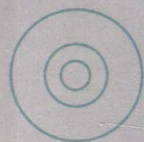


# Java SE 6.0

## 编程指南

吴亚峰 纪超 编著

作者有10年Java开发经验，有多年SUN公司认证授课经验  
全书包括713个典型实例，透彻解析Java SE的方方面面  
对Java SE 6.0的新功能进行了细致的剖析



CD-ROM

# Java SE 6.0

## 编程指南

吴亚峰 纪超 编著

人民邮电出版社  
北京

## 图书在版编目 (CIP) 数据

Java SE 6.0 编程指南 / 吴亚峰, 纪超编著. —北京: 人民邮电出版社, 2007.11  
ISBN 978-7-115-16740-8

I. J… II. ①吴…②纪… III. JAVA 语言—程序设计—指南 IV. TP312-62

中国版本图书馆 CIP 数据核字 (2007) 第 133315 号

## 内 容 提 要

Java 是目前所有编程语言中最流行、发展最快的语言之一, 随着 Java SE 6.0 的发布, Java 也变得更加强大和易用。本书结合丰富的程序实例, 为读者详细讲解了 Java 核心技术的各个方面, 囊括了 Java SE 6.0 中提供的各种新特性。

本书共分为 6 篇, 第 1 篇是起步篇, 详细介绍了学习 Java 语言所必备的一些知识; 第 2 篇为基础篇, 结合 Java 语言详细介绍了面向对象开发的各个方面; 第 3 篇为高级基础篇, 该篇为读者介绍了 Java 核心技术中一些比较高级的知识, 例如异常处理、集合框架、多线程以及内存管理等; 第 4 篇为 Swing GUI 篇, 该篇结合丰富的实例对 Swing 中的各个功能组件进行了详细的介绍; 第 5 篇为图形图像篇, 该篇主要为读者介绍了 Java 中有关图形绘制与图像处理的知识; 第 6 篇为高级特性篇, 该篇为读者详细介绍了 Java 中一些比较实用的高级特性, 例如数据库、流、反射、脚本开发等。

从初学者到具有一定开发经验的开发人员都适合阅读此书。

## Java SE 6.0 编程指南

- 
- ◆ 编 著 吴亚峰 纪 超  
责任编辑 屈艳莲
  - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
北京隆昌伟业印刷有限公司印刷  
新华书店总店北京发行所经销
  - ◆ 开本: 787×1092 1/16  
印张: 56.5  
字数: 1 435 千字 2007 年 11 月第 1 版  
印数: 1-5 000 册 2007 年 11 月北京第 1 次印刷

---

ISBN 978-7-115-16740-8/TP

定价: 98.00 元 (附光盘)

读者服务热线: (010)67132692 印装质量热线: (010)67129223

# 本书编委会

主编：吴亚峰 纪超 项宇峰

编委：尚兴国 唐敏 唐智皞 王安平 王成喜

王淑敏 谢马远 张丹 张迪妮 钟蜀明

竺东 祝庆林 陈水峰 慈元龙 关蔼婷

贺宇 胡立实 姜磊 李来春 李争亭

梁贵 刘吉万 柳玲 罗栋 罗玉霞

秦辉 尚敬宜 尚世博 王石磊 王新平

文奇 吴琪 席国庆 谢超文 臧勇

张国强 张家春 郭玉敏 贺道权 江成海

# 前 言

从 Sun Microsystems 公司推出 Java 语言开始, 由于其跨平台、开放等优良的特性吸引了众多软件公司与开发人员的目光。自 1995 年以来, 先后有很多开发商都选择了 Java, 例如 IBM、BEA 以及 Oracle 等这些开发商的投入, 使得 Java 的生命力更加强大。

时至今日, 随着 Java SE 6.0 的推出, 一个更加强大、更加易用的开发平台展现在开发人员面前。

为了帮助众多的开发人员与学习者熟练地掌握 Java 语言的核心知识, 我们根据多年的项目开发经验以及心得体会编写了本书。

本书不仅为读者介绍了 Java 中的很多核心技术, 还对 Java SE 6.0 提供的大部分新特性进行了详尽的介绍, 下面列出了本书中介绍的一些关于 Java SE 5.0 以及 Java SE 6.0 的新特性:

