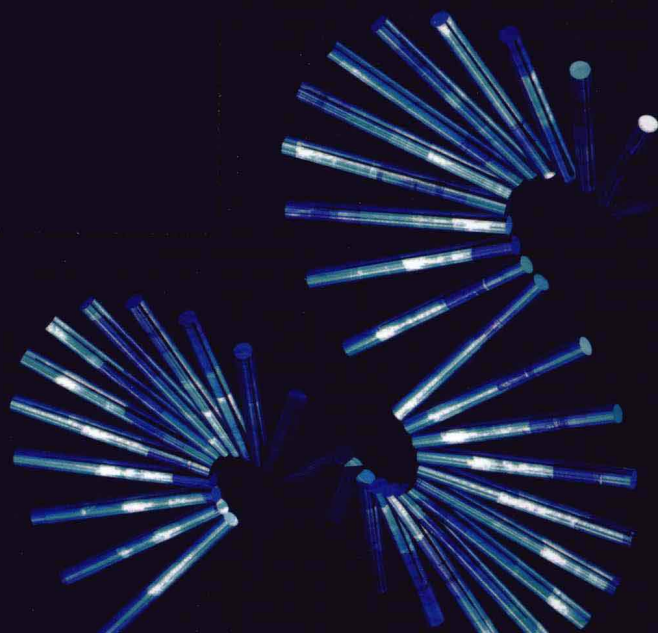
免费提供电子教案

高等院校规划教材
计算机科学与技术系列

Java程序设计

第2版

刘慧宁 等编著



机械工业出版社
CHINA MACHINE PRESS

高等院校规划教材·计算机科学与技术系列

Java 程序设计

第 2 版

刘慧宁　等编著

机 械 工 业 出 版 社

本书全面介绍了 Java SE6，反映了 Java 语言的新特性。

本书从 Java 语言的历史和特点讲起，内容由浅入深，循序渐进，涵盖了语法结构、面向对象程序设计、数组、字符串、图形用户界面、小应用程序、多媒体、异常处理、文件、数据流、枚举、泛型、集合、多线程和网络编程等主题。书中例题丰富、语言流畅、讲解透彻、层次分明、通俗易懂，同时兼顾了理论性与实用性。

本书可作为高等院校 Java 程序设计课程教材，也适合各个层次的读者自学阅读。

图书在版编目（CIP）数据

Java 程序设计/刘慧宁等编著. —2 版. —北京：机械工业出版社，2011.4
高等院校规划教材·计算机科学与技术系列
ISBN 978-7-111-33414-9

Ⅰ. ①J… Ⅱ. ①刘… 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（2011）第 022036 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：唐德凯
责任印制：杨 曦
北京四季青印刷厂印刷（三河市杨庄镇环伟装订厂装订）
2011 年 4 月第 2 版·第 1 次印刷
184mm×260mm·25.25 印张·626 千字
0001-3000 册
标准书号：ISBN 978-7-111-33414-9
定价：43.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
社服务中心：(010)88361066
销售一部：(010)68326294
销售二部：(010)88379649
读者购书热线：(010)88379203

网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com

出 版 说 明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。同时本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前　言

Java是一种面向对象的、分布式的、解释型的编程语言，具有简单、健壮、安全、体系结构中立、可移植、高效、多线程、动态等特点。自1995年5月问世以来，Java语言就被认为是最有前途的计算机编程语言，是软件开发技术演变过程中的一个里程碑。Java是一种优秀的网络编程语言，更是一种功能齐全的、通用的时代编程语言，它不仅可以用于网络程序设计，而且可以解决传统的程序设计问题，甚至还可以用于家电、通信等嵌入式设备的软件开发。现今，Java语言已经得到广泛普及，并且成为最流行、最成功的编程语言，成为全球程序设计领域使用最多的编程语言。有关Java的课程也被纳入多数高校的教学计划中。

