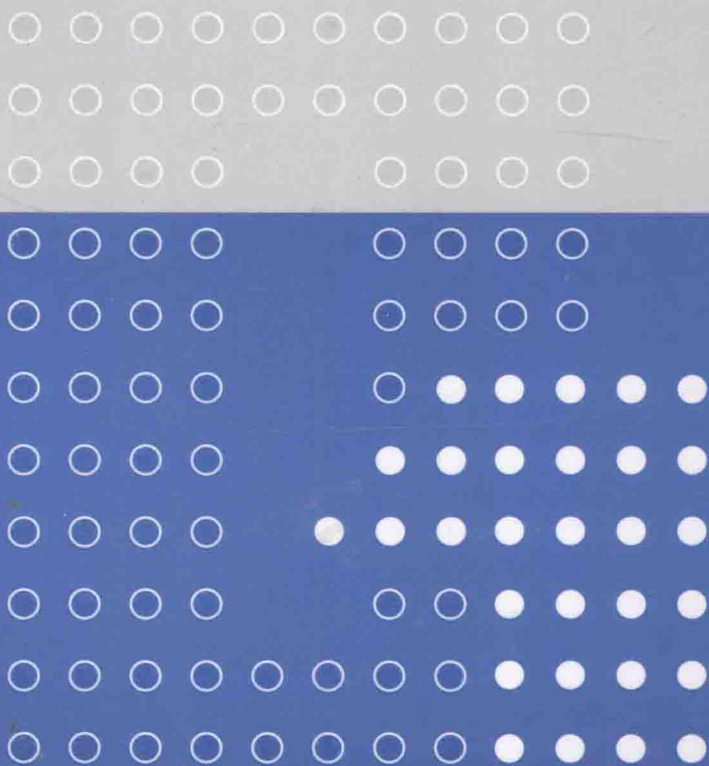




普通高等教育“十一五”国家级规划教材 计算机系列教材

Java语言 与面向对象程序设计



张利锋 孙丽 杨晓玲 编著

清华大学出版社



计算机系列教材

张利锋 孙丽 杨晓玲 编著

Java语言 与面向对象程序设计

清华大学出版社

北京

内 容 简 介

本书全面介绍 Java 的面向对象编程特性,将编程原则和最佳实践融入各个知识点,并通过大量的有应用背景的实例展示面向对象编程的特点和 Java 的实现机制。

本书共 14 章,涵盖 Java 语言的主要知识点,第 1~7 章为 Java 面向对象的编程基础,内容包括类的设计、对象的使用、正则表达式、继承和多态、接口机制、异常处理、注解与反射机制、Lambda 表达式等;第 8~14 章则涵盖 Java 编程的高级主题,内容包括容器类数据结构、用户界面设计、输入输出流(I/O)、新 I/O 框架、多线程编程、线程池和新的线程控制机制、Socket 编程与 Mina 框架以及 JDBC 等;第 13 章通过一个在线游戏项目演示了项目开发的步骤和知识点融合的过程。

本书既可以作为计算机、软件工程、网络工程、信息工程、电子工程等本科专业面向对象程序设计课程和 Java 语言程序设计的教材,也可以作为软件开发人员的参考书。

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 语言与面向对象程序设计/张利锋,孙丽,杨晓玲编著. —北京:清华大学出版社,2015

计算机系列教材

ISBN 978-7-302-40054-7

I. ①J… II. ①张… ②孙… ③杨… 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(2015)第 082494 号

责任编辑:白立军

封面设计:常雪影

责任校对:白 蕾

责任印制:杨 艳

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:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:28.75 字 数:663 千字

版 次:2015 年 6 月第 1 版 印 次:2015 年 6 月第 1 次印刷

印 数:1~2000

定 价:49.00 元

产品编号:062972-01

Java 是一门简单的、动态的跨平台编程语言,它是面向对象编程的典范,因此更适合面向对象程序设计方法论与实践的入门教学。Java 被广泛地应用到各个领域:高伸缩高可靠的服务器端应用开发、复杂的企业生产运营和业务管理系统、移动应用开发和分布式计算、嵌入式应用等。本书不仅全面深入地讲解 Java 语言的语法和编程机制,更注重编程实践和编程思维的训练,通过大量的有应用背景的实例展示 Java 在业务处理中的实现特点和方法。

