第10章 Java Applet基础/225
	8.12.1 组件的颜色/196	10.1 Java Applet的运行原理/226
	8.12.2 组件的字体/196	10.1.1 用记事本编写源文件/226
	8.12.3 组件的大小与位置/198	10.1.2 编译/227
	8.12.4 组件的激活与可见性/199	10.1.3 运行/227
8.13	窗口事件/199	10.2 在Java Applet中播放声音/230
	8.13.1 WindowListener接口/199	10.3 网页传值/231
	8.13.2 WindowAdapter 适配器/200	10.4 小结/231
		习题10/232
8.14	鼠标事件/201	第11章 Java多线程机制/233
	8.14.1 使用MouseListener接口 处理鼠标事件/201	11.1 Java中的线程/234
	8.14.2 使用MouseMotionListener	

- 11.1.1 程序、进程与线程/234
- 11.1.2 线程的状态与
生命周期/235
- 11.1.3 线程调度与优先级/238
- 11.2 Thread的子类创建线程/238
- 11.3 使用Runnable接口/239
 - 11.3.1 Runnable接口与
目标对象/240
 - 11.3.2 关于run方法中的
局部变量/242
 - 11.3.3 在线程中启动
其他线程/243
- 11.4 线程的常用方法/244
 - 11.4.1 start()/244
 - 11.4.2 run()/245
 - 11.4.3 sleep(int millisecond)/245
 - 11.4.4 isAlive()/245
 - 11.4.5 currentThread()/246
 - 11.4.6 interrupt()/247
- 11.5 GUI线程/248
- 11.6 线程同步/251
- 11.7 在同步方法中使用wait()、
notify()和notifyAll()方法/254
- 11.8 挂起、恢复和终止线程/258
- 11.9 计时器线程Timer/263
- 11.10 线程联合/265
- 11.11 守护线程/266
- 11.12 小结 267
- 习题11/268
- 12.2.1 使用文件输入流
读取文件/274
- 12.2.2 处理I/O异常/275
- 12.2.3 从输入流中读取字节/275
- 12.2.4 关闭流/276
- 12.3 FileOutputStream类/276
- 12.4 FileReader和FileWriter类/277
- 12.5 BufferedReader和
BufferedWriter类/279
- 12.6 使用文件对话框打开和保存文件/281
- 12.7 RandomAccessFile类/283
- 12.8 数据流/285
 - 12.8.1 DataInputStream和
DataOutputStream类/285
 - 12.8.2 DataInputStream和
DataOutputStream的
构造方法/285
- 12.9 带进度条的输入流/286
- 12.10 对象流/287
- 12.11 序列化与对象克隆/289
- 12.12 文件锁FileLock/291
- 12.13 小结/293
- 习题12/293

第12章 输入输出流/270

- 12.1 File类/271
 - 12.1.1 文件的属性/271
 - 12.1.2 目录/272
 - 12.1.3 文件的创建与删除/273
 - 12.1.4 运行可执行文件/273
- 12.2 FileInputStream类/274

第13章 Java网络的基本知识/295

- 13.1 使用URL/296
 - 13.1.1 URL的构造方法/296
 - 13.1.2 读取URL中的资源/296
- 13.2 显示URL资源中的html文件/298
- 13.3 处理超链接/300
- 13.4 InetAddress类/302
 - 13.4.1 获取Internet上主机
的地址/302
 - 13.4.2 获取本地机的地址/303
- 13.5 套接字/303
 - 13.5.1 套接字 Socket/303
 - 13.5.2 客户建立连接到

服务器的套接字对象/303

- 13.5.3 ServerSocket类/304
- 13.5.4 把套接字连接放在一个线程中/307
- 13.6 UDP数据报/311
 - 13.6.1 发送数据包/312
 - 13.6.2 接收数据包/312
 - 13.6.3 广播数据报/316
- 13.7 小结/319
- 习题13/319