本书是Java语言的入门教材，第2版按照Java SE6进行了全面更新，反映了Java语言的新特性，实用性强。本书通过丰富的例题、详尽的讲解，深入浅出地将Java语言介绍给读者。

本书是作者多年教学和应用开发实践的结晶。本书既具有大学教材理论严谨、概念准确、逻辑性强的特点，又具有培训教材与科技图书实用性强的优点。

通过本书学习Java语言，读者不需要具备其他任何编程语言的背景。不过，读者必须知道，程序设计课程与其他课程有很大的区别。在程序设计课程中，学生需要从例题中学习，从实践中学习，从错误中学习，需要花费大量时间编写程序、调试程序并修改错误。本书理论联系实际，书中丰富的例题可使读者在学习理论的同时，快速积累编程经验；同时，每章结尾的习题为读者提供了实践的舞台。

本书在第1版的基础上进行了大量修改，章节的安排遵循循序渐进的特点，更加合理。本书主要修改如下。

1. 按Java SE6对所有章节进行了全面改写，增强了内容表述的清晰性，修订了所有例题代码，使其更实用，便于教学。

2. 新增并完全重写了部分章节（如枚举、泛型、多线程等），其他各个章节的内容也更加充实。

3. 将部分章节的内容进行了合并和分解。例如，字符串的内容单独列为一章，小应用程序和多媒体的内容合并为一章。

4. 按照大多数读者的学习习惯，调整了部分章节的顺序。例如，将用户图形界面的内容提前。

本书所有例题均在JDK 6环境下调试运行通过。

参加本书编写工作的同志还有刘晓杰、李清华、刘蕾、向礼图、袁红霞、宛晨霞、刘光华、张德爱、孙连英、程周财、梅向伟、王之清、廖小阳、李国平、那盟、谭梅生、李幼春、周英丽等。

由于作者水平有限，书中难免会有不妥之处，敬请广大读者及专业人士指正。作者的E-mail是liuhnbox@yahoo.com.cn。

作　者

目　录

前言
第 1 章　概述 …… 1
1.1　历史沿革 …… 1
1.2　Java 语言的特点 …… 2
1.3　Java 语言与面向对象编程 …… 4
1.3.1　封装 …… 4
1.3.2　继承 …… 5
1.3.3　多态 …… 5
1.4　Java 语言与因特网 …… 5
1.5　Java 程序开发工具 …… 6
1.5.1　安装和配置 JDK …… 6
1.5.2　JDK 中的关键程序 …… 8
1.6　Java 应用程序 …… 9
1.6.1　编辑、编译和运行 Java 应用程序 …… 9
1.6.2　Java 应用程序剖析 …… 11
1.7　Java 小应用程序 …… 13
1.7.1　编辑、编译和运行 Java 小应用程序 …… 13
1.7.2　Java 小应用程序剖析 …… 15
1.8　习题 …… 15
第 2 章　Java 语言编程基础 …… 17
2.1　标识符和关键字 …… 17
2.1.1　标识符 …… 17
2.1.2　关键字 …… 17
2.2　基本数据类型 …… 18
2.3　文字常量 …… 19
2.3.1　整型文字常量 …… 20
2.3.2　浮点型文字常量 …… 20
2.3.3　字符文字常量 …… 20
2.3.4　字符串文字常量 …… 21
2.3.5　布尔文字常量 …… 22
2.4　变量 …… 22
2.5　运算符和表达式 …… 24
2.5.1　运算符 …… 25

