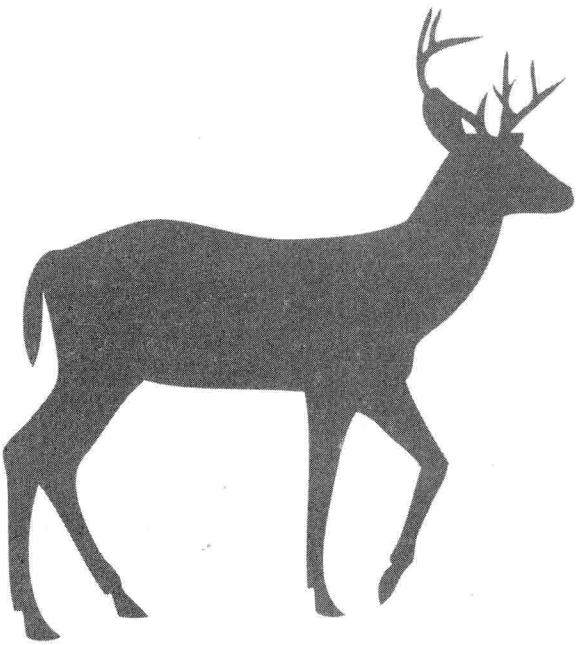
一线教师多年教学经验与工程实践的倾力之作，读者自学与课堂教学的经典读本

Java 程序设计经典课堂

金松河 王捷 黄永丽 编著

- **内容详尽** 对Java程序设计中方方面面的知识点进行了详细解读
- **浅显易懂** 知识引入循序渐进，讲解通俗易懂，学练结合可快速上手
- **超强平台** 书中所有实例程序都在Java SE 7.0环境下调试，可直接使用
- **实例丰富** 本书不仅理论完备，还通过150多个实例夯实基础，并用2个综合型实战项目全面提升实战开发能力

清華大學出版社



Java 程序设计经典课堂

金松河 王捷 黄永丽 编著

清华大学出版社

内 容 简 介

本书全面而系统地介绍了 Java 语言的相关知识，内容循序渐进，讲解通俗易懂，通过一百多个实例将理论与实践相结合，帮助读者轻松掌握 Java 语言编程方法。

本书共 14 章，由浅入深地对 Java 程序设计语言进行了全面地讲解，主要知识点包括 Java 语言的特点、Java 程序的运行与开发环境、Java 语言的基本语法、面向对象编程方法，Java 类的定义、成员变量与成员方法、构造方法、Java 对象的生成与使用、方法参数传递、访问控制、泛型、常用类和接口、继承与多态性、异常处理、图形用户界面设计、常用 Swing 组件、输入/输出流、多线程编程、数据库编程、网络编程等。最后还通过进销存管理系统与图书馆管理系统的开发设计，综合运用书中涉及的知识，使读者不仅可以温故知新，还能提高 Java 语言的综合编程能力。

本书不仅可以作为各类院校和社会培训机构的首选教材，还可以作为 Java 程序设计自学者和编程爱好者的入门指导用书。

图书在版编目（CIP）数据

Java 程序设计经典课堂/ 金松河，王捷，黄水丽编著. —北京：清华大学出版社，2014
ISBN 978-7-302-36635-5
Ⅰ. ①J… Ⅱ. ①金…②王…③黄… Ⅲ. ①JAVA 语言—程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2014）第 113432 号

责任编辑：杨如林
封面设计：姜小雨
责任校对：胡伟民
责任印制：刘海龙

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
印 刷 者：三河市君旺印务有限公司
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：190mm×260mm　　印　　张：29.75　　字　　数：800 千字
版　　次：2014 年 8 月第 1 版　　印　　次：2014 年 8 月第1 次印刷
印　　数：1～3000
定　　价：59.00 元

产品编号：059782-01

前　言

Java 是一种功能强大的程序设计语言，以其面向对象和跨平台的特性风靡全球，它是目前国内外使用最为广泛的程序设计语言。Java 技术已经成为当今世界最流行的开发工具和主流技术。

现在市面上有关 Java 的书很多，但其质量良莠不齐。为此，我们组织一线教师编写了本书——《Java 程序设计经典课堂》，作者结合了自己多年的教学经验和工程实践经验，力图使本书成为最适合课堂教学以及自学使用的读物。本书具有以下特点。

（1）循序渐进，对新概念的引入和讲解循序渐进，逐步展开，确保读者能够更加容易理解和掌握这些新的概念。

（2）浅显易懂，书中在讲解理论的同时，还列举了一百多个实例，通过模仿、练习这些实例，可以帮助读者在较短的时间内掌握较多和较为复杂的知识。

（3）综合应用，书中的综合型实例可以使读者真正掌握 Java 应用程序开发的一般方法和过程，与行业应用接轨。

（4）强调实训，每章结尾的项目实训更偏向于锻炼读者的思维能力与动手能力，从而增强读者对知识的应用能力。

（5）超强平台，全书所有实例程序都在 Java SE 7.0 环境下调试运行通过，读者可以直接参照使用。

全书共分 14 章，其内容如下所述。

章节	章名	内容概述
第 1 章	初识 Java 语言	主要介绍了 Java 语言的发展史和特点，以及 Java 程序的运行开发环境的构建等内容
第 2 章	Java 语言入门必备	主要介绍了标识符和关键字、基本数据类型、常量与变量、运算符、控制语句和数组等内容
第 3 章	面向对象程序设计(基础篇)	主要介绍了面向对象编程的基本概念、Java 类的定义、成员变量与成员方法、构造方法、Java 对象的生成与使用、方法参数传递、常用于修饰类或方法的修饰符、包和接口等内容
第 4 章	面向对象程序设计(高级篇)	
第 5 章	常用类和接口	主要介绍了一些常用的类和接口
第 6 章	Java 异常处理	主要介绍了 Java 的异常处理机制
第 7 章	图形用户界面设计	主要介绍了如何利用 Java 的图形组件创建用户界面
第 8 章	常用 Swing 组件	
第 9 章	输入输出流	主要介绍了常用的输入与输出流

