

宝典丛书

100万

Java JDK

实例

宝典

书中凝结了笔者多年开发经验，代码都是从实际项目中提取的精华

以J2SE 5.0为开发环境，结合了最新的Java技术

通过169个典型实例，囊括了JDK的所有细节

具有良好的编程风格，注释详细，方便学习

独特的目录形式，让你耳目一新，快速把握书中内容



电子工业出版社

Publishing House of Electronics Industry

<http://www.phei.com.cn>

夏先波 编著

宝典丛书

Java JDK 实例宝典

夏先波 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以J2SE 5.0为开发环境,选取Java应用的典型实例,循序渐进地介绍了Java语言的各种开发方法和技巧。全书共有19个章节,169个实例,内容涉及Java的语言基础、面向对象程序设计、数字处理、数组与集合、字符串、异常处理、文件操作、多线程、Swing编程、图形和多媒体编程、反射机制、网络程序设计、数据库编程、Applet、Java与XML、Java Mail、JSP与Servlet,并专门用一章介绍J2SE 5.0的新特性。本书内容丰富,结构清晰,实例的选择紧贴实际应用,具有很强的实用性和针对性,力求让读者通过实例学到更多、更好的编程方法和技术。

本书适用于Java初级、中级开发人员。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Java JDK 实例宝典 / 夏先波编著. —北京: 电子工业出版社, 2007.1

(宝典丛书)

ISBN 978-7-121-03651-4

ISBN 978-7-900222-03-0 (光盘)

I.J... II.夏... III.JAVA语言—程序设计 IV.TP312

中国版本图书馆CIP数据核字(2006)第154741号

责任编辑: 张月萍

印 刷: 北京市天竺颖华印刷厂

装 订: 三河市金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编: 100036

开 本: 787 × 1092 1/16 印张: 48.25 字数: 1296千字

印 次: 2007年1月第1次印刷

定 价: 79.00元(含光盘一张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系电话:(010) 68279077; 邮购电话:(010) 88254888。

质量投诉请发邮件至 zllts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

Java 语言是一种新兴的网络编程语言，其卓越的功能和特性为无数开发人员所推崇，越来越多的应用开发采用了基于Java技术的解决方案。Java语言也是一种真正面向对象的编程语言，它提升了应用程序的编程概念和开发思维。Java 语言提供了丰富的开发类库，为用户编程提供了极大的支持和方便。

目前最新的Java标准开发环境是J2SE 5.0，它在1.4版本的基础上增加了自动装箱和拆箱、泛型编程、枚举类型、可变长参数、静态导入等功能，使编写程序更加方便；强化了1.4版本中的线程编程，可以很方便地进行多线程编程。

为了帮助更多的Java开发人员提高开发技巧，掌握最新的Java特性，笔者精心编著了本书。本书以实例为主，Java基本理论部分介绍相对较少，编程的技巧和方法介绍很多，读者在阅读完本书后可以提高自身的编程技巧，并掌握J2SE 5.0的新特性。

本书在内容的编排和目录组织上都十分讲究，争取让读者能够快速掌握实例的实现方法。统一按照实例功能来为章节命名，让读者明确知道每一节将实现什么实例。具体讲解实例的时候，分为4个步骤，首先介绍实例的功能，然后介绍实现实例时的关键技术点，接着介绍实例的实现代码和运行效果，最后对程序的关键部分进行解读。本书的代码具备良好的编程风格和详尽的中文注释，使读者能够很容易地理解程序代码，并养成良好的编程风格。

本书的特点

1. 实例经典，内容丰富

本书实例丰富，共有19章169个实例，每个实例都紧贴实际应用，如用Servlet生成图形验证码，具有很强的示范性和实用价值，读者可以直接使用书中的实例解决实际问题。

2. 侧重编程技巧

本书所有的实例在Java基本理论方面介绍不多，重点在讲述常用、实用的编程技术。书中采用了多种设计模式，如Singleton、Adapter、MVC等模式，同时在实例中强调应避免的不良编程习惯。

3. 以J2SE 5.0为开发环境

目前介绍J2SE 5.0新特性的书籍很少，很多开发人员对这些新特性不了解，在实际的项目开发中也很少使用这些新特性。本书用一章专门介绍J2SE 5.0的新特性，精心准备了19个实例，通过这些实例可以帮助读者轻松掌握这些新特性的特点，并实际应用于项目开发中，读者会发现，J2SE 5.0的特性的确有利于提高开发效率。

4. 层次清晰，主次分明

全书分为19个章节，包括了Java技术的方方面面，章节之间的先后顺序很重要。本书在章

节安排上采用了由浅入深的策略：先介绍基本的技术，为后面章节打好基础；在后面章节介绍高级的技术，并在实例中使用前面章节介绍的技术，起到巩固的作用。

在讲解实例的时候，首先介绍实例的目标，让读者了解该实例要做什么，使读者在脑海中形成一个实现思路；接着介绍实例中使用的关键技术点，帮助读者解决实现思路中的问题；最后对代码的关键部分进行解读，帮助读者掌握关键技术点。

5. 良好的编程风格，注释详细

每个实例的程序都是经过精心设计的，在实现实例功能的同时，注重程序运行的效率。程序代码具备良好的编程风格，结构清晰，注释非常详细，能帮助读者轻松掌握实例的实现过程。

学习 Java 的心得体会

1. 熟练掌握 Java 的基本语法

如果读者会 C 编程，那么学习 Java 将会非常容易。学习基本语法时，重点理解各关键字的功能与用法，以及功能相似的关键字之间的区别，如 `continue` 和 `break`。