- Desktop 类的知识与应用;
- 系统托盘;
- Derby 数据库的应用;
- 安全类型枚举;
- 嵌入式脚本开发。

## 本书的特点

本书不仅讲述了 Java 的核心技术以及编程思想, 同时详细地介绍了 Java SE 6.0 中带来的各种新特性, 并且对每一个知识点都使用具体的案例进行了说明。这样使得读者在学习的时候可以更加深入地理解 Java 语言的核心思想与具体应用, 例如在介绍数据库开发的章节, 专门提供了一个完整的、有实用价值的数据库观察器案例。

本书在内容的编排以及目录的组织上考虑周到, 力求读者能够循序渐进地掌握 Java 语言核心技术的相关知识, 对每一个示例程序, 都强调了运行的结果以及从中可以看出的问题, 以保证读者在实践的过程中不断地提高开发水平。此外本书还有以下特点。

- 讲解通俗, 步骤详细: 在介绍某些技术的时候, 读者只需按照步骤进行操作, 便能体会到开发程序的乐趣。
- 实例经典, 内容丰富: 在本书中, 每个知识点都配有相应的实例, 这样将会使读者更容易理解书中所讲述的知识, 加快知识理解与消化的过程。同时, 对于一些比较大的知识点, 书中还给出了很多具有实际价值的完整案例。例如, 数据库章的数据库观察器, I/O 流章的图片观察器等。
- 主次分明, 结构清晰: 本书在讲解实例的过程中, 基本的思路是贯穿始终的, 使读者能够更加轻松地阅读, 快速地理解。
- 配有光盘, 辅助教学: 本书配有多媒体光盘, 书中案例的相关文件和源代码按照章节与

案例名称分别存放，读者只需要将案例对照书中的讲解进行学习，便可以很快地掌握知识点。

### 本书面向的读者

---

本书深入浅出地介绍了 Java 语言核心技术的各个方面，内容详尽、覆盖面广，适合所有不同层次的 Java 开发人员与学习者进行阅读。

- **Java 初学者：**本书在讲解知识时，起点低，初学者能够非常容易上手学习。并且书中章节的顺序安排合理，由浅入深，初学者只要按照本书安排的章节进行学习即可循序渐进地掌握核心 Java 各个方面的知识。对于该类读者，本书无疑是一本从入门到精通的优秀教材。
- **对 Java 有一定基础的开发人员：**本书包含了我们多年从事 Java 开发与培训工作的经验，通过这些经验以及心得体会，总结出了很多比较容易被误解的知识点。对于有一定经验的开发人员，通过本书还可以更好地解决学习 Java 时遇到的难题，是快速提高 Java 水平的好帮手。
- **对高级开发人员：**即使读者是一名已经使用 Java 进行程序开发的高手，仍然能够从本书中学习到很多有用的知识。例如，本书中详细介绍了很多 Java SE 6.0 中的新特性，以及一些重要的编程思想、设计模式。

### 关于作者

---

**吴亚峰：**毕业于北京邮电大学，后留学澳大利亚卧龙岗大学取得硕士学位，1998 年开始从事 Java 开发，有多年的 Java 开发与培训经验，主要研究的方向为 Java EE。现在为 Java EE 独立软件工程师，同时兼任 Sun 授权 Java 培训中心认证教师，为数十家著名企业培训了上千名开发人员，曾编写过《精通 NetBeans——Java 桌面、Web 与企业级程序开发详解》一书。

**纪超：**毕业于河北理工大学，从进入大学开始就痴迷于编程，大学二年级的时候就通过了 SCJP 认证，大学四年级时还获得了第二届 Java Cup 全国大学生信息技术大奖赛三等奖，多年来一直致力于使用 Java 开发程序。目前在某著名银行业解决方案提供商任高级开发人员，主要研究方向为业务层与 Web 层的开发。