章节	章名	内容概述
第 10 章	多线程编程	主要通过针对性的例子介绍多线程这一重要概念
第 11 章	数据库编程	主要介绍了数据库编程技术，首先介绍数据库相关的基础知识，进而讲述如何在 Java 程序中连接数据库，并实现数据的存取
第 12 章	网络编程技术	主要介绍了 Java 在网络编程方面的应用技术
第 13 章	进销存管理系统	这两个实例综合运用了本书各章节的知识和技术，既是对全书介绍的知识点的一个复习，还可以提升读者综合应用 Java 编程技术的能力，这将进一步提高读者学习 Java 的兴趣
第 14 章	图书馆管理系统	

本书内容翔实、示例丰富、结构合理、图文并茂，非常适合以下读者群体使用：

- 初学编程的自学者；
- 大中专院校的老师和学生；
- 相关培训机构的老师和学员；
- 初、中级程序开发人员；
- 程序测试及维护人员；
- 编程爱好者。

本书配套资源请读者登录清华大学出版社网站（Http://www.tup.com.cn），搜索到本书页面后按提示下载。

- 本书实例源代码。
- 本书配套教学 PPT。

本书由郑州轻工业学院的金松河老师、王捷老师和黄永丽老师编著，参加本书编写的还有朱会东、宋宝卫、朱付保等老师。其中，全书由钱慎一老师统稿，附录部分由王国胜老师编写。在此，对他们的工作表示衷心的感谢。最后特别感谢郑州轻工业学院教务处对本书的大力支持。

由于编写时间仓促，加之作者水平有限，恳请广大读者给予批评指正。

编者

目　录

第 1 章　初识 Java 语言 1
1.1　Java 语言发展史及特点 1
1.1.1　Java 语言简介 1
1.1.2　Java 语言特点 3
1.2　Java 开发环境的配置 5
1.2.1　JDK 的安装 5
1.2.2　系统环境变量的设置 8
1.2.3　编译和执行程序命令用法 8
1.2.4　集成开发环境 Eclipse 介绍 10
1.3　Java 应用程序示例 14
知识回顾 15
项目实训 16
第 2 章　Java 语言入门必备 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　常量和变量 18
2.2.1　常量 18
2.2.2　变量 18
2.2.3　变量作用域 19
2.3　基本数据类型 20
2.3.1　4 大类基本数据类型 20
2.3.2　数据类型转换 22
2.4　运算符和表达式 23
2.4.1　赋值运算符与赋值表达式 23
2.4.2　算术运算符与算术表达式 23
2.4.3　关系运算符与关系表达式 25
2.4.4　逻辑运算符与逻辑表达式 25
2.4.5　位运算符 26
2.4.6　移位运算符 26
2.4.7　条件运算符 28
2.4.8　运算符的优先级与结合性 28
2.5　控制语句 29
2.5.1　分支语句 29
2.5.2　循环语句 33
2.5.3　跳转语句 38
2.6　Java 注释语句 39
2.7　数组 40
2.7.1　一维数组 40
2.7.2　二维数组 42
2.7.3　数组使用举例 43
知识回顾 45
项目实训 46
第 3 章　面向对象程序设计(基础篇) 47
3.1　面向对象程序设计概述 47
3.1.1　面向过程程序设计 47
3.1.2　面向对象程序设计 48
3.1.3　面向对象方法的特征 48
3.2　类与对象 51
3.2.1　类的声明 51
3.2.2　类的实例对象及使用 54
3.3　构造方法 57
3.3.1　构造方法的定义与作用 57
3.3.2　构造方法的重载 58
3.3.3　构造方法的一些细节 60
3.4　this 引用句柄 61
3.5　垃圾回收机制 64
3.5.1　finalize 方法 64
3.5.2　System.gc 的作用 64
3.6　方法中的参数传递 65
3.6.1　基本数据类型的参数传递 65
3.6.2　引用数据类型的参数传递 66
3.6.3　命令行参数 67
3.7　访问控制 68
3.7.1　Java 中的访问控制 68
3.7.2　static 关键字的使用 69
3.7.3　final 关键字的使用 72
知识回顾 73
项目实训 73
第 4 章　面向对象程序设计(高级篇) 75
4.1　继承和多态性的概念 75
4.1.1　继承的概述 75
4.1.2　多态性的概述 77
4.2　继承机制 77