2. 理解 Java 的面向对象相关概念

面向对象是当今最成熟的程序设计方法，它有 3 个非常重要的特性：继承、封装和多态。Java 是一种面向对象的程序设计语言，在 Java 中，一切都是对象，体现了面向对象的封装性；Java 类之间允许单重继承，体现了继承性；多态性主要体现在允许定义接口类、抽象类，子类能够覆盖和隐藏父类的变量和方法，也属于多态性。读者要理解面向对象的概念，掌握如何在 Java 中进行面向对象编程。

3. 善于使用 JDK 自带的帮助文档

Java 提供了非常丰富的类库，要想掌握常用类的用法，最好的方法是查阅 JDK 自带的帮助文档，尽量看英文原版的，不要依赖于中文版的帮助文档或 API 参考书籍。

4. 多动手写 Java 程序

学习 Java 的最终目标就是为了写程序，解决问题。掌握了 Java 的基本语法、面向对象的概念，能够读懂简单的例子后，接下来就要亲手写程序。第一步要把书上的例子在本机上运行通过；第二步要敢于修改书上的程序；第三步就要合上书本，独立编写程序，哪怕程序的功能与书上的例子一样。

5. 多阅读开源项目的源代码

要想具备良好的编程风格，掌握好的编程技术，阅读优秀的源代码是一条捷径。当前很多开源项目的源代码都值得学习（如 Apache 组织的项目以及 JDK 的源代码）。在阅读源代码时，要注意两点：第一是学习项目的整体框架，结合项目的文档，一步步地调试程序是最常用的手段；第二是学习代码中的技巧、算法，这是一个积累的过程。

6. 理解 Java 虚拟机的运行机理

Java 之所以能够跨平台，主要是因为存在 Java 虚拟机。Java 虚拟机运行在操作系统中，不同操作系统下的 Java 虚拟机不同。将 Java 类编译成字节码，在 Java 虚拟机中运行字节码，使 Java 程序独立于操作系统，具备跨平台的能力。为了更好地理解 Java 程序的运行机理，编写出

更优秀、更高效的程序，读者需要理解 Java 虚拟机的类加载机制和垃圾回收机制等。

本书包括的内容

第1章介绍 Java 的基础知识，实例包括：Java 的基本数据类型，Java 的各种运算符，Java 程序的流程控制，以及如何在 Java 程序中使用命令行参数。

第2章介绍如何用 Java 进行面向对象的程序设计，实例包括：类的设计，对象的克隆，类的继承，抽象类，变量和方法的覆盖与隐藏，Java 的参数传递，类的加载顺序，接口的设计和几个程序设计中常用的设计模式，如单例模式、工厂模式和适配器模式。

第3章介绍 Java 中的数字，实例包括：数字的封装类，数字的舍入，格式化数字，数字进制的转换，生成随机数以及大数字的处理。

第4章介绍 Java 中的数组和集合，实例包括：使用 Arrays 类操作数组，利用数组求质数，动态调整数组的大小，利用二维数组实现矩阵，Java 各种 List 实现类的区别，结合 Random 和 List 实现一个不重复的随机数序列，利用 LinkedList 实现一个先进先出的队列，对 List 中的元素进行排序，Java 中各种 Set 实现类的区别，集合与数组的相互转化，Java 中各种 Map 实现类的区别，对 Map 中的元素进行排序，最后介绍常用于解析配置文件的 Properties 类。

第5章介绍 Java 的字符串操作，实例包括：判断一个字符串是否是合法的 Java 标识符，18 位身份证号码格式的验证，实现一个简单的表达式解析器，对密码进行加密和验证，制作命令程序和用正则表达式验证电话号码的格式。

第6章介绍 Java 的异常处理，实例包括：用于声明和抛出异常的 throws 和 throw 语句的用法，用于捕获处理异常的 try、catch 和 finally 语句的用法，自定义异常类和使用异常的几个原则。

第7章介绍 Java 的输入输出流，重点介绍文件输入输出流的处理，实例包括：获取文件的描述信息、操作文件和目录、读写文件、搜索文件、序列化和反序列化对象、压缩和解压缩文件、处理 Excel 和 PDF 文件，最后介绍一个自定义的写日志文件的类。

第8章介绍 Java 的线程，实例包括：线程的定义，线程的启动与停止，线程的同步，线程 join，线程的优先级，守护线程，线程的死锁，定时器，用线程实现生产者和消费者例子，以及一个简单的线程池。

第9章介绍利用 Java 的 AWT 和 Swing 工具包生成图形用户界面 (GUI)，实例包括：实现图形日历，开窗户小游戏，标准型计算器，更改组件的外观，自定义对话框，制作程序的欢迎画面，一个简单的文本编辑器和 Swing 控件的 Drag 和 Drop。

第10章介绍 Java 的图形处理，实例包括：实现一个圆形按钮，捕捉屏幕，缩放图片，画 2D 和 3D 图形，最后实现一个圆形的时钟。

第11章介绍用 Java 处理多媒体，包括动画、音频和视频的处理。实例包括：滚动的文消息，三维弹球和贪吃蛇游戏，3 种播放音频的方法，最后实现一个媒体播放器。

第12章介绍 Java 的反射机制，实例包括：使用 instanceof 操作符判断对象的类型，通过类名查看类的各种信息 (如类声明的属性、构造方法、公有方法等)，最后介绍了如何动态地调用类的方法。

第13章介绍 Java 的网络编程，实例包括：从 URL 中提取信息，Web 浏览器，获取 IP 地址和域名，访问 HTTP 服务器的客户端，实现 HTTP 服务器，基本的 Socket 编程，一个支持多线程的