2.5.2 算术类型转换 …… 36
2.5.3 表达式中操作数的求值顺序 …… 38
2.6 控制台输入 …… 38
2.7 控制语句 …… 39
2.7.1 语句简介 …… 40
2.7.2 条件语句 …… 40
2.7.3 循环语句 …… 44
2.7.4 跳转语句 …… 48
2.8 编码规范 …… 51
2.9 编程错误 …… 52
2.9.1 编译错误 …… 52
2.9.2 运行时错误 …… 53
2.9.3 逻辑错误 …… 54
2.10 习题 …… 55
第3章 方法 …… 58
3.1 定义方法 …… 58
3.2 调用方法 …… 60
3.3 参数传递 …… 61
3.4 递归 …… 62
3.5 方法重载 …… 64
3.6 习题 …… 67
第4章 类和对象 …… 68
4.1 抽象与封装 …… 68
4.2 类和对象的定义 …… 68
4.3 对象与基本数据类型变量的区别 …… 72
4.3.1 运算 …… 72
4.3.2 把对象传递给方法 …… 74
4.4 静态变量和静态方法 …… 75
4.4.1 静态变量 …… 76
4.4.2 静态方法 …… 77
4.5 数据成员的初始化 …… 78
4.5.1 数据成员的默认值 …… 78
4.5.2 声明时指定初值 …… 79
4.5.3 构造方法 …… 80
4.5.4 静态变量的初始化 …… 83
4.5.5 初始化块 …… 84
4.5.6 初始化小结 …… 86
4.6 包 …… 90
4.6.1 在包中添加类 …… 90

4.6.2 包的命名 …… 92
4.6.3 使用包中的类 …… 93
4.6.4 默认包 …… 95
4.6.5 JAR 压缩工具 …… 95
4.7 访问权限控制 …… 96
4.7.1 类访问权限控制 …… 96
4.7.2 成员访问权限控制 …… 96
4.7.3 不可变对象和类 …… 98
4.8 作用域 …… 99
4.8.1 类成员的作用域 …… 99
4.8.2 局部变量的作用域 …… 100
4.8.3 对象的存在时间与垃圾回收器 …… 101
4.9 关键字 this …… 102
4.10 使用类库中的类 …… 105
4.10.1 类 Math …… 106
4.10.2 类 BigInteger 和 BigDecimal …… 109
4.10.3 类 System …… 109
4.11 习题 …… 110
第5章 继承和多态 …… 115
5.1 继承简介 …… 115
5.2 继承与子类 …… 116
5.2.1 继承的语法 …… 116
5.2.2 修饰符 protected …… 118
5.2.3 子类的构造方法 …… 121
5.2.4 继承与初始化 …… 124
5.3 覆盖与隐藏 …… 127
5.3.1 方法覆盖 …… 128
5.3.2 数据成员和静态方法隐藏 …… 129
5.4 关键字 final …… 130
5.4.1 final 变量 …… 130
5.4.2 final 方法 …… 132
5.4.3 final 类 …… 133
5.5 对象类型转换和运算符 instanceof …… 134
5.6 多态与动态绑定 …… 137
5.7 抽象类和抽象方法 …… 138
5.8 接口 …… 141
5.8.1 定义接口 …… 141
5.8.2 实现接口 …… 142
5.8.3 接口继承 …… 145

5.9 类 Object …… 147
5.9.1 方法 equals …… 147
5.9.2 方法 toString …… 147
5.9.3 方法 clone …… 149
5.10 包装类和自动装箱与拆箱 …… 153
5.10.1 包装类的构造方法 …… 153
5.10.2 静态方法 …… 154
5.10.3 数据类型转换方法 …… 154
5.10.4 其他常用方法 …… 155
5.10.5 常量 MAX_VALUE 和 MIN_VALUE …… 155
5.10.6 自动装箱与拆箱 …… 156
5.11 内部类 …… 158
5.11.1 内部类举例 …… 158
5.11.2 匿名内部类 …… 160
5.12 习题 …… 161
第6章 数组 …… 165
6.1 声明和创建数组 …… 165
6.1.1 声明数组 …… 165
6.1.2 创建数组 …… 165
6.1.3 初始化数组 …… 168
6.2 多维数组 …… 170
6.3 foreach 语句 …… 172
6.4 数组与方法 …… 174
6.4.1 把数组传递给方法 …… 174
6.4.2 可变参数列表 …… 175
6.4.3 返回数组 …… 177
6.5 类 Arrays …… 178
6.5.1 数组的赋值和打印 …… 178
6.5.2 复制数组 …… 181
6.5.3 数组的比较 …… 184
6.5.4 数组的排序 …… 186
6.5.5 在数组中查找 …… 189
6.6 习题 …… 191
第7章 字符串 …… 193
7.1 类 String …… 193
7.1.1 创建 String 对象 …… 193
7.1.2 操作 String 对象 …… 194
7.2 格式化字符串 …… 199
7.3 类 StringBuilder/StringBuffer …… 201