4.2.1 继承的定义 77
4.2.2 继承的传递性 81
4.2.3 类中属性的继承与隐藏 82
4.2.4 类中方法的继承、覆盖与重载 82
4.2.5 在子类中使用构造方法 87
4.3 多态性 89
4.3.1 多态性的体现 89
4.3.2 静态多态性 89
4.3.3 动态多态性 89
4.3.4 父类对象与子类对象间的指代使用和转化 90
4.4 抽象类 92
4.4.1 抽象类的定义 92
4.4.2 抽象类的使用 93
4.5 包与接口 98
4.5.1 Java 中的包 98
4.5.2 Java 中的接口 101
4.6 内部类 105
4.7 匿名类 106
知识回顾 107
项目实训 108
第 5 章 常用类和接口 111
5.1 字符串处理类 111
5.1.1 String 类 111
5.1.2 StringBuffer 类 119
5.2 泛型 122
5.2.1 泛型类 122
5.2.2 泛型方法 124
5.2.3 通配类型参数 125
5.3 集合框架 126
5.3.1 Collection 接口及操作 127
5.3.2 Set 接口及其实现类 128
5.3.3 对象排序 131
5.3.4 List 接口及实现类 134
5.3.5 Map 接口及其实现类 137
5.3.6 集合的输出 141
5.3.7 集合的工具类 Collections 144
5.4 时间日期类 146
5.4.1 Date 类 146
5.4.2 Calendar 类 147
5.4.3 DateFormat 类 148
5.4.4 SimpleDateFormat 类 149
5.5 Math 类 150
5.5.1 Math 类的属性和方法 150
5.5.2 Math 类的应用示例 151
5.6 随机数处理类 Random 152
5.7 系统类 System 和 Runtime 152
5.7.1 System 类 152
5.7.2 RunTime 类 155
知识回顾 157
项目实训 157
第 6 章 Java 异常处理 159
6.1 异常概述 159
6.1.1 Java 的异常处理机制 160
6.1.2 Java 中异常类的层次结构 161
6.2 异常处理 163
6.2.1 使用 try 和 catch 捕获和处理异常 163
6.2.2 使用 throws 子句声明异常 170
6.2.3 throw 语句 171
6.2.4 使用异常处理语句的注意事项 172
6.3 自定义异常 173
知识回顾 175
项目实训 175
第 7 章 图形用户界面设计 177
7.1 GUI 编程基础 177
7.1.1 AWT 与 Swing 的关系 177
7.1.2 GUI 元素的分类 178
7.2 常用容器类 179
7.2.1 顶层容器类 179
7.2.2 中间容器类 181
7.3 布局管理器 184
7.3.1 FlowLayout 184
7.3.2 BorderLayout 185
7.3.3 GridLayout 187
7.3.4 CardLayout 189
7.3.5 BoxLayout 190
7.4 Java 的 GUI 事件处理 194
7.4.1 事件处理的基本过程 194
7.4.2 常用的事件类及其监听者 197
7.5 多监听程序与事件适配器 199
7.5.1 窗口事件的处理 200
7.5.2 事件适配器 201
7.5.3 键盘事件的处理 202
7.5.4 鼠标事件的处理 203
知识回顾 206
项目实训 206
第 8 章 常用 Swing 组件 207
8.1 常用 Swing 组件介绍 207

8.1.1 标签.....207
8.1.2 文本组件.....208
8.1.3 按钮组件.....211
8.1.4 组合框.....218
8.1.5 列表框.....220
8.1.6 表格.....222
8.2 菜单组件.....224
8.2.1 菜单组件概述.....224
8.2.2 下拉式菜单.....225
8.2.3 弹出式菜单.....228
8.3 对话框.....229
8.3.1 Swing 中的 JDialog.....229
8.3.2 标准对话框.....230
8.3.3 文件对话框.....233
知识回顾.....235
项目实训.....235
第 9 章 输入输出流.....237
9.1 文件和目录的操作.....237
9.2 输入输出流概述.....240
9.2.1 流的基本概念.....240
9.2.2 流类的层次结构.....240
9.2.3 流类的基本用法.....242
9.3 常用流类.....243
9.3.1 字节输入流.....243
9.3.2 字节输出流.....246
9.3.3 字符输入流.....251
9.3.4 字符输出流.....255
9.3.5 流的转换.....258
9.4 随机文件访问类.....260
9.5 对象序列化.....261
9.5.1 序列化的概念.....261
9.5.2 ObjectOutputStream 类.....262
9.5.3 ObjectInputStream 类.....263
9.5.4 序列化示例.....263
9.5.5 定制序列化.....265
知识回顾.....265
项目实训.....266
第 10 章 多线程编程.....267
10.1 多线程概述.....267
10.1.1 线程基本概念.....267
10.1.2 线程的运行机制.....267
10.2 线程的创建和启动.....269
10.2.1 线程的创建.....269
10.2.2 线程的启动.....272
10.3 线程状态和转换.....274
10.4 线程控制.....275
10.4.1 线程睡眠.....275
10.4.2 线程让步.....277
10.4.3 线程间协作.....278
10.4.4 后台线程.....280
10.4.5 线程优先级.....281
10.5 线程同步处理.....281
10.5.1 多线程引发的问题.....282
10.5.2 同步代码块.....283
10.5.3 同步方法.....286
10.5.4 线程间通信.....288
10.5.5 死锁.....291
知识回顾.....293
项目实训.....293
第 11 章 数据库编程.....295
11.1 数据库基础.....295
11.1.1 关系数据库.....295
11.1.2 数据的定义.....295
11.1.3 数据的操纵语言.....296
11.1.4 数据查询语言.....297
11.2 JDBC 基础.....297
11.2.1 JDBC 简介.....297
11.2.2 JDBC 驱动类型.....299
11.3 使用 JDBC 访问数据库.....300
11.3.1 JDBC 使用基本流程.....300
11.3.2 数据库驱动程序的加载.....301
11.3.3 连接数据库.....302
11.3.4 数据库的操纵.....304
11.3.5 操作结果的处理与访问.....307
11.3.6 JDBC 的关闭操作.....309
11.4 数据库编程实例.....310
11.4.1 数据连接的创建.....310
11.4.2 数据库表的创建.....312
11.4.3 添加数据信息.....313
11.4.4 数据信息的修改.....315
11.4.5 数据信息的删除.....316
11.4.6 数据信息的查询.....317
11.5 事务的处理.....318
知识回顾.....321
项目实训.....321
第 12 章 网络编程技术.....323
12.1 网络通信基础.....323
12.1.1 基本概念.....323