面向对象的观点是软件构造的主要范型,但很多初学者习惯于使用面向过程编程的观点和编程思路,因此,对他们而言,掌握面向对象的设计方法具有很大的挑战性。本书基于多年的 Java 教学经验和实例,力求组织的知识点反映 Java 语言的主要特色,并将设计模式和最佳编程实践融入教学环节,以便降低面向对象编程的门槛和学习成本,快速提高程序设计的能力。

本书适应面向对象技术和 Java 开发技术的最新发展,注重计算思维能力的培养,结合大量案例参悟问题解决之道。本书的特色包括以下 8 个方面。

(1) 贯彻面向对象的编程思想,不仅重视技术的讲解,更强调程序设计原则、理念和最佳实践的理解和掌握。

(2) 将设计模式思想融合到知识点和案例教学中,比如图形用户界面框架 Swing 中的模型视图控制器架构、事件处理的观察者模式、单例模式和 I/O 流中装饰模式等。

(3) 体现 Java 语言的最新发展,将 Lambda 表达式、注解、反射、异步 I/O 框架(NIO)、新并发机制和线程池等内容引入教材。

(4) 将编程风格的训练作为程序设计能力的重要部分,并将其贯穿全书的示例代码。

(5) 将流行开源框架(Mina/Netty, Spring 等)的设计理念和机制融入知识点,开阔学习者视野,深入理解其应用场景和效果。

(6) 明确知识点的学习动机和背景,将项目分解到各个知识点的示例中,有助于知识点之间的融会贯通。

(7) 提供完整案例教学,设计思路贴近项目开发实战,详细介绍分析设计过程并通过 UML 表达,有助于提高学习者的开发能力。

(8) 丰富的习题,辅助巩固每章的知识点,力求快速提高学习者的分析和编程能力。

第 1 章 Java 语言概述	/1
1.1 开发 Java 的原因	/1
1.2 Java 语言的特点	/1
1.3 Hello World	/2
1.4 执行 Java 程序	/3
1.5 Java 开发环境	/5
1.6 Java 与 Android	/9
1.7 本章小结	/9
1.8 本章习题	/10
第 2 章 Java 编程基础	/11
2.1 标识符	/11
2.2 变量	/12
2.3 基本数据类型	/12
2.3.1 布尔型	/12
2.3.2 整型	/13
2.3.3 浮点型	/14
2.3.4 字符型	/14
2.3.5 基本数据类型的转换	/15
2.3.6 常量	/16
2.4 运算符和表达式	/16
2.4.1 算术运算符	/17
2.4.2 关系运算符	/18
2.4.3 逻辑运算符	/19
2.4.4 赋值运算符	/20
2.4.5 位运算符	/20
2.4.6 表达式	/23
2.4.7 运算符优先级	/23
2.5 语句	/24
2.5.1 if 语句	/25

	2.5.2	switch 语句	/25
	2.5.3	for 语句	/26
	2.5.4	while 和 do-while 语句	/27
	2.5.5	break 和 continue 语句	/27
	2.5.6	return 语句	/29
	2.6	程序设计风格	/29
	2.7	程序错误与调试	/30
	2.8	本章小结	/31
	2.9	本章习题	/31
	第 3 章 封装与类 /32		
	3.1	类	/33
	3.1.1	成员变量和局部变量	/33
	3.1.2	成员修饰符	/34
	3.2	方法重载	/35
	3.3	对象	/37
	3.3.1	构造方法	/38
	3.3.2	创建对象	/39
	3.3.3	使用对象	/41
	3.3.4	对象参数传值	/42
	3.3.5	垃圾回收机制	/43
	3.4	实例变量和类变量	/44
	3.5	类的发现与设计	/46
	3.5.1	分类	/46
	3.5.2	抽象	/47
	3.5.3	封装	/48
	3.5.4	CRC 卡	/49
	3.6	对象的交互	/50
	3.6.1	对象协作	/50
	3.6.2	不可变对象	/51
	3.7	UML 类图及关系	/52
	3.7.1	类图	/53