第14章 泛型与集合框架/320

- 14.1 泛型/321
 - 14.1.1 泛型类声明/321
 - 14.1.2 使用泛型类声明对象/321
 - 14.1.3 泛型接口/323
- 14.2 链表/324
 - 14.2.1 LinkedList<E>泛型类/324
 - 14.2.2 常用方法/325
 - 14.2.3 遍历链表/326
- 14.3 堆栈/331
- 14.4 散列映射/333
 - 14.4.1 HashMap<K,V>泛型类/333
 - 14.4.2 常用方法/333
 - 14.4.3 遍历散列映射/333
- 14.5 树集/334
 - 14.5.1 TreeSet<E>泛型类/334
 - 14.5.2 节点的大小关系/335
 - 14.5.3 TreeSet类的常用方法/335
- 14.6 树映射/337
- 14.7 自动装箱与拆箱/339
- 14.8 小结/339
- 习题14/340

第15章 图形与图像/342

- 15.1 绘制文本/343
- 15.2 绘制基本图形/343
 - 15.2.1 绘制直线/343
 - 15.2.2 绘制矩形/343
 - 15.2.3 绘制圆角矩形/343
 - 15.2.4 绘制椭圆/344
 - 15.2.5 绘制圆弧/344
 - 15.2.6 绘制多边形/345
- 15.3 设置字体/346
- 15.4 清除/346
- 15.5 Java 2D/347
 - 15.5.1 绘制直线/347
 - 15.5.2 绘制矩形/347
 - 15.5.3 绘制圆角矩形/347
 - 15.5.4 绘制椭圆/348
 - 15.5.5 绘制圆弧/348
 - 15.5.6 绘制二次曲线和三次曲线/349
 - 15.5.7 控制图形线条的粗细/350
 - 15.5.8 颜色渐变/351
 - 15.5.9 旋转图形/351
- 15.6 图形的布尔运算/352
- 15.7 绘制图像/354
 - 15.7.1 在小应用程序中绘制图像/354
 - 15.7.2 在应用程序中绘制图像/356
- 15.8 小结/357
- 习题15/357

第 1 章

Java 语言简介

Java 语言是由 Sun 公司于 1995 年 5 月 23 日推出的一种面向对象的编程语言。它是在 C++ 语言的基础上开发出来的，旨在提供一种简单、安全、跨平台的编程语言。Java 语言的设计理念是“一次编写，到处运行”，这使得 Java 程序可以在不同的操作系统和硬件平台上运行。Java 语言的语法与 C++ 类似，但去掉了 C++ 中的一些复杂特性，如指针、内存管理等，从而提高了程序的安全性和可移植性。Java 语言还引入了垃圾回收机制，减轻了程序员的管理内存的负担。此外，Java 语言还引入了类（Class）的概念，使得程序的组织更加清晰和模块化。Java 语言的这些特点使其成为企业级应用开发的首选语言之一。随着互联网的快速发展，Java 语言在 Web 应用开发、移动应用开发等领域得到了广泛的应用。Java 语言的生态体系也非常完善，拥有丰富的库和框架，为开发者提供了极大的便利。总的来说，Java 语言是一种强大且易于学习的编程语言，适合用于开发各种类型的软件应用。

Java 的特点

Java 语言具有以下几个主要特点：1. 简单性：Java 语言去掉了 C++ 中的一些复杂特性，使得学习更加容易。2. 面向对象：Java 语言是一种面向对象的编程语言，支持封装、继承和多态等特性。3. 跨平台性：Java 语言通过虚拟机（JVM）实现跨平台运行，真正做到“一次编写，到处运行”。4. 安全性：Java 语言引入了垃圾回收机制，避免了内存泄漏的问题。同时，Java 语言还引入了安全沙箱机制，防止恶意代码的执行。5. 健壮性：Java 语言具有严格的类型检查和异常处理机制，使得程序更加健壮。6. 多线程支持：Java 语言原生支持多线程编程，方便开发并发应用。7. 丰富的 API：Java 语言提供了大量的类库和 API，方便开发者快速开发应用。8. 社区支持：Java 语言拥有庞大的社区支持，提供了丰富的文档、教程和开发工具。



Java 具有面向对象、与平台无关、安全、稳定和多线程等优良特性，是目前软件设计中优秀的编程语言。Java 不仅可以用来开发大型的应用程序，而且特别适合于 Internet 的应用开发。Java 确实具备了“一旦写成处处可用”的特点，这也是 Java 风靡全球的主要原因。Java 不仅是一门正在被广泛使用的编程语言，而且许多新的技术领域都涉及到了 Java 语言，Java 已成为网络时代最重要的语言之一。