12.1.2 通信协议 324
12.1.3 Java 网络编程技术 325
12.2 URL 程序设计 325
12.2.1 URL 326
12.2.2 URLConnection 327
12.2.3 InetAddress 329
12.3 TCP 程序设计 330
12.3.1 网络套接字 331
12.3.2 Socket 331
12.3.3 ServerSocket 333
12.3.4 TCP 编程实例 335
12.4 UDP 程序设计 339
12.4.1 数据报通信 339
12.4.2 DatagramPacket 340
12.4.3 DatagramSocket 341
12.4.4 MulticastSocket 347
知识回顾 353
项目实训 354
第 13 章 进销存管理系统 355
13.1 进销存管理系统概述 355
13.1.1 开发背景 355
13.1.2 需求描述 355
13.2 系统设计 356
13.2.1 系统目标 356
13.2.2 系统功能结构 356
13.2.3 开发环境 357
13.2.4 文件夹组织结构 357
13.2.5 数据库设计 357
13.3 系统实现 360
13.3.1 主窗体设计 360
13.3.2 基础信息模块设计 360
13.3.3 进货管理模块设计 361
13.3.4 查询统计模块设计 362
13.3.5 库存管理模块设计 363
13.3.6 Java 类的设计 363
13.4 系统打包与发布 396
知识回顾 397
第 14 章 图书馆管理系统 399
14.1 系统概述 399
14.1.1 项目背景 399
14.1.2 业务描述 399
14.2 系统设计 400
14.2.1 系统框架 400
14.2.2 角色分析 401
14.2.3 数据库设计 401
14.3 系统实现 403
14.3.1 代码组织 403
14.3.2 系统界面 405
14.3.3 代码设计 407
14.3.4 配置文件 446
14.3.5 图片文件 447
14.4 系统发布 448
14.4.1 运行环境 448
14.4.2 数据源配置 448
14.4.3 系统运行 449
14.4.4 系统部署 450
知识回顾 451
附录 A 程序编码规范及常见问题 453
A.1 命名规范 453
A.2 注释 454
A.2.1 类注释 455
A.2.2 方法注释 455
A.2.3 语句块注释 455
A.2.4 尾端注释 456
A.2.5 行末注释 456
A.2.6 文档注释 456
A.3 常见 Java 不规范代码 457
A.3.1 在 Eclipse 中格式化源代码并管理 import 语句 457
A.3.2 避免在方法中出现多个 return 语句（退出点） 457
A.3.3 简化 if-else 方法 458
A.3.4 在代码块周围使用大括号 458
A.3.5 把方法的参数声明为 final 类型 458
A.3.6 把多个 if 语句合并成一个 459
A.3.7 避免重复使用同样的字符串，创建一个常量吧 459
A.4 常见问题汇总 459
A.4.1 字符串连接误用 459
A.4.2 错误使用 StringBuffer 460
A.4.3 测试字符串相等性 460
A.4.4 数字转换成字符串 461
A.4.5 利用不可变对象(Immutable) 461
A.4.6 未指定字符编码 461
A.4.7 未对数据流进行缓存 461
A.4.8 不必要的初始化 462
A.4.9 用数组来描述一个结构 462

实例目录

【例 1-1】编程输出字符串：Hello world！ 14
【例 2-1】定义常量 PI，并令 PI=3.14 18
【例 2-2】定义两个 double 型的变量 d1，d2 19
【例 2-3】定义两个 dyte 型变量 x1、y1，并分别为其赋初值 20
【例 2-4】定义两个 short 型变量 x1 和 y1，并分别为其赋初值 20
【例 2-5】定义两个 int 型变量 x1 和 y1，并分别为其赋初值 20
【例 2-6】定义两个 long 型变量 x1 和 y1，并分别为其赋初值 21
【例 2-7】定义两个 float 型变量 x1 和 y1，并分别为其赋初值 21
【例 2-8】定义两个 double 型变量 x1 和 y1，并 ..... 分别为其赋初值 21
【例 2-9】定义两个 char 型变量 x1，y1 21
【例 2-10】定义两个逻辑型变量 b1，b2 22
【例 2-11】“++”和“--”运算符在程序中的使用 24
【例 2-12】关系运算符与逻辑运算符在程序中的使用 ..25
【例 2-13】无符号右移运算符和无符号右移赋值运算符的使用 27
【例 2-14】条件运算符的使用 28
【例 2-15】如果变量 a 的值大于 b 的值，则互换 a，b 的值，然后打印 a，b 的值 30
【例 2-16】根据考试分数打印提示信息 31
【例 2-17】用嵌套的 if-else 语句判断两个数字是大于、小于或等于关系，并输出结果 31
【例 2-18】用 switch 语句处理表达式中的运算符，并输出运算结果 33
【例 2-19】使用 while 语句计算 1 到 100 之间的整数之和 34
【例 2-20】使用 do-while 语句计算 1 到 100 之间的整数之和 35
【例 2-21】使用 for 语句计算 1 到 100 之间的能被 3 整除的数之和 36
【例 2-22】使用嵌套循环语句输出九九乘法表 37
【例 2-23】使用 break 语句跳出循环 38
【例 2-24】输出 1 到 10 之间所有不能被 3 整除的自然数 38
【例 2-25】采用两种不同方式分别声明整型一维数组 a1 和 b1 41
【例 2-26】数组的声明、初始化和其长度的测定 42
【例 2-27】数组的赋值示例 43
【例 2-28】利用一维数组输出 8 行杨辉三角形 44
【例 2-29】二维数组的建立与输出 45
【例 3-1】自定义类 Employee，创建并使用类 Employee 的两个对象 55
【例 3-2】自定义类 Employee，创建并使用类 Employee 的三个构造方法 58
【例 3-3】自定义类 SimpleValue 65
【例 3-4】自定义类 ReferenceValue 66
【例 3-5】命令行参数使用的实例 67
【例 3-6】静态方法访问成员变量的实例 71
【例 4-1】自定义父类 Teacher，创建其两个子类 JavaTeacher 和 DotNetTeacher 79
【例 4-2】自定义父类 Person，创建其子类 SubStudent 83
【例 4-3】自定义父类 Area，创建其携带不同参数及返回类型的同名方法 85
【例 4-4】子类中使用构造方法的实例 87
【例 4-5】抽象类的使用实例 93
【例 4-6】多态性的使用实例 95
【例 4-7】接口实现实例 102
【例 4-8】复杂的接口实现实例 103
【例 4-9】内部类和外部类之间的访问 105
【例 5-1】带参的 String 构造方法应用 112