3.7.2	依赖	/53
3.7.3	聚合	/54
3.7.4	组合	/55
3.8	日志分析系统设计测试	/55
3.9	本章小结	/58
3.10	本章习题	/58

第4章 数组和字符串 /61

4.1	为什么需要数组	/61
4.2	数组的基本用法	/61
4.2.1	数组的创建和初始化	/62
4.2.2	数组操作	/62
4.2.3	for-each 循环	/63
4.2.4	数组复制	/64
4.2.5	传递数组	/65
4.2.6	可变长参数列表	/66
4.2.7	二维数组	/66
4.2.8	二维数组的应用	/67
4.3	数组的应用	/69
4.3.1	二分查找	/69
4.3.2	Arrays 类	/70
4.3.3	对象数组	/71
4.4	String	/72
4.4.1	构造字符串	/72
4.4.2	不可变字符串	/73
4.4.3	字符串长度	/74
4.4.4	字符串比较	/74
4.4.5	equals()方法	/75
4.4.6	子串查找	/75
4.4.7	截取子串	/76
4.4.8	分割字符串	/77
4.4.9	int 转换为 String	/78

4.4.10	自动装箱/拆箱(Autoboxing /Unboxing)	/78
4.4.11	增强的 switch 语句	/79
4.5	StringBuffer	/80
4.5.1	StringBuffer 的初始化	/81
4.5.2	StringBuffer 的常用方法	/81
4.5.3	回文数的判断	/83
4.6	正则表达式	/84
4.6.1	匹配量词	/85
4.6.2	split()方法	/86
4.6.3	字符串替换	/87
4.7	Scanner 类	/88
4.8	本章小结	/89
4.9	本章习题	/89
第 5 章 继承和接口 /92		
5.1	包	/92
5.1.1	package 语句	/92
5.1.2	import 语句	/93
5.1.3	jar 命令	/94
5.2	继承	/95
5.2.1	创建子类	/95
5.2.2	子类能做的事	/96
5.2.3	构造方法与子类的内存结构	/98
5.2.4	Java 中的修饰符	/100
5.3	多态	/102
5.3.1	向上转型	/102
5.3.2	方法重写	/103
5.3.3	多态的实现	/103
5.3.4	equals()方法	/106
5.3.5	实现多态的注意事项	/106
5.4	抽象类	/106

- 5.4.1 抽象类的实例化 /107
- 5.4.2 final 的作用 /108
- 5.5 继承与组合 /109
- 5.6 接口 /112
 - 5.6.1 接口定义 /113
 - 5.6.2 接口实现 /113
 - 5.6.3 接口的特性 /115
- 5.7 接口和抽象类的区别 /116
- 5.8 内部类 /118
 - 5.8.1 静态内部类 /119
 - 5.8.2 成员内部类 /120
 - 5.8.3 局部内部类 /120
 - 5.8.4 匿名内部类 /121
- 5.9 面向对象的设计原则* /123
 - 5.9.1 最小化访问权限 /125
 - 5.9.2 以类代替基本数据类型 /126
 - 5.9.3 单一职责 /126
 - 5.9.4 不要重复造轮子 /127
 - 5.9.5 开放封闭原则 /128
 - 5.9.6 里氏替换原则 /130
 - 5.9.7 依赖倒置原则 /131
 - 5.9.8 迪米特法则 /134
- 5.10 Lambda 表达式 /134
 - 5.10.1 函数式编程 /134
 - 5.10.2 Lambda 表达式语法 /135
 - 5.10.3 函数式接口 /135
 - 5.10.4 自动类型推导 /136
 - 5.10.5 方法引用 /136
 - 5.10.6 接口的默认方法 /136
- 5.11 重构日志数据分析器 /137
- 5.12 本章小结 /138
- 5.13 本章习题 /139