本章将对 Java 语言做一个简单的介绍，并初步了解什么是 Java 应用程序，什么是 Java Applet 程序，有关的细节会在后续的章节中讨论。

1.1 Java 的诞生

Java 是 1995 年 6 月由 Sun 公司引进的革命性的编程语言，之所以称 Java 为革命性编程语言，是因为传统的软件往往与具体的实现环境有关，一旦环境有所变化就需要对软件作一番改动，耗时费力。Java 语言和其他语言相比，最大的优势就是编写的软件能在执行码上兼容，可以在所有的计算机上运行。Java 之所以能做到这一点，是因为 Java 可以在计算机的操作系统之上再提供一个 Java 运行环境，即一个运行 Java 程序的平台，这个平台由 Java 虚拟机 (Java Virtual Machine)、类库以及一些核心文件组成，也就是说，只要计算机提供了 Java 运行环境，Java 编写的软件就能在其上运行。

Java 语言的出现源于对独立平台语言的需要，希望这种语言可以编写出能嵌入各种家用电器等设备的芯片上，且易于维护的程序。但是，人们发现当时的编程语言，如 C、C++ 等都有一个共同的缺点，那就是它们都针对特定的 CPU 芯片进行编译。这样，一旦电器设备更换了芯片就不能保证程序正确地运行，这就可能需要修改程序并针对新的芯片重新进行编译。1990 年 Sun 公司成立了由 James Gosling 领导的开发小组，开始致力于开发一种可移植的、跨平台的语言，该语言能生成正确运行于各种操作系统、各种 CPU 芯片上的代码。他们的精心专研和努力促进了 Java 语言的诞生。Java 的快速发展得益于 Internet 和 Web 的出现，Internet 上有各种不同的计算机，它们可能使用完全不同的操作系统和 CPU 芯片，但仍希望运行相同的程序，Java 的出现标志着真正的分布式系统的到来。



注意

印度尼西亚有一个著名的盛产咖啡的岛屿叫 Java，中文名叫爪哇，开发人员为这种新的语言起名为 Java，其寓意是为世人端上一杯热咖啡。

1.2 Java 的特点

Java 是目前使用最为广泛的网络编程语言之一。它具有简单、面向对象、稳定、与平台无关、多线程、动态等特点。

1.2.1 简单

Java 语言简单是指这门语言既易学又好用。不要将简单误解为这门语言很干瘪。用户可



能很赞同这样的观点：英语要比阿拉伯语言容易学。但这并不意味着英语就不能表达丰富的内容和深刻的思想，许多诺贝尔文学奖的作品都是英文写的。如果用户学习过 C++ 语言，会感觉 Java 很熟悉，因为 Java 中许多基本语句的语法和 C++ 一样，如常用的循环语句、控制语句等和 C++ 几乎一样，但不要误解为 Java 是 C++ 的增强版，Java 和 C++ 是两种完全不同的语言，它们各有各的优势，将会长期并存下去，Java 语言和 C++ 语言已成为软件开发者应当掌握的语言。如果从语言的简单性方面看，Java 要比 C++ 简单，C++ 中许多容易混淆的概念，或者被 Java 弃之不用，或者以一种更清楚更容易理解的方式实现，例如，Java 不再有指针的概念。

1.2.2 面向对象

基于对象的编程更符合人的思维模式，使人们更容易解决复杂的问题。Java 是面向对象的编程语言，将在第 4 章详细、准确地讨论类、对象、继承、多态、接口等重要概念。

1.2.3 与平台无关

与平台无关是指软件的运行不因处理器、操作系统的变化而无法运行或出现运行错误。其他语言编写的程序面临的一个主要问题是操作系统的变化、处理器升级以及核心系统资源的变化，都可能导致程序出现错误或无法运行。Java 语言和其他语言相比，最大的优势就是能够在所有的计算机上运行，因为 Java 可以在计算机的操作系统之上再提供一个 Java 运行环境，即一个运行 Java 程序的平台，这个平台由 Java 虚拟机 (Java Virtual Machine)、类库以及一些核心文件组成。Java 编写的程序可以在任何安装了 Java 运行环境的计算机上正确运行，Sun 公司实现了“一次写成，处处运行”的目标。