【例 5-2】“+”运算符的应用.....113
【例 5-3】getChars()方法的应用.....114
【例 5-4】equals()和 equalsIgnoreCase()的应用.....115
【例 5-5】equals()与==的区别.....116
【例 5-6】 toString()方法的重载.....118
【例 5-7】泛型类测试.....123
【例 5-8】泛型方法的使用.....124
【例 5-9】通配类型参数测试.....125
【例 5-10】HashSet 类的应用.....129
【例 5-11】TreeSet 的用法.....131
【例 5-12】Comparable 接口的用法.....132
【例 5-13】比较器 Comparator 的用法.....133
【例 5-14】ArrayList 的使用.....135
【例 5-15】LinkedList 的使用.....136
【例 5-16】HashMap 的使用.....138
【例 5-17】TreeMap 的使用.....139
【例 5-18】Iterator 的使用.....141
【例 5-19】ListIterator 的使用.....142
【例 5-20】使用 Iterator 输出 Map 集合中的元素.....143
【例 5-21】Collections 的使用.....145
【例 5-22】Date 类的应用.....146
【例 5-23】Calendar 类中相关方法的使用.....147
【例 5-24】parse()方法的应用.....149
【例 5-25】Math 常用方法应用举例.....151
【例 5-26】arraycopy()方法的用法.....153
【例 5-27 】System 类中方法的应用.....154
【例 5-28】使用 Runtime 的相关方法对内存进行管理.....155
【例 5-29】exec()方法的使用.....157
【例 6-1】在程序中不捕获发生的异常.....159
【例 6-2】在程序中使用 try-catch 捕获异常.....164
【例 6-3】使用多个 catch 捕获可能产生的多个异常.....165
【例 6-4】多 catch 语句的应用.....165
【例 6-5】Exception 异常类的应用.....166
【例 6-6】同时捕获父类和子类异常时的顺序.....167
【例 6-7】使用 finally 语句进行善后处理.....168
【例 6-8】使用嵌套的 try 语句捕获程序中产生的所有异常.....169
【例 6-9】使用 throws 子句声明异常.....170
【例 6-10】throw 语句的使用.....171
【例 6-11】自定义异常的应用.....174
【例 7-1】顶层容器 JFramel 的应用.....179
【例 7-2】面板类 JPanel 的应用.....181
【例 7-3】JScrollPane 的应用.....183
【例 7-4】FlowLayout 布局管理器的使用.....184
【例 7-5】BorderLayout 布局管理器的使用.....186
【例 7-6】GridLayout 布局管理器的使用.....187
【例 7-7】CardLayout 布局管理器的使用.....189
【例 7-8】BoxLayout 布局管理器的使用.....190
【例 7-9】Box 容器的应用.....191
【例 7-10】Box 类的透明组件的使用.....192
【例 7-11】事件处理演示程序，内部类作为事件的监听者.....194
【例 7-12】使用组件所在类作为事件监听程序.....195
【例 7-13】使用匿名内部类作为事件监听程序.....196
【例 7-14】窗口事件的使用.....200
【例 7-15】键盘事件的使用.....202
【例 7-16】鼠标事件的响应.....204
【例 8-1】文本框 JTextField 的应用.....209
【例 8-2】命令按钮和文本域的使用.....212
【例 8-3】JCheckBox 的使用.....215
【例 8-4】JRadioButton 组件的使用.....216
【例 8-5】JComboBox 的使用.....219
【例 8-6】JList 的使用.....221
【例 8-7】JTable 的使用.....222
【例 8-8】创建系统菜单.....226
【例 8-9】弹出式菜单的创建.....228
【例 8-10】文件对话框的应用.....233
【例 9-1】File 类的应用.....239
【例 9-2】流的基本用法实例.....242
【例 9-3】ByteArrayInputStream 类的应用.....244
【例 9-4】DataInputStream 流的应用.....245
【例 9-5】ByteArrayOutputStream 类的应用.....247
【例 9-6】FileOutputStream 类的应用.....248
【例 9-7】DataOutputStream 类的应用.....249
【例 9-8】BufferedOutputStream 类的应用.....249
【例 9-9】CharArrayReader 类的应用.....252
【例 9-10】StringReader 类的应用.....253