7.3.1 创建可变字符串对象 …… 202
7.3.2 在可变字符串中追加和插入新内容 …… 202
7.3.3 其他常用操作 …… 203
7.3.4 字符串相加 …… 204
7.4 命令行参数 …… 205
7.5 被废弃的类 StringTokenizer …… 207
7.6 习题 …… 207
第8章 图形用户界面 …… 209
8.1 GUI 组件简介 …… 209
8.2 框架 …… 211
8.2.1 创建框架 …… 211
8.2.2 添加组件 …… 212
8.3 事件处理 …… 213
8.3.1 事件和事件源 …… 213
8.3.2 事件监听器 …… 215
8.3.3 监听器接口适配器 …… 218
8.4 布局管理器 …… 219
8.4.1 BorderLayout …… 219
8.4.2 FlowLayout …… 220
8.4.3 GridLayout …… 223
8.4.4 CardLayout …… 224
8.5 文本组件 …… 226
8.5.1 文本域 …… 226
8.5.2 文本区 …… 227
8.5.3 密码域 …… 227
8.5.4 面板 …… 227
8.6 选择组件 …… 229
8.6.1 按钮与标签 …… 230
8.6.2 复选框、单选按钮与边框 …… 233
8.6.3 组合框与列表 …… 238
8.6.4 滑块与进度条 …… 241
8.7 菜单组件 …… 243
8.7.1 菜单 …… 243
8.7.2 弹出式菜单 …… 246
8.8 对话框 …… 248
8.8.1 标准对话框 …… 248
8.8.2 创建自定义对话框 …… 252
8.9 绘图 …… 255
8.9.1 以面板作为画布 …… 256

8.9.2 绘制基本几何图形 …… 256
8.10 字体和颜色 …… 259
8.10.1 字体 …… 259
8.10.2 颜色 …… 261
8.11 JavaBeans 简介 …… 262
8.12 习题 …… 263
第 9 章 applet 和多媒体 …… 264
9.1 applet 小应用程序 …… 264
9.1.1 类 Applet …… 264
9.1.2 类 JApplet …… 265
9.1.3 向 applet 传递参数 …… 266
9.2 Java applet 与 application …… 267
9.2.1 applet 与 application 的不同 …… 267
9.2.2 程序作为 applet 和 application 运行 …… 268
9.3 显示图像 …… 269
9.3.1 使用类 Applet …… 269
9.3.2 使用类 ImageIcon …… 271
9.3.3 使用类 MediaTracker 跟踪图像下载 …… 273
9.4 播放动画 …… 274
9.4.1 使用类 Timer 播放动画 …… 274
9.4.2 双缓冲技术 …… 276
9.5 播放音频文件 …… 277
9.5.1 在 applet 中播放音频文件 …… 277
9.5.2 在 application 中播放音频文件 …… 280
9.6 习题 …… 280
第 10 章 异常处理 …… 281
10.1 异常简介 …… 281
10.2 异常和异常类 …… 282
10.3 抛出和声明异常 …… 284
10.4 捕获异常 …… 285
10.4.1 异常捕获的基本语法 …… 285
10.4.2 捕获所有异常 …… 287
10.5 finally 语句 …… 288
10.6 习题 …… 289
第 11 章 文件和流 …… 291
11.1 文件和流简介 …… 291
11.2 文件和目录管理 …… 291
11.3 字节流 …… 294
11.3.1 类 InputStream 和 OutputStream …… 294