C、C++ 等语言，都是针对特定的 CPU 芯片进行编译，生成机器代码，机器代码是能被机器直接识别的一种代码，这种代码中的指令都是由 0、1 组成的序列，称为机器指令。每一条机器指令都是由计算机的硬件设计定义好的，例如，某种型号的计算机用 8 位二进制信息 10101010 表示一次加法，以 10010011 表示一次减法等。因此，C/C++ 编译生成的机器代码就和特定的 CPU 有关，一旦环境有所变化就可能需要修改源程序，并针对新的环境重新编译，生成新的机器代码。

Java 不像 C++，它不针对特定的 CPU 芯片进行编译，而是把程序编译为称做字节码的一种“中间代码”。字节码是很接近机器码的文件，可以在提供了 Java 虚拟机 (JVM) 的任何系统上被解释执行，也就是说，在一个计算机上编译得到的字节码文件可以复制到任何一个安装了 Java 运行环境的计算机上直接使用。字节码由 Java 虚拟机负责解释运行，即 Java 虚拟机负责将字节码翻译成本地计算机的机器码，并将机器码交给本地的操作系统来运行。Java 被设计成为解释执行的程序，即翻译一句，执行一句，不产生整个的机器代码程序，翻译过程如果不出现错误，就一直进行到完毕，否则将在错误处停止执行。同一个程序，如果是解释执行的，那么它的运行速度通常比编译为可执行的机器代码的运行速度慢一些。但是，对 Java 来说，二者的差别不太大，Java 的字节码经过仔细设计，使得 Java 运行环境很容易便能使用 JIT (即时编译方式) 编译技术将字节码直接转化成高性能的本地机器码，即将字节码编译成

本地机器码，并将该机器码交给本地的操作系统运行，而不必解释一句执行一句。Sun 公司的 Java 运行环境提供了这样一个字节码编译器——JIT (Just In Time)，它是 Java 虚拟机的一部分。Java 运行环境在提供 JIT 的同时仍具有平台独立性，因而“高效且跨平台”对 Java 来说不再矛盾。

1.2.4 多线程

Java 的特点之一就是内置对多线程的支持。多线程允许同时完成多个任务。实际上多线程使人产生多个任务在同时执行的错觉，因为，目前的计算机处理器在同一时刻只能执行一个线程，但处理器可以在不同的线程之间快速地切换，由于处理器速度非常快，远远超过了人接收信息的速度，所以给人的感觉好像多个任务在同时执行。C++ 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的设计。

1.2.5 安全

当用户准备从网络上下载一个程序时，用户最大的担心是程序中含有恶意的代码，例如试图读取或删除本地机上的一些重要文件，甚至该程序是一个病毒程序等。当用户使用支持 Java 的浏览器时，用户可以放心地运行 Java Applet 程序，不必担心病毒的感染和恶意的企图，Java Applet 程序将被限制在 Java 运行环境中，不允许它访问计算机的其他部分。将在第 10 章详细讲述 Java Applet 程序。

1.2.6 动态

学习了第 4 章后，大家就会知道，Java 程序的基本组成单元是类，有些类是自己编写的，有一些是从类库中引入的，而类又是运行时动态装载的，这就使得 Java 可以在分布环境中动态地维护程序及类库。C/C++ 编译时就将函数库或类库中被使用的函数、类同时生成机器码，那么每当其类库升级后，如果 C/C++ 程序想具有新类库提供的功能，程序就必须重新修改、编译。

1.3 安装 SUN 公司的 SDK

Sun 公司要实现“编写一次，到处运行 (Write once, run anywhere)”的目标，就必须提供相应的 Java 运行环境，即运行 Java 程序的平台。目前，Java 平台主要分为以下三种。

1.3.1 三种平台简介

- J2SE：称为 Java 标准版或 Java 标准平台。J2SE 提供了标准的 SDK 开发平台（以前称做 JDK 开发平台）。利用该平台可以开发 Java 桌面应用程序和低端的服务器应用程序，也可以开发 Java Applet 程序。目前最新的 SDK 版本为 J2SE SDK 1.5，Sun 公司把这一最新版本命名为 JDK 5.0，但人们仍习惯地称做 JDK 1.5。