【例 9-11】FileReader 类的应用.....253
【例 9-12】BufferedReader 类的应用.....254
【例 9-13】CharArrayWriter 类的应用.....256
【例 9-14】FileWriter 类的应用.....256
【例 9-15】InputStreamReader 类的应用.....258
【例 9-16】OutputStreamWriter 类的应用.....259
【例 9-17】随机访问文件应用实例.....260
【例 9-18】序列化应用实例.....263
【例 10-1】多线程应用实例.....268
【例 10-2】从 Thread 类派生创建线程.....270
【例 10-3】实现 Runable 接口创建多线程.....271
【例 10-4】线程启动实例.....273
【例 10-5】线程睡眠应用实例.....276
【例 10-6】线程让步应用实例.....277
【例 10-7】线程间协作应用实例.....278
【例 10-8】后台线程应用实例.....280
【例 10-9】多线程并发可能引发的问题.....282
【例 10-10】同步代码块应用实例.....284
【例 10-11】同步方法实例.....286
【例 10-12】线程间通信实例.....289
【例 10-13】产生死锁的实例.....291
【例 11-1】数据库连接类实例.....311
【例 11-2】创建数据表实例.....313
【例 11-3】向数据表中增加数据实例.....314
【例 11-4】修改数据表.....315
【例 11-5】删除记录.....316
【例 11-6】查询数据表信息.....317
【例 12-1】使用 URL 类获取远端主机上指定文件的内容.....327
【例 12-2】使用 URLConnection 类获取 Web 页面信息.....328
【例 12-3】使用 InetAddress 类远端主机的 IP 地址和主机名.....329
【例 12-4】使用 Socket 类获取指定连接的状态信息.....332
【例 12-5】使用 ServerSocket 类获取服务器的状态信息.....334
【例 12-6】模拟用户存话费和手机漫游的 C/S 结构应用系统.....335
【例 12-7】设计点对点的快速通信系统.....342
【例 12-8】设计多播系统.....348
【例 13-1】从客户信息表中查询客户名称以“联想”开头的客户信息.....362
【例 13-2】从客户信息表中查询客户名称中包含“联想”的客户信息.....362
【例 13-3】从客户信息表中查询客户名称第一个字符之后是“enovo”的客户信息.....362
【例 13-4】从客户信息表中查询客户所在地以"A"、"L"或"N"开头的客户信息.....362
【例 13-5】从客户信息表中查询客户所在地不以"A"开头的客户信息.....362

第 1 章　初识 Java 语言

Java 是一种安全的程序设计语言，它提供了诸多安全保障机制。Java 语言适用于 Internet 环境，是一种被广泛使用的网络编程语言。本章将对 Java 的发展、特点、开发运行环境，以及如何编译并执行 Java 程序等内容进行介绍。通过本章的学习，读者将会对 Java 有一个初步地了解，并能够顺利地搭建 Java 应用程序的运行开发环境。

1.1　Java 语言发展史及特点

本节主要介绍 Java 语言的发展历史和 Java 语言的特点。

1.1.1　Java 语言简介

Java 是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 开发运行平台的总称。下面我们来简单了解一下其发展历程。

1. Java 发展简史

Java 的历史要追溯到 1991 年，当时美国 Sun Microsystems 公司的 Patrick Naughton 及其伙伴 James Gosling 带领的工程师小组想要设计一种小型的计算机语言，主要应用对象是像有线电视转换盒这类消费设备。由于这些消费设备的处理能力和内存都很有限，所以语言必须非常小且能够生成非常紧凑的代码。另外，由于不同的设备生产商会选择不同的中央处理器（CPU），因此这种语言的关键是不能与任何特定的体系结构捆绑在一起。这个项目被命名为 Green。

项目开始时，项目组首先从改写 C/C++语言编译器着手，但是在改写过程中感到仅仅使用 C 语言无法满足需要，而 C++语言又过于复杂，安全性也差，无法满足项目设计的需要。于是项目组从 1991 年 6 月开始开发一种新的编程语言，并命名为 Oak，但后来发现 Oak 已是另一个公司的注册商标，于是又将其改名为 Java，并配了一杯冒着热气的咖啡图案作为它的标志。

由于要求 Java 这种语言必须非常小而且能够生成紧凑的代码，还要求该语言与平台无关。这些要求促使开发团队想起了很早以前的一种模型，某些 Pascal 的实现曾经在早期的 PC 上尝试过这种模型。以 Pascal 的发明者 Niklaus Wirth 为先驱，率先设计出一种为假想的机器（虚拟机）生成中间代码的可移植语言，这种中间代码可以应用于所有已经正确安装解释器的机器上。于是，Green 项目组的工程师也使用了虚拟机（Java 虚拟机），从而解决了课题中的主要问题（平台无关性）。

1992 年，Green 项目组发布了它的第一个产品，称之为“*7”。这个产品具有非常智能的远程控制。遗憾的是 Sun 公司对生产这个产品并不感兴趣，并且 Green 项目组的人员也没有找出其他的方法来将他们的技术推向市场。到了 1994 年 Green 项目组（这时换了一个新名字——First Person 公司）解散了。

在此期间，Internet 的万维网也日渐发展壮大，Web 的关键是把超文本页面转换到屏幕上的浏

览器，当时的浏览器主要是 Mosaic。

Java 语言的开发者设计并开发了一个功能更加强大的浏览器，该浏览器最终演变为 HotJava 浏览器。为了炫耀 Java 语言超强的能力，HotJava 浏览器采用 Java 编写，他们让 HotJava 浏览器具有执行网页中内嵌代码的能力。这一“技术印证”在 1995 年的 SunWorld 上得到了展示，同时引发了人们延续至今的对 Java 的狂热追逐。

1996 年初，Sun 发布了 Java 的第 1 个版本 Java 1.0，但 Java 1.0 不能用来进行真正的应用开发，后来的 Java 1.1 弥补了其中的大多部分明显的缺陷，大大改进了反射能力，并为 GUI 编程增加了新的事件处理模型。