服务器框架，基于服务器框架的代理服务，访问 Telnet 服务的客户端，UDP 编程，聊天室的服务器和客户端，以及访问 FTP 站点的客户端。

第 14 章介绍 Java 的数据库编程，实例包括：连接各种数据库，获得数据库和数据表的元数据，查询和更新数据库，SQL 语句的批处理，事务的提交和回滚，使用 PreparedStatement 执行 SQL 语句，读写二进制数据到数据库，读写 Blob 数据到数据库，使用 ResultSet 更新数据库，使用 RowSet 操作数据库，调用存储过程，最后实现了一个数据库连接池。

第 15 章介绍 Applet 编程，实例包括：实现一个 Applet 时钟，在 Applet 中处理键盘和鼠标事件，英文打字游戏，两个 Applet 间的通信，最后用 Applet 实现一个汉诺塔的游戏。

第 16 章介绍 J2SE 5.0 的新特性，实例包括：简单类型的自动装箱和拆箱，新的 for 循环，枚举类型，方法和属性的静态导入，可变长的参数，类 C 风格的格式化输出，使用 ProcessBuilder 调用本地命令，泛型编程，注释功能，Java 虚拟机的监控与管理，以及 Java 线程编程。相比 J2SE 1.4，线程编程有了质的飞跃，主要体现在：Callable 和 Future、新的任务执行框架、Lock 锁、Condition 条件、Semaphore、CountDownLatch、CyclicBarrier、Exchanger 和 BlockingQueue。

第 17 章介绍用 Java 操作 XML 文档。实例包括：用 DOM 处理 XML 文档，用 SAX 处理 XML 文档，用 XSLT 转换 XML，以及 XML 文档与对象之间的相互转换。

第 18 章介绍用 Java 收发邮件，实例包括：用 SMTP 协议发送简单邮件，发送带附件的邮件，发送邮件给多人和用 POP3 协议收邮件。

第 19 章介绍 JSP (Java Server Page) 和 Servlet 技术，实例包括：获取访问 JSP 页面的客户端的真实 IP，在 JSP 中读取和设置 Cookie，无刷新的 JSP 聊天室，在 JSP 中上传文件，用 Servlet 生成图形验证码和用 Servlet 实现分页查看数据库。

适合阅读本书的读者

本书具有知识全面、实例精彩、指导性强的特点，力求以全面的知识性及丰富的实例来指导读者透彻学习 Java 各方面的技术。本书可以帮助 Java 初级、中级开发人员提高开发技能，掌握 J2SE 5.0 的新特性，书中的实例对高级开发人员也有一定的启发意义。

参加本书编写和整理的人员还有原奕、殷颖、张传毓、张帆、张岫、郑林、邹竹彪、陈杰、程志艳、董方、范永龙、郭瑞、郭刚、段文睿、刘乃丽、张庆华、李潞、贾来喜、陈晨、王建、王坤、徐宁、夏强、赵志升、尹亮亮、王征、姜海英、陈洁、李刚、刘智勇、刘丹、刘小东。由于时间有限，疏漏之处在所难免，恳请广大读者批评指正。

作者

2007 年 1 月



目 录

第 1 章 Java 基础	1
1.1 转换基本数据类型	1
本节实例讲述了Java基本类型的自动提升规则和强制类型转换技术，重点介绍了何时需要进行强制类型转换，以及在进行转换时由于舍入而引起的精度问题。	
1.2 Java 的运算符	4
本节实例讲述了Java各种运算符的用法，包括计算运算符、比较运算符、布尔运算符、赋值运算符、位运算符和其他运算符。重点介绍了功能相似的运算符的差别。	
1.3 控制程序的流程	10
本节实例讲述了Java中用于控制程序流程的语句的用法，用各种控制语句把一个整数值逐步变成目标值。重点介绍了各种控制流程语句的用法以及它们适用的场合。	
1.4 计算阶乘	13
本节实例讲述了for语句的用法，实现了求非负整数的阶乘的功能。重点介绍了如何确定方法的返回类型和参数范围，对于大数字计算，要避免溢出。	
1.5 实现命令行程序	14
本节实例讲述了Java中实现命令行程序的技术，实现了一个简单的命令行程序：用户给命令行程序提供一系列参数，程序返回参数的值。重点讲述了读取命令行程序参数的方法。	
第 2 章 Java 面向对象程序设计	16
2.1 复数类	16
本节实例讲述了Java类的设计方法，实现了一个复数类，可以按需构造复数类，并实现了复数的加法、减法、乘法和除法运算。重点讲述了如何以面向对象的技术设计Java类。	
2.2 equals、hashCode 和 clone 方法	20
本节实例讲述了Object类的equals、hashCode和clone方法的作用，一般自定义的数据类需要覆盖这些方法，本节在2.1节的基础上，实现了一个更加完整的复数类。重点讲述了覆盖这3种方法所用到的基本技术。	
2.3 Java 的参数传递	23
本节实例讲述了在调用对象的方法时，Java是如何传递参数的，对于对象类型的参数和基本类型的参数，Java虚拟机采用了不同的处理方式。	
2.4 自定义形状类	24
本节实例讲述了类的继承、抽象类的技术，定义了一个形状类，并实现了三角形和矩形类，它们都继承形状类。	
2.5 类的加载顺序	30
本节实例讲述了具有继承关系的父、子类的加载顺序，定义了一个父类和一个子类，在新建一个子类对象时，将会清晰地看到代码的执行顺序。	
2.6 方法和变量在继承时的覆盖与隐藏	33
本节实例讲述了具有继承关系的父、子类的方法和变量可能引起的覆盖和隐藏问题，	