### 致谢

---

本书在编写过程中得到了唐山百纳科技有限公司 Java 培训中心的大力支持，同时苏亚光、王海峰、段洁男以及作者的家人为本书的编写提供了很多帮助，在此表示衷心感谢！

编者  
2007 年 9 月

# 目 录

## 第一篇 起步篇

### 第 1 章 初识 Java —— 3

#### 1.1 Java 简介 —— 3

##### 1.1.1 Java 的不同平台 —— 3

##### 1.1.2 Java 发展的历程 —— 3

##### 1.1.3 Java 的特点 —— 4

#### 1.2 安装开发工具包 —— 5

##### 1.2.1 下载 JDK —— 5

##### 1.2.2 安装 JDK —— 6

##### 1.2.3 安装后 Java 目录的解读 —— 7

#### 1.3 学会使用 API —— 7

#### 1.4 第一个 Java 程序 —— 8

##### 1.4.1 开发源代码 —— 8

##### 1.4.2 编译运行 —— 9

#### 1.5 小结 —— 11

### 第 2 章 基本数据类型——构建 Java 大厦的基础 —— 12

#### 2.1 源代码注释 —— 12

##### 2.1.1 单行注释 —— 12

##### 2.1.2 区域注释 —— 12

##### 2.1.3 文档注释 —— 13

#### 2.2 基本数据类型 —— 14

##### 2.2.1 整型 —— 15

##### 2.2.2 浮点型 —— 17

##### 2.2.3 char 型 —— 17

##### 2.2.4 boolean 型 —— 18

#### 2.3 基本数据类型值间的转换 —— 18

##### 2.3.1 自动转换 —— 18

##### 2.3.2 手动强制转换 —— 19

##### 2.3.3 隐含强制转换 —— 19

2.4 标识符命名规范 —— 20

2.4.1 正确的命名标识符 —— 20

2.4.2 提倡的命名习惯 —— 21

2.5 小结 —— 21

第3章 表达式——描述行为的元素 —— 22

3.1 不简单的算术运算符 —— 22

3.1.1 “+”运算符 —— 22

3.1.2 “-”运算符 —— 24

3.1.3 “\*”运算符 —— 25

3.1.4 “/”运算符 —— 25

3.1.5 “%”运算符 —— 26

3.2 自增自减运算 —— 27

3.3 关系运算 —— 28

3.3.1 等于/不等于运算 —— 28

3.3.2 比较大小运算 —— 29

3.4 逻辑运算 —— 30

3.4.1 “与”运算 —— 30

3.4.2 “或”运算 —— 31

3.4.3 “非”运算 —— 32

3.5 三元运算符 —— 32

3.6 位运算 —— 33

3.7 移位运算 —— 34

3.7.1 “<<”左移 —— 35

3.7.2 “>>”右移 —— 35

3.7.3 “>>>”无符号右移 —— 36

3.7.4 令人困扰的例子 —— 37

3.8 赋值运算 —— 37

3.8.1 普通赋值运算 —— 37

3.8.2 运算赋值运算 —— 38

3.9 括号及运算符间的优先级关系 —— 38

3.10 常用数学工具包——java.lang.Math 类 —— 39

3.10.1 数学常量 —— 39

3.10.2 常用数学函数 —— 40

3.11 小结 —— 41

第4章 流程控制——Java 世界的航行舵手 —— 42

4.1 if 条件语句 —— 42



4.1.1	简略形式	42
4.1.2	完全形式	43
4.1.3	语句的嵌套	43
4.2	switch 多分支语句	45
4.2.1	基本语法	45
4.2.2	合法的判断表达式	46
4.2.3	合法的 case 表达式	47
4.2.4	详细执行流程	49
4.3	while 循环语句	50
4.4	do-while 循环语句	52
4.5	for 循环语句	53
4.5.1	基本语法	53
4.5.2	声明的三大组成部分	54
4.5.3	复杂的 for 循环案例	55
4.5.4	用 for 实现其他循环	55
4.6	break 中断语句	56
4.7	continue 继续语句	57
4.8	小结	58