1998 年，JavaOne 会议的头号新闻是即将发布 Java 1.2 版。这个版本取代了早期玩具式的 GUI，并且它的图形工具箱更加精细而且具有较强的可伸缩性，更加接近“一次编写，随处运行”的承诺。然后，Sun 公司将其名称改为更加吸引人的“Java 2 标准版软件开发工具箱 1.2 版”。

标准版的 1.3 和 1.4 版本对最初的 Java 2 版本做出了某些改进，扩展了标准类库，提高了系统性能。5.0 版是自 1.1 版以来第一个对 Java 语言做出重大改进的版本（这一版本原来被命名为 1.5 版，在 2004 年的 JavaOne 会议之后，版本数字升至 5.0）。这个版本添加了泛型类型（generic type），其挑战性在于添加这一特性并没有对虚拟机做出任何修改。

2005 年 6 月，JavaOne 大会召开，Sun 公司发布了 Java SE 6。此时，Java 的各种版本已经更名并取消其中的数字“2”，J2EE 更名为 Java EE，J2SE 更名为 Java SE，J2ME 更名为 Java ME。

2010 年 Sun 公司被 Oracle 公司收购。

目前最新版本是 2011 年甲骨文公司发布的 Java 7.0 正式版。

Oracle 公司计划在今年（2014 年）发布的 Java 8.0 正式版。

2. Java 语言的影响及应用前景

Java 语言是新一代面向对象的程序设计语言，特别适合于 Internet 应用程序的开发，它的硬件和软件平台的无关性直接威胁到 Windows 和 Intel 的垄断地位。用 Java 编程成为技术人员的一种时尚，并对未来软件的开发产生了重大影响。

Java 语言对软件开发技术的影响可以归纳为如下几个方面：

- 软件的需求分析。可将用户的需求进行动态的、可视化描述，以提供设计者更加直观的要求。而用户的需求是各种各样的，不受地区、行业、部门、爱好的影响，这些需求都可以用 Java 语言描述清楚。
- 软件的开发方法。由于 Java 语言是纯面向对象的程序设计语言，所以完全可以用面向对象的技术与方法来开发软件，这符合最新的软件开发规范要求。
- 动画效果。Java 语言的动画效果远比 GUI 技术逼真，尤其是利用 WWW 提供的巨大动画资源空间，可以共享全世界的动态画面资源。
- 软件最终产品。用 Java 语言开发的软件具有“可视化、可听化、可操作化、交互、动画、动作”等特点。
- Java 语言有着广泛的应用前景，大体上可以从以下几个方面来考虑其应用。
- 所有面向对象的应用开发，包括面向对象的事件描述、处理等。
- 计算过程的可视化、可操作化的软件开发。
- 动态画面的设计，包括图形图像的调用。
- 交互操作的设计（选择交互、定向交互、控制流程等）。

- Internet 的系统管理功能模块的设计，Web 页的动态设计、管理交互操作设计等。
- Intranet 上的软件开发（直接面向企业内部用户的软件）。
- 其他应用类型的程序（移动计算、嵌入式 Java 技术、实时 Java 等）。

1.1.2　Java 语言特点

Java 众多的突出特点使得它受到了大众的欢迎。归纳起来，Java 语言具有以下一些显著特点。

1. 简单性

人们希望构建一个无需深奥的专业训练就可以进行编程的系统，并且要符合当今的标准惯例。因此，尽管人们发现 C++不太适用，但在设计 Java 的时候还是尽可能地接近 C++，以便系统更易于理解。Java 剔除了 C++中许多很少使用、难以理解、易混淆的特性。例如，Java 中没有指针、结构和类型定义等概念，没有#include 和#define 等预处理器，也没有多重继承的机制。

简单的另一个方面是小。Java 的目标之一是支持开发能够在小型机器上独立运行的软件。基本的解释器以及类支持大约仅为 40KB；再加上基础的标准类库和对线程的支持（基本上是一个自包含的微内核）大约需要增加 175KB。在当时，这是一个了不起的成就。当然，由于不断的扩展，类库已经相当庞大了。现在有一个独立的具有较小类库的 Java 微型版（Java Micro Edition）用于嵌入式设备。

2. 面向对象性

Java 是一个纯粹的面向对象的语言，强调的是面向对象的特性，能够为软件工程技术提供很强的支持。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及动态的接口模型。与其他面向对象的语言一样，Java 具备继承、封装及多态性这些通常的特性，更提供了一些类的原型，程序员可以通过继承机制，实现代码的复用。另外，Java 的继承机制很独特，在设计时去掉了不安全的因素，因此使用 Java 可以编制出非常复杂但逻辑清晰的系统。

3. 分布式与安全性

Java 从诞生之日起就与网络联系在一起，由于它强调网络特性，使它成为一种分布式程序设计语言。Java 语言包括一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库，它提供一个 Java.net 包，通过它可以完成各种层次上的网络连接。因此 Java 语言编写的应用程序可以凭借 URL 打开并访问网络上的对象，其访问方式与访问本地文件系统几乎完全相同。Java 语言的另一个 Socket 类提供了可靠流式网络的连接，使程序设计者可以非常方便地创建分布式应用程序。

Java 程序在语言定义阶段、字节码检查阶段及程序执行阶段进行的三级代码安全检查机制，对参数类型匹配、对象访问权限、内存回收、Java 小应用程序的正确使用等都进行了严格的检查和控制，可以有效地防止非法代码的侵入，阻止对内存的越权访问，能够避免病毒的侵害。

4. 与平台无关性

如果基本数据类型设计依赖于具体的计算机和操作系统，会给程序的移植带来很大不便。Java 语言通过定义独立于软、硬件平台的基本数据类型及其相关运算，确保数据在任何硬件平台上保持一致。为了实现平台无关性，Java 语言规定了统一的基本数据类型。