定义了一个父类和一个子类，它们之间有同名的方法和变量，在访问子类或者父类的对象的方法或变量时，将会清晰地看到究竟哪个方法或变量被访问了。

- 2.7 排序类 36
本节实例讲述了接口技术，定义了一个实现排序功能的接口类，然后用4种排序方法（选择排序法、冒泡法、线性插入法和快速排序法）实现了这个接口。
- 2.8 Singleton 单例模式 42
本节实例讲述了一种常见的软件设计模式：单例模式，常用于创建一个全局共享的对象，即在整个应用程序中只存在一个对象，介绍了2种实现单例模式的常用方法。
- 2.9 Factory 工厂模式 45
本节实例讲述了另一种常见的软件设计模式：工厂模式，常用于创建那些用户只关心对象功能而不关心创建过程的对象。用工厂模式实现各种排序类对象的创建。
- 2.10 Adapter 适配器模式 47
本节实例讲述了另一种常见的软件设计模式：适配器模式，常用于将一个类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。用适配器模式实现了一个既可以打印数组，又能够对数组进行排序的类。

第3章 数字 49

- 3.1 数字与数字封装类 49
本节实例讲述了基本类型的数据与封装类对象之间互相转换的技术，基本类型包括字节（byte）、短整型（short）、整型（int）、长整型（long）、单精度浮点型（float）、双精度浮点型（double）、布尔型（boolean）、字符型（char），其相应的封装类分别为：字节类（Byte）、短整型类（Short）、整型类（Integer）、长整型类（Long）、单精度浮点型类（Float）、双精度浮点型类（Double）、布尔类（Boolean）和字符类（Character）。
- 3.2 格式化数字 51
本节实例讲述了格式化数字的技术，可以给数字配上单位并设置精度，或者用科学计数法表示数字，主要使用了 `java.text.DecimalFormat` 类。
- 3.3 数字的舍入 53
本节实例讲述了数字的舍入技术，实现了多种舍入模式，如四舍五入等。返回结果可以是整数，也可以是小数，能够指定舍入后结果的小数位数。实例中主要使用了 `java.math.BigDecimal` 类。
- 3.4 转换数字的进制 57
本节实例讲述了数字进制的转换技术，实现二进制、八进制、十进制、十六进制数之间的互相转换，还可以把数字转换成任意进制的数。
- 3.5 生成随机数 58
本节实例讲述了生成随机数序列的技术，能指定随机数所在的区间（范围），但不保证生成的随机数序列中没有重复的元素。实例中主要使用了 `java.util.Math` 类和 `java.util.Random` 类。
- 3.6 处理大数字 61
本节实例讲述了处理大数字的技术，实现了任意非负整数的阶乘计算，支持高精度的加法、减法、乘法和除法运算。

第4章 数组与集合 66

- 4.1 使用 Arrays 66

本节实例讲述了 java.util.Arrays 类的使用, 通过它实现了数组的赋值、排序, 比较两个数组元素是否相同, 在排好序的数组中查找某个元素。	
4.2 求质数	68
本节实例讲述了一种通过布尔数组求质数的技术, 实现了将指定范围内所有质数打印出来的功能。	
4.3 动态调整数组长度	70
本节实例讲述了复制数组的技术, 实现了数组长度的动态调整, 而原数组的内容保持不变。实例中主要使用了 java.lang.System 类的 arraycopy 方法。	
4.4 矩阵	72
本节实例讲述了多维数组的技术, 实现了矩阵类, 并支持矩阵的基本运算, 包括: 矩阵的构造、矩阵的加法、矩阵的减法、矩阵的数乘、矩阵与矩阵相乘、矩阵的除法、矩阵的逆矩阵和转置矩阵等。	
4.5 ArrayList、Vector 和 LinkedList	84
本节实例讲述了 Java 的列表接口 java.util.List, 演示了它的三个实现类 (ArrayList、Vector 和 LinkedList) 的用法和区别, 重点介绍了它们各自适用的场合。	
4.6 生成不重复的随机数序列	89
本节实例讲述了两种生成不重复随机数序列的技术, 在 3.5 节介绍的生成随机数的技术基础上, 使用 Java 的列表, 使得随机数序列不重复。	
4.7 自定义队列	92
本节实例讲述了使用 Java 列表实现队列的技术, 通过 LinkedList 实现了队列的基本功能, 包括元素的插入、弹出、查看、删除和查找, 能判断队列是否为空, 并能清空队列里的所有元素。	
4.8 对 List 排序	95
本节实例讲述了给 Java 列表排序的技术, 主要使用了 java.util.Collections 类的 sort 方法, 通过自定义比较器 (java.util.Comparator), 可以使列表中的元素按照自己想要的顺序排列。	
4.9 HashSet、LinkedHashSet 和 TreeSet	97
本节实例讲述了 Java 的集合接口 java.util.Set, 演示了它的三个实现类 (HashSet、LinkedHashSet 和 TreeSet) 的用法和区别, 重点讲述了它们各自适用的场合。	
4.10 列表、集合与数组的互相转换	100
本节实例讲述了数组、列表和集合之间相互转换的技术, 通过 Arrays 的 asList 方法能够把数组转换成 List 和 Set 对象, 而 List 和 Set 对象的 toArray 方法能把 List 和 Set 对象转换成数组。	
4.11 HashMap、Hashtable、LinkedHashMap 和 TreeMap	102
本节实例讲述了 Java 的映射接口 java.util.Map, 演示了它的 4 个实现类 (HashMap、Hashtable、LinkedHashMap 和 TreeMap) 的用法和区别, 重点讲述了它们各自适用的场合。	
4.12 对 Map 排序	106
本节实例讲述了对 Map 中元素根据键进行排序的技术, 把 Map 对象转换成 TreeMap, 就能自动地实现排序。	
4.13 Properties 属性文件	108
本节实例讲述了 Java 的属性接口 java.util.Properties, 介绍 Properties 的特点和用法, 包括向 Properties 中添加键值对, 从 Properties 中取出某个键的值, 把整个 Properties 对象	