第 6 章	异常处理	/144
6.1	为什么引入异常处理	/144
6.2	Java 中的异常	/147
6.2.1	异常架构	/147
6.2.2	异常抛出	/148
6.2.3	异常处理	/148
6.3	异常处理	/149
6.3.1	捕获异常	/149
6.3.2	使用 throws 声明异常	/151
6.3.3	throw 抛出异常对象	/152
6.3.4	异常转译	/153
6.4	自定义异常	/153
6.5	异常处理的原则	/157
6.6	本章小结	/160
6.7	本章习题	/160
第 7 章	反射与注解	/164
7.1	反射	/164
7.2	反射的实现	/164
7.2.1	Class 类	/165
7.2.2	获取构造器	/166
7.2.3	获取成员变量	/167
7.2.4	获取方法	/167
7.2.5	数组	/168
7.3	动态代理*	/169
7.3.1	无代理类的调用	/169
7.3.2	有代理类的调用	/170
7.3.3	动态代理	/171
7.3.4	装饰 IVehicle	/172
7.3.5	通用装饰代理类	/173
7.4	注解	/174
7.4.1	什么是注解	/174

7.4.2	为什么使用注解	/175
7.4.3	如何使用注解	/175
7.4.4	开发注解	/178
7.5	元注解	/179
7.5.1	限定生命周期@Retention	/179
7.5.2	限定使用对象@Target	/181
7.5.3	生成文档@Documented	/181
7.5.4	子类继承@Inherited	/182
7.6	运行时读取注解	/183
7.7	实例分析	/185
7.7.1	过滤方法	/185
7.7.2	自动化测试框架	/187
7.8	单一抽象方法注解	/189
7.9	本章小结	/190
7.10	本章习题	/190
第 8 章 集合与泛型 /192		
8.1	为什么需要集合	/192
8.2	Java 集合框架	/193
8.2.1	集合接口	/193
8.2.2	集合类	/195
8.3	泛型	/196
8.3.1	泛型类	/196
8.3.2	泛型方法	/196
8.3.3	类型限制	/197
8.3.4	类型擦除	/198
8.3.5	开发泛型类	/198
8.3.6	泛型最佳实践	/200
8.4	List 接口	/201
8.4.1	ArrayList	/201
8.4.2	LinkedList	/203
8.4.3	抽象层次的一致性	/205

8.4.4	可变对象作为容器参数	/205
8.5	Set 接口	/206
8.5.1	HashSet	/206
8.5.2	比较器	/208
8.5.3	TreeSet	/210
8.6	Map 接口	/211
8.6.1	Hashtable	/212
8.6.2	HashMap	/213
8.7	迭代器	/213
8.8	集合算法	/216
8.8.1	Collections	/216
8.8.2	Lambda 与批数据操作	/218
8.8.3	Stream 实现批操作	/218
8.8.4	惰性求值	/220
8.9	同步性与数据增长	/221
8.10	本章小结	/221
8.11	本章习题	/222
	第 9 章 图形用户界面	/225
9.1	图形用户界面的概念和组成	/225
9.2	Swing 组件库	/226
9.2.1	Swing 组件分类	/226
9.2.2	组件基类 JComponent	/227
9.2.3	GUI 设计步骤	/228
9.3	顶层容器	/229
9.3.1	JFrame	/229
9.3.2	JWindow	/230
9.4	中间容器	/231
9.4.1	Jpanel	/232
9.4.2	JTabbedPane	/232
9.4.3	JScrollPane	/234
9.4.4	对话框	/235

9.4.5	文件对话框	/237
9.5	基本组件	/239
9.5.1	按钮控件	/239
9.5.2	复选框 JCheckBox	/241
9.5.3	单选按钮 JRadioButton	/241
9.5.4	下拉列表框 JComboBox	/243
9.5.5	列表框 JList	/245
9.6	不可编辑组件	/247
9.6.1	JLabel	/247
9.6.2	菜单和工具栏	/247
9.6.3	工具提示	/250
9.7	可编辑组件	/251
9.7.1	文本输入	/251
9.7.2	文本区 JTextArea	/251
9.7.3	密码框 JPasswordField	/251
9.8	布局管理器	/252
9.8.1	为什么要使用布局管理	/253
9.8.2	布局方式	/253
9.8.3	CardLayout	/254
9.8.4	BorderLayout	/255
9.8.5	FlowLayout	/255
9.8.6	GridLayout	/256
9.8.7	BoxLayout	/258
9.8.8	GroupLayout	/259
9.8.9	GridBagLayout	/261
9.9	事件处理机制	/264
9.9.1	事件处理的角色	/264
9.9.2	事件分发机制的实现：观察者模式	/265
9.9.3	事件分类	/268
9.9.4	委托事件模型	/269
9.9.5	事件监听器	/272