Java 程序编译后生成二进制代码，即字节码（bytecode）。字节码就是虚拟机的机器指令，与平台无关。字节码有统一的格式，不依赖于具体的硬件环境。在任何安装 Java 运行时环境的系统

上，都可以执行这些代码。也就是说，只要安装了 Java 运行环境，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上运行。运行环境针对不同的处理器指令系统，把字节码转换为不同的具体指令，保证了程序能“到处运行”。

5. 解释和编译特性

Java 开发环境在 Java 源程序编译后生成一种称为字节代码（bytecode）的中间代码，字节代码非常类似于机器指令代码，但并不是二进制的机器指令代码，且字节代码并不专对一种特定的机器，所以 Java 程序不需重新编译便可在众多不同的计算机上执行，只要该机器上预先装有 Java 语言运行系统，这是其编译特性。Java 程序编译后产生字节代码，其运行要借助于 Java 解释器，Java 解释器直接对 Java 字节代码进行解释执行。以字节代码形式发布的 Java 程序运行在 JVM 环境上，JVM 将字节代码翻译成具体的 CPU 机器指令，因此 Java 解释器是与硬、软件平台有关的，在不同的平台上用不同的 JVM 实现（与平台相关部分的工作由 JVM 而不是 Java 编译器来完成，因为平台的种类比起应用软件的数量要少得多）。Java 解释器使 Java 程序在某一特定硬、软件平台环境中直接运行目标代码指令，这种连接程序通常比编译程序所需的资源少，所以程序员可以将更多的时间用在创建源程序上，而不必考虑运行环境，这是其解释特性。

6. 多线程

多线程机制使应用程序能够并行执行，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，这样就很容易实现网络上的实时交互行为和实时控制性能。

大多数高级语言（包括 C、C++等）都不支持多线程，用它们只能编写顺序执行的程序（除非有操作系统 API 的支持）。Java 内置了语言级多线程功能，提供现成的 Thread 类，只要继承这个类就可以编写多线程的程序，使用户程序并行执行。Java 提供的同步机制可保证各线程对共享数据的正确操作，完成各自的特定任务。在硬件条件允许的情况下，这些线程可以直接分布到各个 CPU 上，充分发挥硬件性能，减少用户等待的时间。

7. 动态执行

Java 执行代码是在运行时动态载入的，这种动态特性使它适合于一个不断发展的环境。在网络环境下，Java 语言编写的代码用于瘦客户机架构可减少维护工作。另外，类库中增加的新方法和其他实例不会影响到原有程序的执行，并且 Java 语言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可扩展性。

8. 自动废区回收性

在用 C 及 C++编写大型软件时，编程人员必须自己管理所用的内存块，这项工作非常困难，并往往成为出错和内存不足的根源。在 Java 环境下，编程人员不必为内存管理操心，Java 语言系统有一个叫做“无用单元收集器”的内置程序，它扫描内存，并自动释放那些不再使用的内存块。Java 语言的这种自动废区收集机制，对程序不再引用的对象自动取消其所占资源，彻底消除了出现存储器泄漏之类的错误，并免去了程序员管理存储器的繁琐工作。

9. 丰富的 API 文档和类库

Java 为用户提供了详尽的 API 文档说明。Java 开发工具包中的类库包罗万象、应有尽有，这使程序员的开发工作可以在一个较高的层次上展开。这也正是 Java 受欢迎的重要原因之一。

1.2　Java 开发环境的配置

要开发 Java 程序，必须要有一个开发环境，而一提到开发环境，读者可能首先想到的就是那些大名鼎鼎的集成开发工具，如 MyEclipse、JBuilder、JCreator Pro、Net Beans、Sun Java Studio、Microsoft Visual J++等。但实际上，在如此众多的开发工具中，除了 Microsoft Visual J++是使用自己的编译器外，其余的大多是使用 Sun 公司提供的免费的 JDK 作为编译器，只不过是开发了一个集成环境套在外面，方便程序员编程而已。

这些集成开发工具，虽然方便了程序员开发大型软件，但是它们封装了很多有关 JDK 的基本使用方法，在某些方面又过于复杂，并不太适合初学者使用。因此，本书将介绍最基本的 JDK 的安装和使用。

1.2.1　JDK 的安装

JDK（Java Development Kit）是 Sun 公司发布的免费的 Java 开发工具，它提供了调试及运行一个 Java 程序所有必需的工具和类库。在正式开发 Java 程序前，需要先安装 JDK，JDK 的最新版本可以到 http://www.oracle.com/technetwork/java/javase/downloads/index.html 上免费下载。由于 2010 年 Oracle 公司收购了 SUN 公司，所以下载的网址和以前有所区别。目前最新版本是 2011 年 7 月 28 日，甲骨文发布的 Java 7.0 正式版。根据运行时所对应的操作系统，JDK 7 可以划分为 for Windows、for Linux 和 for MacOS 等不同版本。

> **说明：** 本书实例基于的Java SE平台是JDK 7 for Windows。

下面就以 JDK 7 for Windows 为例，介绍它的安装和配置。

Step 1　到 http://www.oracle.com/technetwork/java/javase/downloads/index.html 下载 JDK 7 for Windows，文件名为 jdk-7u51-windows-i586-p.exe。然后，双击该文件，首先出现“欢迎”窗口，如图 1.1 所示。

图 1.1　“欢迎”窗口

Step 2　单击“下一步”按钮，进入如图 1.2 所示的“自定义安装”窗口。通过此窗口，可以选择要安装的模块和路径。