- J2EE：称为 Java 企业版或 Java 企业平台。使用 J2EE 可以构建企业级的服务应用，J2EE 平台包含了 J2SE 平台，并增加了附加类库，以便支持目录管理、交易管理和企业级消息处理等功能。
- J2ME：称为 Java 微型版或 Java 小型平台。J2ME 是一种很小的 Java 运行环境，用于嵌入式的消费产品中，如移动电话、掌上电脑或其他无线设备等。

无论上述哪种 Java 运行平台都包括了相应的 Java 虚拟机 (Java Virtual Machine)，虚拟机负责将字节码文件 (包括程序使用的类库中的字节码) 加载到内存，然后采用解释方式来执行字节码文件，即根据相应硬件的机器指令翻译一句执行一句。

1.3.2 安装 J2SE 平台

J2SE 平台是学习掌握 Java 语言的最佳平台，而掌握 J2SE 又是进一步学习 J2EE 和 J2ME 所必需的。学习 Java 必须从 J2SE 开始，因此，本书基于 J2SE 来学习 Java。

目前有许多很好的 Java 程序开发环境可用，包括来自 Sun、Borland、Sysmanec 公司的产品，例如 JBuilder 等。但学习 Java 最好选用 Sun 公司推出的软件开发工具 SDK。学习 Java 语言可以使用 Sun 公司发布的 SDK 1.5，可以登录到 Sun 公司的网站 <http://java.sun.com>，免费下载 SDK 1.5 (例如 jdk-1_5_0_06-windows-i586-p)。如果安装 SDK 时选择安装到 F:\jdk1.5 目录下，则会生成如图 1-1 所示的目录结构。

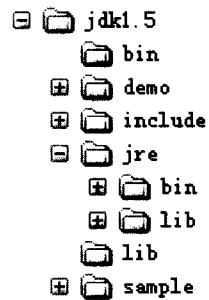


图 1-1 SDK 目录结构

现在，就可以编写 Java 程序，并进行编译、运行程序，因为安装 SDK 的同时，计算机就安装上了能运行 Java 程序的 Java 运行环境 JRE。如果只想运行别人的 Java 程序，可以只安装 Java 运行环境 JRE，JRE 由 Java 虚拟机、Java 的核心类以及一些支持文件组成。可以登录 Sun 公司的网站免费下载 Java 的 JRE。

1. 系统环境变量 Path 值的设置

SDK 安装目录的 bin 文件夹中含有编译器 (javac.exe)、解释器 (java.exe) 和一些其他的可执行文件，为了方便地使用编译器和解释器，应当将 bin 包含在 Path 的设置中。

对于 Windows 2000、Windows 2003、Windows XP，右击“我的电脑”，在弹出的快捷菜单中选择“属性”命令，弹出“系统属性”对话框，再单击该对话框中的“高级”标签，然后在“高级”选项卡中单击“环境变量”按钮，弹出“环境变量”对话框。如果未设置过 Path，可单击该对话框中的“新建”按钮，然后在“新建系统变量”对话框中添加系统环境变量，设置“变量名”为 Path，“变量值”为 F:\jdk1.5\bin。

如果曾经设置过环境变量 Path，可双击“系统变量”列表框中的 Path，对该变量进行编辑操作，将需要的值输入即可，如图 1-2 所示。也可以在命令行窗口，如 MS-DOS 窗口输入以下命令：