9.9.6	事件适配器	/274
9.9.7	自定义用户事件	/277
9.10	模型	/279
9.11	拖放操作	/280
9.12	综合示例	/281
9.12.1	JTree	/281
9.12.2	JTable	/286
9.13	本章小结	/292
9.14	本章习题	/292
第 10 章 输入输出流 /294		
10.1	流的概念	/294
10.1.1	Java 中的 I/O 流	/295
10.1.2	Java 流框架和分类	/295
10.2	辅助类	/296
10.2.1	新建目录	/297
10.2.2	文件操作	/298
10.2.3	文件名过滤器	/300
10.3	字节流	/301
10.3.1	InputStream	/302
10.3.2	OutputStream	/302
10.3.3	文件输入输出流	/303
10.3.4	缓冲输入输出流	/305
10.3.5	数据输入输出流	/307
10.3.6	字节数组输入输出流	/309
10.3.7	随机文件存取	/310
10.4	字符流	/312
10.4.1	Reader 抽象类	/312
10.4.2	Writer 抽象类	/312
10.4.3	FileReader/FileWriter	/313
10.4.4	BufferedReader/BufferedWriter	/315
10.4.5	字节流和字符流之间的转换	/316

10.5	Java New I/O	/317
10.5.1	通道	/318
10.5.2	文件通道	/318
10.5.3	缓冲区	/319
10.5.4	容量、位置和读写范围	/321
10.5.5	分散与聚集	/322
10.5.6	编码与字符集	/323
10.5.7	通道间的数据传输	/324
10.6	本章小结	/325
10.7	本章习题	/326
第 11 章 多线程编程 /327		
11.1	进程与线程	/327
11.2	创建线程	/328
11.2.1	继承 Thread 创建	/328
11.2.2	实现 Runnable 接口	/329
11.2.3	定时器	/330
11.3	线程状态的转换	/331
11.4	线程控制	/332
11.4.1	线程睡眠	/333
11.4.2	线程让步	/333
11.4.3	线程加入	/333
11.4.4	线程的优先级	/334
11.4.5	线程分组管理	/335
11.5	线程同步	/335
11.5.1	同步方法	/337
11.5.2	同步代码块	/338
11.6	线程协作	/339
11.6.1	wait 与 notify 原语	/339
11.6.2	生产者-消费者问题	/340
11.7	线程池	/345
11.7.1	使用线程池的原因	/345

11.7.2	创建线程池	/346
11.7.3	提交任务到线程池	/347
11.7.4	关闭线程池	/348
11.7.5	监控线程池	/348
11.7.6	使用线程池	/349
11.8	线程同步控制的新特征	/353
11.8.1	条件变量	/353
11.8.2	原子变量	/356
11.8.3	障碍物	/357
11.8.4	信号量	/359
	11.8.5 Callable 与 Future	/360
11.9	综合实例	/362
11.10	本章小结	/367
11.11	本章习题	/367
第 12 章 网络编程 /369		
12.1	计算机网络通信	/369
12.2	网络编程范型	/370
12.3	基于 URL 的网络编程	/372
12.3.1	使用 URL 读取资源	/372
12.3.2	通过 URLConnection 建立连接	/372
12.3.3	设置连接属性	/373
12.3.4	URLDecoder 和 URLEncoder	/374
12.4	Socket 编程	/374
12.4.1	Socket 通信过程	/374
12.4.2	创建 Socket	/375
12.4.3	编程实例	/376
12.4.4	Socket 的选项	/377
12.4.5	关闭 Socket	/378
12.4.6	Socket 状态测试	/379
12.5	UDP 编程	/379