11.3.2 类 FilterInputStream 和 FilterOutputStream …… 296
11.4 字符流 …… 299
11.5 随机文件访问 …… 301
11.6 标准 I/O …… 302
11.7 类 Scanner …… 303
11.8 对象 I/O 与序列化 …… 305
11.8.1 对象 I/O …… 305
11.8.2 对象序列化 …… 307
11.9 习题 …… 309
第 12 章 枚举和泛型 …… 310
12.1 枚举类型 …… 310
12.2 泛型类、接口和方法 …… 312
12.2.1 泛型的引入 …… 312
12.2.2 泛型类和接口 …… 313
12.2.3 泛型方法 …… 317
12.3 通配符 …… 319
12.4 Java 语言泛型的实现和局限性 …… 323
12.4.1 Java 语言泛型的实现 …… 323
12.4.2 Java 语言泛型的局限性 …… 324
12.5 习题 …… 326
第 13 章 集合 …… 328
13.1 集合简介 …… 328
13.1.1 集合架构 …… 328
13.1.2 泛型集合的引入 …… 331
13.2 迭代器与 foreach 语句 …… 333
13.3 接口 Collection …… 336
13.4 接口 List 及其常用实现类 …… 340
13.5 接口 Queue、Deque 及其常用实现类 …… 343
13.5.1 接口 Queue 及其常用实现类 …… 343
13.5.2 接口 Deque 及其常用实现类 …… 345
13.6 类 Collections …… 347
13.7 遗留的集合类型 …… 349
13.8 习题 …… 350
第 14 章 多线程 …… 351
14.1 线程简介 …… 351
14.2 创建任务和线程 …… 351
14.3 线程属性 …… 353
14.3.1 线程优先级 …… 353
14.3.2 守护线程 …… 353

14.4 线程池 …… 355
14.5 异常与线程 …… 356
14.6 共享资源 …… 357
14.6.1 共享资源冲突 …… 357
14.6.2 使用 Lock 锁实现同步 …… 359
14.6.3 使用关键字 synchronized 实现同步 …… 361
14.6.4 线程间协作 …… 363
14.6.5 死锁 …… 366
14.7 线程的状态 …… 367
14.8 线程安全的类 …… 369
14.9 Swing 与线程 …… 370
14.10 习题 …… 374
第 15 章 网络编程 …… 375
15.1 计算机网络基础 …… 375
15.1.1 客户端及服务器 …… 375
15.1.2 IP 地址 …… 375
15.1.3 端口 …… 377
15.1.4 套接字 …… 377
15.2 流套接字通信 …… 377
15.2.1 流套接字连接的建立过程 …… 378
15.2.2 简单的服务器与客户端程序 …… 379
15.2.3 服务多个客户 …… 381
15.3 数据报 …… 383
15.4 读取服务器上的文件 …… 384
15.5 浏览网页 …… 387
15.5.1 使用 applet 浏览网页 …… 387
15.5.2 创建一个简单的浏览器 …… 388
15.6 习题 …… 390
参考文献 …… 391

第1章 概 述

Java是一种功能强大的编程语言，在软件开发技术的演变过程中，它的出现可以说是一个里程碑。Java语言不仅可以解决传统的程序设计问题，更主要的是它与互联网密切相关，可以使用它开发因特网（Internet）上的程序。