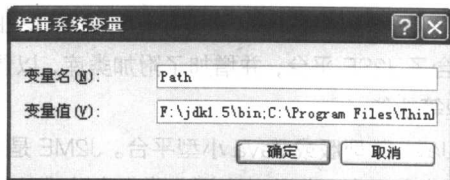


图 1-2 设置系统环境变量 Path

```
Path=F:\jdk1.5\bin;
```

对于 Win9x, 用记事本编辑 Autoexec.bat 文件, 将以下的设置语句输入即可。

```
Path=F:\jdk1.5\bin;
```

2. 系统环境变量 ClassPath 值的设置

SDK 安装目录的 jre 文件夹中包含着 Java 应用程序运行时所需要的 Java 类库, 这些类库被包含在 jre\lib 目录下的压缩文件 rt.jar 中。安装 SDK 一般不需要设置环境变量 ClassPath 的值, 如果用户的计算机安装过一些商业化的 Java 开发产品或带有 Java 技术的一些产品, 如 PB、Oracle 等, 那么这些产品在安装后可能会修改 ClassPath 的值, 当用户运行 Java 应用程序时, 可能加载这些产品所带的以前版本的类库, 可能导致程序要加载的类无法找到, 使用户的程序出现运行错误。用户可以重新编辑系统环境变量 ClassPath 的值, 对于 Windows 2000、Windows 2003、Windows XP, 右击“我的电脑”, 在弹出的快捷菜单中选择“属性”命令, 弹出“系统属性”对话框, 再单击该对话框中的“高级”标签, 然后单击“环境变量”按钮, 弹出“环境变量”对话框。如果未设置过 ClassPath, 可单击该对话框中的“新建”按钮, 在“新建系统变量”对话框中添加系统环境变量, 设置“变量名”为 ClassPath, “变量值”为 F:\jdk1.5\jre\lib\rt.jar;. , 如图 1-3 所示。

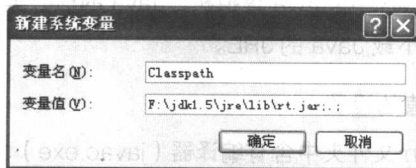


图 1-3 设置系统环境变量 ClassPath

如果曾经设置过环境变量 ClassPath, 可双击“系统变量”列表框中的 ClassPath, 对该变量进行编辑操作, 将需要的值输入即可。也可以在命令行窗口, 如 MS-DOS 窗口输入以下命令:

```
set classpath=F:\jdk1.5\jre\lib\rt.jar; .;
```

对于 Win9x, 用记事本编辑 Autoexec.bat 文件, 将以下的设置语句输入即可。

```
set classpath=F:\jdk1.5\jre\lib\rt.jar; .;
```

ClassPath 环境变量设置中的“.”是指可以加载应用程序当前目录及其子目录中的类。建议下载 Sun 公司的 Java 类库文档, 例如 jdk-1_5_0-beta2-doc。



注意

使用 SDK 环境开发 Java 程序, 需运行 MS-DOS 命令行窗口。用户需要掌握简单的 DOS 操作命令, 例如, 从逻辑分区 C 转到逻辑分区 D, 需在命令行输入 D:, 按回车键确定。进入某个子目录(文件夹)的命令是“cd 目录名”; 退出某个子目录的命令是“cd..”。例如, 从目录 example 退到目录 boy 的命令是“c:\boy>example>cd..”

1.4 一个 Java 程序的开发过程

Java 程序的开发过程如图 1-4 所示。

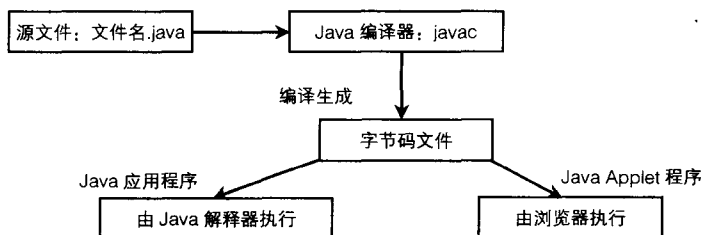


图 1-4 Java 程序的开发过程

- 编写源文件: 使用一个文本编辑器, 如 Edit 或记事本, 来编写源文件。不可使用 Word 编辑器, 因它含有不可见字符。将编写好的源文件保存起来, 源文件的扩展名必须是 .java。
- 编译 Java 源程序: 使用 Java 编译器 (javac.exe) 编译源文件, 得到字节码文件。
- 运行 Java 程序: 利用 J2SE 平台可以开发 Java 应用程序, 也可以开发 Java Applet 程序。Java 应用程序通过 Java 平台中的 Java 解释器 (java.exe) 来解释执行字节码文件; 运行 Java Applet 程序并不需要安装 J2SE 平台, Java Applet 可以由支持 Java 的浏览器直接运行, 该浏览器必须包含相应的 Java 虚拟机。

1.5 一个简单的 Java 应用程序

1.5.1 编写源文件

源文件代码如下:

```
public class Hello
{
    public static void main (String args[ ])
    {
        System.out.println("这是一个简单的应用程序");
        Student stu=new Student();
        stu.speak("We are students");
    }
}
```