## 第 5 章 数组——以不变应万变的哲学 59

5.1	数组的声明及创建	59
5.1.1	声明数组引用	59
5.1.2	创建数组对象	60
5.2	Java 中数组的实现机制	61
5.3	数组的初始化	63
5.3.1	默认初始化	63
5.3.2	利用循环初始化	64
5.3.3	枚举初始化	66
5.4	数组的相互赋值	67
5.4.1	基本类型数组赋值规则	67
5.4.2	引用型数组赋值规则	68
5.5	数组的常用操作	69
5.5.1	数组复制	69
5.5.2	数组排序	71
5.5.3	搜索指定元素	72
5.5.4	比较数组中的元素	73
5.6	关于 args[]	73
5.7	小结	74

## 第二篇 基础篇

### 第 6 章 对象和类——Java 世界的细胞 —— 77

- 6.1 面向对象概述 —— 77
  - 6.1.1 面向对象程序设计思想的诞生 —— 77
  - 6.1.2 面向过程与面向对象思想的对比 —— 78
  - 6.1.3 面向对象技术的背景和特点 —— 79
- 6.2 类的定义与对象的创建 —— 80
- 6.3 成员变量 —— 81
  - 6.3.1 成员变量的开发与使用 —— 81
  - 6.3.2 成员变量的初始值 —— 82
  - 6.3.3 对象引用变量的比较 —— 84
- 6.4 方法 —— 85
- 6.5 变长参数 —— 86
- 6.6 引用问题 —— 87
  - 6.6.1 调用不存在的方法或成员变量 —— 87
  - 6.6.2 用空引用进行调用 —— 88
  - 6.6.3 数组的空引用问题 —— 89
- 6.7 局部变量 —— 89
  - 6.7.1 局部变量的作用域 —— 89
  - 6.7.2 局部变量的初始化 —— 90
- 6.8 this 预定义对象引用 —— 92
- 6.9 擅用系统已有的类 —— 94
  - 6.9.1 Java 中的 Date 类 —— 94
  - 6.9.2 Java 中的 GregorianCalendar 类 —— 96
  - 6.9.3 擅用系统已有类的思想 —— 98
- 6.10 小结 —— 99

### 第 7 章 访问控制——Java 世界的卫兵 —— 100

- 7.1 包的使用 —— 100
  - 7.1.1 声明创建包 —— 100
  - 7.1.2 引入包内的资源 —— 102
  - 7.1.3 静态引入 —— 104
- 7.2 类的访问控制 —— 105
  - 7.2.1 公有访问级别 —— 105
  - 7.2.2 默认访问级别 —— 106
  - 7.2.3 类与源代码文件的搭配 —— 106
- 7.3 成员的访问控制 —— 107
  - 7.3.1 公共类型 —— 107

7.3.2	私有类型	108
7.3.3	默认类型	109
7.3.4	保护类型	109
7.3.5	Java 中封装的实现	110
7.4	final 的变量	112
7.4.1	final 的成员变量	113
7.4.2	final 的局部变量	115
7.5	static 关键字的使用	116
7.5.1	静态成员	116
7.5.2	静态成员的访问	117
7.5.3	静态最终成员变量	119
7.6	小结	121
第 8 章	继承——多态的支柱	122
8.1	继承概述	122
8.1.1	类之间的关系	122
8.1.2	面向对象中的继承性	124
8.2	类的继承	125
8.3	成员变量的继承与隐藏	126
8.3.1	成员变量的继承规则	126
8.3.2	成员变量的隐藏	129
8.4	对象引用的使用	130
8.4.1	对象引用能指向的对象类型	130
8.4.2	对象引用的强制类型转换	131
8.4.3	对象引用所能调用的成员	132
8.4.4	对象引用的赋值与比较	133
8.5	方法的继承与重写	135
8.5.1	方法的继承规则	135
8.5.2	方法重写的基本知识	136
8.5.3	构成重写的条件	137
8.5.4	返回类型的规则	138
8.5.5	访问级别的要求	140
8.5.6	重写基于继承	141
8.5.7	静态方法没有重写	141
8.5.8	通过重写扩展父类方法的功能	143
8.5.9	替代性原理	144
8.6	方法的重载	145
8.6.1	方法重载的规则	145
8.6.2	重载方法的匹配	145