本章介绍Java语言的历史和特点、Java语言与面向对象编程、Java语言与因特网、Java程序开发工具、Java程序的开发过程及简单Java程序的组成。

1.1 历史沿革

自从个人计算机（PC）问世以来，计算机的应用领域迅速扩大，全球使用计算机的人数快速增长。计算机技术正在不断地改变着人们的生活和工作。

1991年，为了将计算机技术引入人们的日常家庭生活，Sun公司（2009年，Oracle公司收购了Sun公司）启动了一个代号为“Green”的内部研究计划。该计划旨在使家电计算机化，能够相互通信，从而达到以智能化方式控制家居环境的目的。

要为家用电子设备编程，就需要一种编程语言。由于不同厂商生产的家用电子设备芯片很可能不同，因此，为家用电子设备编写的软件应该占用内存少，而且能工作在不同的芯片上。当时的计算机语言（包括C、C++）都不能满足这个要求。因此，计划负责人James Gosling领导计划组以C++语言为基础设计了一种新的程序设计语言，并受办公室窗外一大片橡树的启发，将其命名为Oak。后来，计划组成员发现Oak已经是另一种程序设计语言的名字，于是，该语言被更名为Java。据说，Java这个名字是计划组成员在喝咖啡时想到的。

或许Sun公司的预测过于乐观，事实上，市场对智能化家电的需求并没有快速增加。Green计划及与之相关的Java语言也面临夭折的命运。

峰回路转，1993年互联网开始蓬勃发展。Sun公司意识到Green计划与互联网技术的高度类似性，而Java语言的中立性使得Java程序能在各种不同的计算机平台上运行，非常适合编写网络应用软件。因此，Sun公司改变了研究方向，Java语言也随之得到了极大的发展。

借助互联网风暴，Sun公司于1995年5月正式推出了Java语言，并在全世界引起了轰动。不过，对于Java语言而言，软件产业中的“三次改版”法则（产品若不经过三次改版，就得不到很好的质量）同样适用。1996年初，Java语言的第1个版本（通常称作Java 1.0，其对应的开发工具包版本是1.0，英文为Java Development Kit 1.0，简称JDK 1.0）发布后，人们很快意识到它不能用来进行真正的应用开发。Java 1.1弥补了其中的大多数明显缺陷，但它仍然具有很大的局限性。直到1998年12月，Java 1.2的发布才使该语言真正成熟，为了突出Java 1.2与旧版本Java之间的重大差异，Sun公司甚至将Java语言更名为Java 2（相对应，其开发工具包版本被更名为Java 2 Platform Standard Edition Software Development Kit 1.2，Java 2标准版软件开发工具包1.2版，简称J2SE SDK 1.2，不过，程序员通常还是将

其简称为JDK 1.2)。随后，Sun公司还推出了两个与Java标准版（Java Standard Edition，Java SE）配套的版本，其中，一个是用于服务器端应用开发的企业版（Java Enterprise Edition，Java EE)；另一个是用于手机等嵌入式设备软件开发的微型版（Java Micro Edition，Java ME)。本书主要介绍Java标准版，它主要用于工作站和个人计算机应用开发，也可用于简单的服务器应用开发。

Java 2发布后，Java语言的发展并未停止。在不断的改版过程中，Java语言一直以降低编程复杂度为主要目标，修正不足及缺点，添加新功能。Java 1.3和Java 1.4对Java 2作了一些改进，比如，修正了原有版本的bug，扩展了标准类库，提高了系统性能。Java 1.5（2004年，JavaOne会议后，这一版本又被改称作Java 5.0）对Java 2作了重大改进，特别是添加了泛型类型，增强了多线程编程能力等。2006年底，Sun公司发布了本书编写时的Java语言最新版本Java 6（个别文献将其称作Java 1.6)，这一版本没有在语言方面再进行改进，但是改进了速度等其他性能，并扩展了标准类库。同时，也是在2006年，为了避免语言命名与语言版本号给人们造成困惑，Java 2这一命名被弃用。