保存到文件，以及将文件中的键值对信息直接加载到 Properties 对象中。

第 5 章 字符串	111
5.1 使用 String	111
本节实例讲述了用 String 类操作字符串的技术，包括查找子串、截取子串、替换子串、转换字符串大小写等。	
5.2 基本数据类型与字符串的转化	116
本节实例讲述了基本数据类型与字符串的互相转化的技术，如把一个 int 值 555 转换成字符串 “555”，把字符串 “555” 转换成 int 值 555。	
5.3 判断 Java 标识符	117
本节实例讲述了如何判断一个字符串是合法的 Java 标识符的技术，主要使用了 java.lang.Character 类的 isJavaIdentifierPart 方法。	
5.4 使用 StringBuffer	119
本节实例讲述了常用于修改字符串的 StringBuffer 类的用法，包括查找子串、截取子串、替换子串、插入子串、删除子串和对字符串进行排序。	
5.5 IP 地址转化成整数	122
本节实例讲述了数字和字符串的转换技术，将字符串形式的 IP 地址转换成整数，再把整数形式的 IP 地址转换成字符串。	
5.6 18 位身份证格式验证	124
本节实例讲述了验证 18 位身份证格式的技术，格式正确的身份证号码不一定是真的，但是格式不正确的身份证号码一定不是真的。	
5.7 表达式解析器	127
本节实例实现了一个表达式解析器，它可以计算由数字、运算符和括号组成的表达式的值，并能处理变量。为了处理简单，它只支持一个字母的变量，且不区分变量字母的大小写，因此，最多只支持 26 个变量。	
5.8 字符串编码的转换	136
本节实例讲述了字符串编码的转换技术，能把一个任意编码格式的字符串转换成另一个编码格式的字符串，这在 Web 应用和数据库访问中会经常使用。	
5.9 字符串对齐器	139
本节实例实现了一个能对齐字符串的工具类，可以指定字符串在显示时的对齐格式，有左对齐、居中和右对齐三种格式，并且能够指定一行最多能显示的字符数。	
5.10 密码加密与验证	143
本节实例讲述了在 Java 中使用 MD5 进行加密的技术，将输入字符串通过 MD5 加密成另一个字符串，而且根据新的字符串无法获得原字符串，但可以在不知道原始密码的情况下进行密码验证。实例中主要使用了 java.security.MessageDigest 类。	
5.11 制作命令行程序	145
本节实例讲述了健壮的命令行程序的开发技术，主要使用了 Apache 组织的 cli 项目提供的类库，专门用于解决命令行参数的解析问题。	
5.12 使用 StringTokenizer	150
本节实例讲述了使用 java.util.StringTokenizer 类分割字符串的技术，把一个字符串分割成多个标记，功能类似于 String 类的 split 方法。	
5.13 使用正则表达式操作字符串	152
本节实例讲述了 Java 中的正则表达式技术，包括精确匹配字符串、查找子串以及替换	

子串等。实例中主要使用了 <code>java.util.regex</code> 包。	
5.14 使用正则表达式验证电话号码的格式	161
本节实例实现了一个用于验证国内电话号码格式是否正确的正则表达式,只是验证电 话号码的格式,不验证电话号码是否有效。	
 第6章 Java 异常处理	164
6.1 <code>throw</code> 、 <code>throws</code> 、 <code>try</code> 和 <code>catch</code>	164
本节实例讲述了与 Java 异常处理机制相关的 4 个关键字的用法,包括 <code>throw</code> 、 <code>throws</code> 、 <code>try</code> 和 <code>catch</code> 。重点介绍了当异常发生时,Java 程序的执行流程。	
6.2 自定义异常类	166
本节实例讲述了自定义异常类的技术,通过继承 <code>java.lang.Exception</code> 或者 <code>java.lang.</code> <code>Throwable</code> 类,可以定义自己的异常类。	
6.3 使用 <code>finally</code>	169
本节实例讲述了异常处理中 <code>finally</code> 关键字的用法,无论是否发生异常, <code>finally</code> 块中的代 码都会被执行,实例演示了它的这一特性。	
6.4 使用异常的技巧与原则	172
本节实例讲述了一些异常处理的技巧和原则,因为异常处理涉及到程序流程的跳转, 所以,虚拟机需要保存程序的执行流程,以便异常发生时能进行正确的跳转,这也就 导致了使用异常时会引起额外的开销,所以,要谨慎地使用异常。	
 第7章 IO——输入输出流	175
7.1 获取文件的属性信息	175
本节实例讲述了用 <code>java.io.File</code> 类获取文件属性信息的技术,包括文件的文件名、文件 所在的目录、文件是否可读、是否可写、文件的大小以及最后修改时间等。	
7.2 列出指定目录下的文件	177
本节实例讲述了使用 <code>java.io.File</code> 类列出指定目录下的文件的技术,既可以列出目录 下的所有文件(包括子目录),还可以使用 <code>java.io.FilenameFilter</code> 过滤文件。	
7.3 创建文件和目录	180
本节实例讲述了在文件系统中创建新文件或者目录的技术,新建的目录是空的,新建 的文件大小为 0,还能够生成临时文件。实例中主要使用了 <code>java.io.File</code> 类的 <code>createNewFile</code> 、 <code>mkdir</code> 、 <code>mkdirs</code> 和 <code>createTempFile</code> 方法。	
7.4 删除文件和目录	183
本节实例讲述了在文件系统中删除文件或者目录的技术,能删除单个文件,还能把整 个目录,包括子目录下的文件全部删除。实例中主要使用了 <code>java.io.File</code> 类的 <code>delete</code> 方法。	
7.5 移动文件和目录	186
本节实例讲述了在文件系统中移动文件或者目录的技术,能移动单个文件,还能把整 个目录包括子目录下的文件全部移动到新的位置。实例中主要使用了 <code>java.io.File</code> 类的 <code>renameTo</code> 方法。	
7.6 复制文件和目录	190
本节实例讲述了在文件系统中复制文件或者目录的技术,能复制单个文件,还能把整 个目录包括子目录下的所有文件复制到新的位置。实例中主要使用了 <code>FileInputStream</code> 类和 <code>FileOutputStream</code> 类来读写文件。	
7.7 一个简单的文件搜索器	195