- 8.6.3 重写与重载的区别 ——— 149
- 8.7 final 与继承 ——— 149
  - 8.7.1 最终的类 ——— 149
  - 8.7.2 最终的方法 ——— 150
- 8.8 abstract 与继承 ——— 151
  - 8.8.1 抽象的类 ——— 151
  - 8.8.2 抽象的方法 ——— 152
- 8.9 基于继承的多态 ——— 154
- 8.10 小结 ——— 155

**第 9 章 接口——灵活性的基石 ——— 156**

- 9.1 概述及其特性 ——— 156
- 9.2 成员变量在接口中的使用 ——— 157
  - 9.2.1 语法规则 ——— 157
  - 9.2.2 接口中成员变量的作用 ——— 158
- 9.3 方法在接口中的使用 ——— 159
  - 9.3.1 语法规则 ——— 159
  - 9.3.2 如何实现接口 ——— 160
  - 9.3.3 接口引用的使用 ——— 162
  - 9.3.4 接口中方法无法使用的修饰符 ——— 165
- 9.4 接口与抽象类 ——— 166
  - 9.4.1 语法上的不同 ——— 167
  - 9.4.2 具体含义的不同 ——— 167
- 9.5 基于接口的多态 ——— 169
- 9.6 接口与回调 ——— 170
- 9.7 instanceof 的使用 ——— 172
  - 9.7.1 基本语法与使用 ——— 172
  - 9.7.2 不允许进行测试的情况 ——— 174
- 9.8 小结 ——— 175

**第 10 章 构造器——对象制造的工厂 ——— 176**

- 10.1 基础知识 ——— 176
  - 10.1.1 编写构造器的语法规则 ——— 176
  - 10.1.2 访问限制修饰符与构造器 ——— 176
  - 10.1.3 构造器与返回类型 ——— 179
- 10.2 创建对象 ——— 180
- 10.3 重载构造器 ——— 181
- 10.4 级联调用的构造器 ——— 182
  - 10.4.1 构造器的调用流程及默认构造器 ——— 182

- 10.4.2 自定义构造器需要注意的问题 —— 185
- 10.4.3 不能继承构造器 —— 186
- 10.4.4 调用兄弟构造器 —— 187
- 10.5 单列模式 —— 189
- 10.6 Java 程序的加载过程 —— 190
- 10.7 小结 —— 192

### 第三篇 高级基础篇

#### 第 11 章 异常处理——Java 世界的医生 —— 195

- 11.1 异常处理的基本知识 —— 195
  - 11.1.1 try 和 catch 捕获异常 —— 195
  - 11.1.2 异常的传播过程 —— 198
  - 11.1.3 finally 语句块的使用 —— 199
  - 11.1.4 try、catch 及 finally 语句块之间需要注意的问题 —— 201
- 11.2 异常的层次结构 —— 203
  - 11.2.1 捕获异常 —— 203
  - 11.2.2 未捕获异常 —— 205
- 11.3 再次抛出异常 —— 206
  - 11.3.1 什么是异常的再抛出 —— 206
  - 11.3.2 显性再抛出 —— 207
  - 11.3.3 隐性再抛出 —— 209
  - 11.3.4 方法重写对抛出异常声明的约束 —— 210
- 11.4 定义自己的异常 —— 212
  - 11.4.1 创建自己的异常类 —— 212
  - 11.4.2 使用自定义的异常类 —— 213
  - 11.4.3 显性再抛出作用的体现 —— 215
- 11.5 异常的匹配 —— 217
  - 11.5.1 同时捕获多种异常 —— 217
  - 11.5.2 多个 catch 语句的先后顺序 —— 218
- 11.6 断言 —— 219
  - 11.6.1 什么是断言 —— 219
  - 11.6.2 如何启用/关闭断言 —— 220
  - 11.6.3 防止滥用断言 —— 222
- 11.7 小结 —— 222