1.2 Java语言的特点

Java语言之所以流行，是由它的特点决定的。具体地说，Java语言有如下特点。

1. 简单性

Java语言是从C++语言衍生来的，但它在C++语言的基础上作了重大改进，它摒弃了C++语言中许多很少使用、难以理解、易混淆的特性，例如头文件、指针、结构、联合、运算符重载、虚基类、多继承等。而且，Java语言还提供了一个被称作“垃圾回收器”的机制来自动进行内存管理。因此，Java语言比C++语言简单得多，更加容易学习和使用。

如果读者具有C++语言的背景知识，学习Java语言将是一件愉快的事。当然，这不是必需的，通过本书学习Java语言并不需要C++语言的基础。

2. 面向对象

Java语言是一种杂合型语言。不过，与C++语言相比，它是一种更纯粹的面向对象的程序设计语言。面向对象程序设计（Object-Oriented Programming，OOP）就是使用对象进行程序设计。Java语言中，除了基本数据类型外，可以说一切都是对象。世界上的任何事物都能抽象成对象，例如，一个人、一幅图、一个圆、一场比赛都可以看作一个对象。有关面向对象的知识将在1.3节详细介绍。

3. 分布性

Java语言提供的标准类库可以处理TCP/IP协议。使用Java语言可以很方便地通过统一资源定位符（Uniform Resource Locator，URL）访问网络上的其他对象，取得用户需要的资源，其便捷程度就好像访问本地文件一样。因此，Java语言非常适合因特网和分布式环境下的编程。

4. 解释型

Java源程序经编译后生成类文件。类文件由字节码组成，字节码是一种虚拟的机器指令代码，不针对特定的机器。运行时，Java解释程序（对程序员来说，可以将它简单地对应为Java虚拟机，即Java Virtual Machine，简称JVM，Java虚拟机是执行Java程序的软件平

台）负责将字节码解释成本地机器指令代码。

5. 健壮性

Java 语言摒弃了 C、C ++ 语言中的指针数据类型，不允许对内存进行直接操作。Java 语言的垃圾回收器能自动进行内存管理，防止程序员在管理内存时产生错误。同时，Java 系统在编译和运行程序时，都对可能出现的问题进行检查，以减少错误的产生。即使 Java 程序运行时真出现错误，Java 语言集成的异常处理机制也能确保程序不会崩溃，而是抛出一个异常并作相应处理。

6. 安全性

Java 语言是一种网络语言，大量用于网络和分布式环境下的软件开发。因此，对安全性有更高的要求。Java 语言设计了多层安全机制，使得网络和分布式环境下的 Java 程序不会充当攻击本地资源的病毒或其他恶意操作的传播者，确保了安全。

7. 体系结构中立

Java 源程序编译后生成的类文件与平台无关。因此，Java 程序编译生成的类文件可以在任意机器平台上运行，只要那台机器上安装了相应的 Java 解释程序即可。而其他大多数语言程序（如 C、C ++ 等）则需要针对特定的机器平台重新编译。

8. 可移植性

体系结构中立性使得 Java 程序不必重新编译就能在不同的平台上运行。同时，Java 语言对不同平台使用了完全统一的语言文本。例如，每种基本数据类型所占存储空间的大小不会随机器平台或编译器不同而改变，int 型总是 32 位的整数，long 型总是 64 位的整数。而 C、C ++ 等语言并非如此，它们的数据类型所占存储空间随着机器平台或编译器的变化可能略有不同。

Java 语言提供的标准类库可以访问不同平台的基本操作系统。使用这些标准类库后，Java 程序可在支持 Java 语言的任意平台上运行。此外，Java 编译程序是用 Java 语言编写的，这使得 Java 系统本身也具有可移植性。

以上机制保证了 Java 语言的可移植性，从而实现了软件的“一次开发，随处运行”，开创了程序设计的新时代。

9. 高性能

Java 语言比典型的解释型语言和脚本语言运行速度快，但是，如果使用最原始的 Java 解释程序，Java 语言的运行速度大概只有 C、C ++ 语言的 1/20。然而，由于计算机硬件速度不断提高，大多数应用程序都能接受这种速度。