- 本节实例实现一个简单的文件搜索器,根据文件名进行搜索,并且支持通配符(通配符“?”表示任意一个字符,通配符“*”表示任意多个字符)。
- 7.8 读文件 197
本节实例讲述了读取文件系统中文件内容的技术,能够读文本文件(如.txt文件)、二进制文件(如.jpg文件),并采用了多种读文件的方式:按字节读取、按字符读取、按行读取和随机读取。
- 7.9 写文件 202
本节实例讲述了在文件系统中写文件的技术,能写文本文件、二进制文件,采用多种写文件的方式:按字节写、按字符写和按行写。
- 7.10 添加内容到文件尾 205
本节实例讲述了在文件系统中从文件尾部添加内容的技术,文件原内容保持不变,主要使用了 java.io.RandomAccessFile 类和 java.io.FileWriter 类。
- 7.11 文件的分割与合并 207
本节实例讲述了在文件系统中分割文件和合并文件的技术,将一个大文件分割成多个小文件,和将多个小文件合并成一个大文件。实例中主要使用了文件的输入输出流。
- 7.12 从键盘接收数据并输出到文件 210
本节实例讲述了从标准输入(键盘)中读取数据的技术,从键盘接收数据,并将数据保存到文件中。
- 7.13 使用 StreamTokenizer 统计文件的字符数 212
本节实例用 java.io.StreamTokenizer 分析流数据的技术,实现了统计文件字符数的功能,能统计出文件所包含的数字数、字母数、标点符号数以及总的字符数。
- 7.14 序列化和反序列化对象 214
本节实例介绍了对象的序列化和反序列化技术,通过序列化技术把对象保存到文件,通过反序列化技术把文件恢复成对象,被序列化的对象必须实现 java.io.Serializable 接口。实例中主要使用了 java.io.ObjectOutputStream 类和 java.io.ObjectInputStream 类。
- 7.15 控制对象的序列化和反序列 216
本节实例讲述了控制对象序列化和反序列的技术,能够有选择地序列化对象的部分数据,此时,对象必须实现 java.io.Externalizable 接口。
- 7.16 读 jar 包的资源文件 219
本节实例讲述了读取 jar 包中资源文件的技术.jar 包属于压缩文件,读其中的资源文件与读普通文件有很大差别,需要使用 java.lang.Class 类的 getResource 方法和 getResourceAsStream 方法。
- 7.17 用 Zip 格式压缩和解压缩文件 221
本节实例讲述了按 Zip 格式压缩和解压缩文件的技术,能将一个目录包括子目录下的全部文件压缩成 Zip 文件,然后解压到另一个目录下。实例中主要使用了 java.util.zip 包。
- 7.18 操作 Excel 文件 226
本节实例讲述了读写 Excel 文件的技术,能够生成一个 Excel 文件,并能解析出 Excel 文件中的内容,主要使用了 Apache 组织的 poi 项目提供的类库。
- 7.19 操作 PDF 文件 231
本节实例讲述了读写 PDF 文件的技术,能够生成一个 PDF 文件,并能解析出 PDF 文件中的内容,在生成 PDF 文件时使用了 iText 库,在读取 PDF 文件时使用了 pdfbox 库。
- 7.20 自定义日志文件类 236
本节实例实现一个日志文件类,在日志文件中以“时间:消息”的形式存储日志信息,