#### 第 12 章 封装类——鸿沟之上的桥梁 —— 223

- 12.1 封装类的基本知识 —— 223
  - 12.1.1 封装类概述 —— 223
  - 12.1.2 创建封装类对象 —— 223

12.1.3	封装类对象的其他知识	225
12.2	数据转换功能	226
12.2.1	基本数据类型值转换为字符串	226
12.2.2	字符串转换为基本数据类型值	229
12.3	其他常用方法	231
12.3.1	静态工厂方法	231
12.3.2	isNaN 方法	232
12.3.3	equals 方法	233
12.4	自动打包/解包	235
12.4.1	自动打包	235
12.4.2	自动解包	236
12.5	特殊的数值计算	237
12.5.1	特大整数的计算	237
12.5.2	浮点数的精确计算	239
12.6	小结	242
<b>第 13 章 字符串——优异的内存组织机制</b> 243		
13.1	String 类的基础知识	243
13.1.1	对象的创建	243
13.1.2	巧用构造器	244
13.1.3	String 类的重要方法	245
13.2	String 对象的内存机制	248
13.2.1	一段令人困惑的字符串程序	248
13.2.2	“一次投入，终身回报”的 String 内存机制	249
13.2.3	String 对象特殊机制付出的代价	252
13.3	StringBuffer 类	253
13.3.1	弥补 String 不足的 StringBuffer 类	253
13.3.2	编写方法链以及 StringBuffer 类的重要方法	255
13.4	StringBuilder 类	258
13.5	正则表达式	259
13.5.1	正则表达式的基本语法	259
13.5.2	Pattern 类简介	262
13.5.3	Matcher 类简介	263
13.5.4	Pattern 与 Matcher 类的综合应用	264
13.6	String 类中正则式的应用	266
13.6.1	模式匹配检查	266
13.6.2	利用正则式进行查找替换	267
13.6.3	利用正则式对字符串进行分析	268
13.7	小结	269

<b>第 14 章 集合框架——强大的对象管理器</b>	<b>270</b>
14.1 Object 类——所有类的超类	270
14.1.1 toString 方法的重写	270
14.1.2 equals 方法的意义	271
14.1.3 hashCode 方法的意义	272
14.2 重写 equals 与 hashCode 方法	273
14.2.1 重写 equals 方法	273
14.2.2 重写 hashCode 方法	275
14.3 集合框架的层次结构	277
14.4 Ordered 与 Sorted 的接口	278
14.4.1 Ordered 的排序	278
14.4.2 Sorted 的排序	279
14.5 列表	279
14.5.1 列表接口——List	279
14.5.2 列表的数组实现	281
14.5.3 历史悠久的向量	282
14.5.4 列表的链接实现	284
14.5.5 依赖性倒置原理	285
14.5.6 将数组转换为列表	285
14.6 集合	286
14.6.1 Set 接口及含义	286
14.6.2 HashSet 类的使用	287
14.6.3 equals 与 hashCode 方法重写规定的作用	288
14.6.4 LinkedHashSet 类的使用	291
14.6.5 SortedSet 接口与 TreeSet 类	292
14.6.6 自定义满足 Sorted 集合的类	293
14.6.7 定制 SortedSet 的排序规则	296
14.6.8 集合的遍历	298
14.6.9 使用 for-each 循环遍历集合	300
14.7 映射集	301
14.7.1 Map 接口及含义	301
14.7.2 HashMap 类的使用	302
14.7.3 Hashtable 类的使用	303
14.7.4 LinkedHashMap 类的使用	304
14.7.5 SortedMap 接口与 TreeMap 类	305
14.7.6 映射的遍历	308
14.8 栈在 Java 中的实现	309
14.8.1 Stack 类的使用	309

- 14.8.2 Deque 接口的使用 ——— 310
- 14.8.3 利用栈计算数学表达式 ——— 311
- 14.9 集合元素的常用操作 ——— 314
  - 14.9.1 元素排序 ——— 315
  - 14.9.2 搜索特定元素 ——— 316
  - 14.9.3 任意打乱元素顺序 ——— 317
  - 14.9.4 其他的简单操作 ——— 318
- 14.10 小结 ——— 320
  