不过，新的 Java 开发工具采用了“实时编译”技术，并且 Sun 公司推出了“hotspot”运行引擎技术。随着这些新技术的应用，目前，Java 程序的运行速度已大大提高，甚至接近 C、C ++ 程序的运行速度。而且，Java 程序的运行速度还在不断提升之中。

使用 Java 语言，可以大幅缩短软件开发周期。与使用 C、C ++ 语言开发相同的程序相比，一般来说，使用 Java 语言的开发周期可以缩短一半以上，这极大地提升了程序员的生产效率。这也是许多 C ++ 程序员转而使用 Java 语言的主要原因之一。

10. 多线程

线程是指一个程序中可以独立运行的片段。多线程处理能使同一程序中的多个线程同时运行，即程序中多个任务并发执行。

Java 语言内建多线程机制，同时还提供同步机制保证对共享资源的正确操作。因此，与其他语言相比，Java 语言的多线程编程更加便捷。

11. 动态性

Java 语言的设计使它能适应不断发展的环境。在一个类中可以自由地加入新的方法和数据成员而不会影响到原来使用该类的程序的运行。此外，与其他许多语言的程序在启动过程中便会全部被加载到内存不同，运行 Java 程序时，每个类文件只有在必要时（即第 1 次使用这个类时）才被加载。在第 4 章，读者会知道 Java 程序中的每个类经编译后都会生成一个独立的类文件。

1.3 Java 语言与面向对象编程

计算机不懂人类的语言，所以需要使用编程语言与计算机进行交流。使用编程语言编写的程序（即通常所说的软件）就是对计算机操作的指令，通过程序告诉计算机做什么。没有程序，计算机就是一台裸机。

简单地说，面向对象编程是一种程序设计技术。在面向对象编程之前，大部分常用的编程语言是面向过程的（如 FORTRAN、BASIC、C 等）。在面向过程程序设计中，数据和处理数据的过程分别存储于不同的地方，数据和过程之间没有逻辑或组织上的联系。这种编程方法在编写小型和中型程序时，表现得相当好。不过，由于其数据和过程分离，对于较大的程序，开发和维护就格外复杂。

人们吸取面向过程程序设计的优点，改变了程序中数据和处理数据的过程分离的状况，提出了面向对象的程序设计方法。面向对象编程是程序设计方法的巨大变化，是当今最流行的编程方法。其本质是把数据和处理数据的过程抽象成一个具有特定身份和某些属性的自包含实体——对象。

面向对象系统最突出的特点是封装性、继承性和多态性。

1.3.1 封装

组装计算机时，通常只需关心相应组件的接口规范，并不用关心它的内部构造和工作原理。这里，装配人员所关心的计算机组件只是一个“黑匣子”，只要符合规范就可以，即计算机各组件对于装配人员来说是一个自包含实体，是封装的。这种无需知道封装单元是如何工作，就能使用的思想被称作数据隐藏。

面向对象编程的核心就是封装。在面向对象的程序设计中，封装是一种数据隐藏技术，它通过把一组数据和与数据有关的操作集合放在一起形成对象来实现。对象通过特定的接口与外部发生联系，而内部的具体细节则被隐藏起来，对外是不可见的。封装的目的就是防止非法访问，用户只能通过对象的接口使用对象提供的服务，看不到其中的具体实现细节。

Java 语言中，封装的基本单元是类。类是数据及其相关操作的封装体，是对对象的抽象和描述，对象是类的实例。一个类的所有对象都具有相同的数据结构和操作代码。就像按照同一张汽车设计图纸，可以造出许多具体的小汽车。换成计算机语言，就是使用一个汽车类可以创建多个汽车对象。

类是数据及其相关操作的封装体。类中的具体操作细节被封装起来，用户在使用一个已