允许插入文本消息和异常消息。使用了单例模式，能够通过配置文件设置日志文件的路径。

第8章 线程	241
8.1 定义和启动线程	241
本节实例讲述了两种定义线程的方法，通过实现java.lang Runnable接口或者继承java.lang.Thread类，可以实现线程类，通过Thread类的start方法启动线程。	
8.2 停止线程	243
本节实例讲述了停止线程的技术，为了避免使用Thread的stop方法停止线程时可能引起的死锁，常用的方法是为线程设置一个运行标志位，通过修改标志位的值来停止线程。	
8.3 线程的互斥	245
本节实例讲述了多线程在使用临界区资源时的互斥技术，两个线程同时操作同一个账户，其中一个线程往里存钱，另一个从中取钱，在不使用互斥技术时，会导致操作前后不一致的问题，而使用互斥技术后，问题消失。实例中主要使用了synchronized关键字。	
8.4 线程的协作	248
本节实例讲述了多线程在使用临界区资源时的协作技术，一个线程不断地打印消息队列中的消息，而另一线程不断地往消息队列中添加消息。当消息队列为空时，打印线程处于阻塞状态，当消息队列不为空时，被堵塞的打印线程又继续运行。实例中主要使用了java.lang.Object类的wait、notify和notifyAll方法。	
8.5 线程join	251
本节实例讲述了多线程的汇聚技术。主线程启动多个子线程，而且这些子线程的运行时间不一样，而主线程必须等待这些子线程运行结束后才能继续往下执行。实例中主要使用了java.lang.Thread类的join方法。	
8.6 生产者、消费者问题	253
本节实例实现了一个生产者、消费者问题的解决方案，生产者负责生产产品，并将产品存放到仓库；消费者从仓库中获取产品并消费。当仓库满时，生产者必须停止生产，直到仓库有位置存放产品；当仓库为空时，消费者必须停止消费，直到仓库中有产品。本实例还允许多个生产者和多个消费者共享一个仓库。	
8.7 线程优先级	260
本节实例实现了设置线程优先级的技术，对于两个同时启动的线程，优先级高的线程先获取CPU资源，先被真正运行，优先级低的线程后获取CPU资源，后被运行。实例中主要使用了java.lang.Thread的setPriority方法。	
8.8 列出虚拟机中所有的线程	261
本节实例讲述了列举Java虚拟机中活动线程的技术，以树状结构显示所有活动线程的名字，主要使用了线程组java.lang.Thread的currentThread和getThreadGroup方法。	
8.9 守护线程 Daemon	263
本节实例讲述了实现守护线程的技术，它们在后台运行，为非守护线程提供服务，当所有非守护线程结束时，守护线程自动结束。实例中主要使用了Thread的setDaemon方法。	
8.10 线程池	266
本节实例通过线程组技术实现了一个线程池，将若干个执行任务的线程放在池中，当	

有任务要执行时，从池中取出一个空闲线程来处理任务，处理完任务后，又将线程对象放入池中。实例中主要使用了 `java.lang.ThreadGroup` 类。

8.11 一个死锁的例子 271

本节实例介绍了一个多线程因为争夺共享资源而死锁的例子，在实际编程过程中，要避免死锁，应及时释放不用的共享资源。

8.12 定时器 `Timer` 272

本节实例介绍了定时器的技术，定时执行指定的任务，包括在指定时间点执行任务，在指定时间之后执行任务，以及在某个时间之后，每隔某个时间段重复执行任务。实例中主要使用了 `java.util.Timer` 类和 `java.util.TimerTask` 类。

第9章 Java GUI 275

9.1 日历 275

本节实例讲述了 AWT 和 Swing 开发图形界面的技术，实现了一个具有图形界面的日历程序，可以查看星期、日期、月份和年份的信息。实例中主要使用了窗体、按钮、面板等控件，以及将这些控件组织在一起的布局管理器，并且为按钮添加了事件处理。

9.2 开窗户游戏 281

本节实例实现一个开窗户的小游戏，有 25 个小窗户，单击一个窗户，周围窗户的颜色就会发生变化，当所有的窗户变成同一颜色时，游戏获胜。游戏过程中可以重置所有窗户到初始状态。实例中主要使用了窗体、按钮、面板等控件，以及将这些控件组织在一起的布局管理器，并且为按钮添加了事件处理。

9.3 标准型计算器 286

本节实例实现了一个标准型计算器，该计算器的界面和功能与 Windows 操作系统的附件所带的计算器相仿，支持简单的加、减、乘、除运算。使用了文本框控件。

9.4 更改组件的外观 292

本节实例讲述了更改图形外观风格的技术，可以将图形界面变成 3 个不同的风格，分别是 `MetalLookAndFeel`、`MotifLookAndFeel` 和 `WindowsLookAndFeel` 风格。

9.5 自定义对话框 294

本节实例讲述了自定义控件的技术，实现了一个自定义的对话框，在对话框中输入文本，单击“确定”按钮后将输入的文本显示到主窗口中。

9.6 制作欢迎画面 297

本节实例讲述了制作欢迎（启动）画面的技术，欢迎画面既是宣传产品的方法之一，又可以标示应用程序未完全启动。本节实例实现的欢迎画面可以指定它所采用的图片和画面的显示时间，当用户单击画面时，画面关闭，当显示时间到期时，画面自动关闭。

9.7 一个简单的编辑器 300

本节实例实现了一个文本编辑器，能打开、编辑和保存 `html`、`java`、`cpp`、`txt` 文件，能够在文件中进行查找和替换，而且支持 Java 源文件的编译与运行。实例中主要使用了菜单栏、工具栏、分隔窗格、文本域、多页面板、文件选择器、选择对话框、表格、树等控件。

9.8 Swing 的 Drag 和 Drop 324

本节实例讲述了图形界面的拖放技术，程序界面为一个文本框和一个列表框，列表框中添加了若干文件的基本信息，当单击一个文件基本信息时，文本框中显示文件详细信息；能把文件从文件系统中拖放到列表框中，列表框中将增加新拖入文件的基本信