- 第 15 章 内部类——Java 世界的多面手 ——— 321**
  - 15.1 非静态内部类 ——— 321
    - 15.1.1 语法规则 ——— 321
    - 15.1.2 外部类之内创建内部类对象 ——— 322
    - 15.1.3 外部类之外创建内部类对象 ——— 323
    - 15.1.4 内部类与外部类之间的成员互访 ——— 324
    - 15.1.5 内部类与外部类的预定义对象引用 this ——— 327
  - 15.2 局部内部类 ——— 328
    - 15.2.1 局部内部类的定义及创建 ——— 328
    - 15.2.2 局部变量与局部内部类 ——— 329
    - 15.2.3 静态方法中的局部内部类 ——— 331
  - 15.3 静态内部类 ——— 332
    - 15.3.1 语法规则 ——— 332
    - 15.3.2 创建静态内部类的对象 ——— 332
    - 15.3.3 静态/非静态内部类的区别 ——— 333
  - 15.4 匿名内部类 ——— 334
    - 15.4.1 基于继承的匿名内部类 ——— 334
    - 15.4.2 基于接口实现的匿名内部类 ——— 335
    - 15.4.3 匿名内部类的初始化 ——— 337
    - 15.4.4 匿名内部类作用的体现 ——— 337
  - 15.5 理解内部类 ——— 339
  - 15.6 内部接口 ——— 340
    - 15.6.1 定义在类中的内部接口 ——— 340
    - 15.6.2 定义在接口中的内部接口 ——— 341
  - 15.7 小结 ——— 342
  
- 第 16 章 多线程——Java 中的并发协作 ——— 343**
  - 16.1 线程的基本知识 ——— 343
    - 16.1.1 多线程编程的意义 ——— 343
    - 16.1.2 定义自己的线程 ——— 344



- 16.1.3 创建线程对象 ——— 345
- 16.1.4 启动线程 ——— 347
- 16.1.5 同时使用多个线程 ——— 348
- 16.2 线程的状态 ——— 350
- 16.3 线程的调度 ——— 351
  - 16.3.1 睡眠 ——— 351
  - 16.3.2 线程的优先级 ——— 353
  - 16.3.3 线程的让步 ——— 355
  - 16.3.4 守护线程 ——— 357
- 16.4 线程的同步 ——— 359
  - 16.4.1 同步方法简介 ——— 359
  - 16.4.2 简单使用同步方法 ——— 360
  - 16.4.3 线程同步调度的方法 ——— 362
  - 16.4.4 “生产者—消费者”案例的框架 ——— 362
  - 16.4.5 “生产者—消费者”案例的实际运行 ——— 365
  - 16.4.6 notify 方法的使用 ——— 366
  - 16.4.7 同步的语句块 ——— 367
  - 16.4.8 线程的死锁 ——— 369
  - 16.4.9 防止错误的使用 wait、notify、notifyAll 方法 ——— 371
- 16.5 获取当前正在运行的线程 ——— 372
- 16.6 volatile 关键字的含义与使用 ——— 372
- 16.7 小结 ——— 373

## 第 17 章 高级线程开发 ——— 374

- 17.1 线程池的使用 ——— 374
  - 17.1.1 线程池的基本思想 ——— 374
  - 17.1.2 JavaSE 5.0 中固定尺寸线程池的基本知识 ——— 374
  - 17.1.3 自定义尺寸固定线程池的使用 ——— 375
  - 17.1.4 单任务线程池的使用 ——— 377
  - 17.1.5 可变尺寸线程池的使用 ——— 378
  - 17.1.6 延迟线程池的使用 ——— 380
  - 17.1.7 使用自定义参数的线程池 ——— 381
- 17.2 有返回值的线程调用 ——— 384
  - 17.2.1 Callable 接口简介 ——— 384
  - 17.2.2 Future 接口简介 ——— 384
  - 17.2.3 Callable 与 Future 接口的具体使用 ——— 385
- 17.3 资源的封锁 ——— 386
  - 17.3.1 Lock 接口与 ReentrantLock 类简介 ——— 386
  - 17.3.2 ReentrantLock 锁的具体使用 ——— 387