息; 可以把列表框中的文件基本信息拖放到文件系统中, 将会在文件系统中放置文件。

第 10 章 Java 图形	334
10.1 一个圆形的按钮	334
本节实例讲述了在图形界面中画圆的技术, 实现了一个圆形的按钮, 当单击按钮时, 按钮颜色发生变化。	
10.2 捕捉屏幕	336
本节实例讲述了捕捉显示器屏幕的技术, 相当于键盘上的 PrintScreen 键的功能。将捕获到的屏幕图像显示在一个画布上。	
10.3 缩放图片	339
本节实例讲述了缩放图片的技术, 实现了一个工具类, 能将原始图片放大或者缩小, 常用于为图像生成缩略图。	
10.4 2D 图形	344
本节实例讲述了画 2D 图形的技术, 实现一个 2D 的画板程序, 能够画 2D 的线、圆和矩形, 能指定图形的颜色, 并且能够撤销和恢复图形。实例中主要使用了 java.awt. Graphics 类。	
10.5 3D 图形	355
本节实例讲述了画 3D 图形的技术, 包括最基本的 3D 编程步骤、设置光源、变换物体坐标、设置纹理和材质, 以及将 3D 图形与 2D 图形结合的技术。	
10.6 一个时钟程序	364
本节实例实现了一个圆形的时钟, 时钟上有时、分、秒的刻度以及指针, 还能显示星期和日期信息。时钟的背景使用了多张图片, 图片内容不断发生变化, 透明度也不断变化。	
第 11 章 Java 多媒体	372
11.1 滚动的消息	372
本节实例讲述了用 Java 实现动画程序的基本技术, 实现了一个具有滚动效果的文本消息, 并使用了双缓冲技术尽量减少屏幕的闪烁。	
11.2 三维弹球	375
本节实例讲述了用 Java 实现 3D 动画的基本技术, 实现了一个跳动的球, 在触地的一瞬间, 球还有被压扁的现象, 按 a 键将球左移, 按 s 键将球右移。	
11.3 贪吃蛇游戏	379
本节实例实现了贪吃蛇游戏, 蛇在不断地运动, 蛇体不断地变长, 食物位置不断地变化, 这是一种动画效果。通过键盘可以控制蛇的运行方向和速度, 可以暂停和恢复游戏。当蛇吃掉一个食物时, 蛇体的长度增加, 每吃到一个食物, 根据得分规则给玩家加分。当蛇头碰到边界或者碰到蛇体时, 游戏结束。实例中主要使用了 MVC 的设计模式。	
11.4 Java 声音处理	391
本节实例讲述了 Java 的声音处理技术, 包括实现响铃 (蜂鸣), 播放 wav、au 等音频文件, 以及控制声音的大小和声道等。	
11.5 媒体播放器	400
本节实例讲述了 JMF (Java 媒体框架) 技术, 实现一个媒体播放器, 能够处理音频和视频, 包括 mp3, avi 等常见的多媒体文件。	

第 12 章 反射	406
12.1 instanceof 操作符	406
本节实例讲述了用 instanceof 操作符鉴别对象类型的技术, 重点介绍了两个对象在用 instanceof 操作符做比较时, 返回 true 的条件。	
12.2 获取类的信息	407
本节实例讲述了用反射机制获取类的基本信息的技术, 包括类声明的属性、构造方法和非构造方法。在文本框中输入类名, 单击“查看类信息”按钮, 将把类的信息显示在一个文本域中。	
12.3 动态调用类的方法	413
本节实例讲述了用反射机制调用类方法的技术, 包括动态生成类的实例, 调用类的静态方法和实例方法, 有参数的方法和无参数的方法。	
第 13 章 网络编程	418
13.1 获取 URL 的信息	418
本节实例讲述了统一资源定位符 (URL), 演示了如何从 URL 中提取信息, 包括 URL 的协议、主机地址、端口和它所引用的资源的信息。	
13.2 Web 浏览器	420
本节实例实现了一个简单的 Web 浏览器, 功能类似于 Windows 的 Internet Explorer, 能显示 HTML 网页、前进、后退、刷新、访问主页和新建窗口。实例中主要使用了 javax.swing.JEditorPane 控件。	
13.3 获取 IP 地址和域名	430
本节实例讲述了获取计算机 IP 地址和域名的技术, 能够获取本机的 IP 地址、机器名, 以及远程服务器的 IP 地址。实例中主要使用了 java.net.InetAddress 类。	
13.4 HTTP 客户端	432
本节实例实现了一个访问 HTTP 服务器的客户端, 以命令行的方式启动, 用户需要指定 HTTP 服务器的地址 (URL), 能得到该 URL 对应网页的源代码。实例中主要使用了 java.net.URL 类的 openConnection 和 getInputStream 方法。	
13.5 基本的 Socket 编程	434
本节实例介绍了 Java 的 Socket 编程技术, 实现了一个最简单的 Socket 服务器和客户端, 客户端通过 Socket 发送信息给服务器, 然后服务器返回消息给客户端。服务器端用 java.net.ServerSocket 类实现, 而客户端使用 java.net.Socket 类实现。	
13.6 HTTP 服务器	439
本节实例实现了一个 HTTP 服务器, 能够接收 HTTP 请求, 并且能同时处理多个请求。客户端可以是一般的浏览器, 如 Windows 的 Internet Explorer, 也可以是自定义的 HTTP 客户端程序。实例中主要使用了 Socket 编程技术。	
13.7 一个支持多线程的服务器框架	446
本节实例实现了一个服务器框架, 在该框架下, 能够在不同的端口上部署各类服务, 客户端通过 Socket 访问服务。支持多个客户端同时访问服务, 为了控制服务器框架, 还实现了一个控制服务, 能够动态地部署、卸载服务器框架上服务, 也可以通过控制服务改变服务器框架支持的最大并发请求数。实例中主要使用了 Socket 编程技术。	
13.8 代理服务器	460
本节实例在 13.7 节实现的服务器框架基础上, 实现了一个代理服务, 将代理服务部署在服务器框架中。用户需要指定目标服务器的 IP 地址和端口, 然后在浏览器中输	