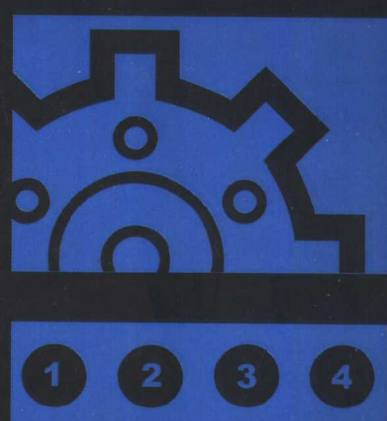
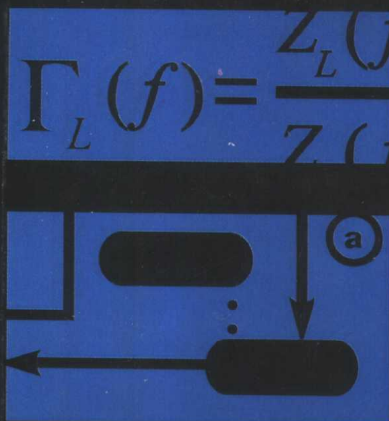


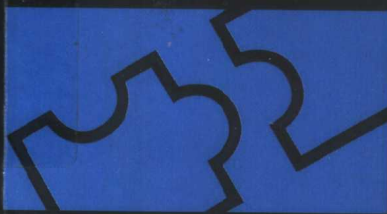
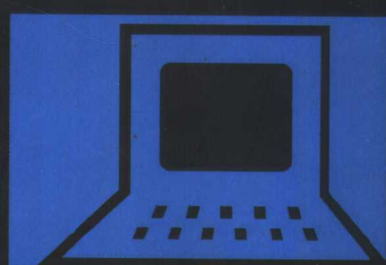
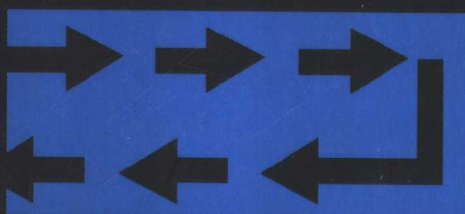
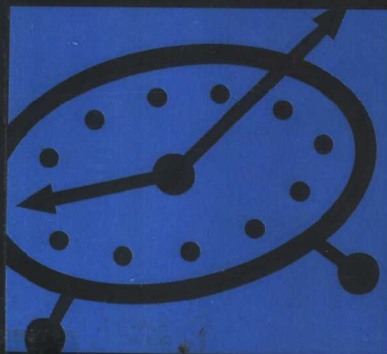


(美)Chuck Cavaness Geoff Friesen Brian Keeton 著

Java

完全探索 (第2版)

Special Edition Using Java 2 Standard Edition



```
let ntimes=0;while (( ntimes  
50 )); do print -u4 'p lbo  
# Send command to coproc  
s read -ru3 time bo  
# Read output from coproc  
print $tim ebolt >> times  
let ntimes=ntimes+1 done
```



中国青年出版社

que®

Java

完全探索 (第2版)

Special Edition Using Java 2 Standard Edition

(美) Chuck Cavaness, Geoff Friesen, Brian Keeton 著
师夷工作室 译



中国青年出版社
CHINA YOUTH PRESS

(京)新登字 083 号

本书简体中文版由 QUE 公司授权中国青年出版社独家出版。未经出版者书面许可,任何单位和个人均不得以任何形式复制或传播本书的部分或全部。

Authorized translation from the English language edition, entitled Special Edition Using Java 2, Standard Edition, 2nd Edition, published by Que

Copyright©2001

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Chinese Simplified language edition published by China Youth Press

Copyright©2001

版权贸易合同登记号: 01-2001-0861

策 划: 胡守文

王修文

郭 光

责任编辑: 江 颖

朱新媛

责任校对: 肖新民

书 名: 《Java 完全探索》(第 2 版)

编 著: (美) Chuck Cavaness Geoff Friesen Brian Keeton

出版发行: 中国青年出版社

地址: 北京市东四十二条 21 号 邮政编码: 100708

电话: (010) 84015588 传真: (010) 64053266

印 刷: 山东高唐印刷有限责任公司

开 本: 787mm × 1092mm 1/16

版 次: 2001 年 9 月北京第 1 版

印 次: 2001 年 9 月第 1 次印刷

印 数: 1-5000

书 号: ISBN 7-5006-4529-5/TP · 215

定 价: 89.00 元

译者序

近年来全球信息化进程明显加速，特别是 Internet 技术的迅猛发展，使世界正进入以计算机网络为中心的计算机时代。人们利用 Internet 这个全球最大的网络即时直接交流，共享网络上丰富的信息资源。目前，我国也已有成千上万的用户入网。怎样更加充分、合理、安全地利用 Internet 上的大量信息资源，已经引起了社会的普遍重视。

1995 年出现的 Java，具有简单、独立于平台、面向对象、分布式、可靠性、安全性、可移植性、动态性、多线程等特性，正符合 Internet 的发展。Java 语言自 1995 年推出的第一个版本以来，经过了几年的磨练，现已成为分布式软件开发的主流。随着 B2B、B2C、C2C 电子商务的迅速崛起，随着分布式计算模式的广泛采用，Java 将越来越被行家看好。目前，Java 语言发展的速度简直超乎人们的想象，Java 语言编写的应用程序已经在各行各业广泛使用，而 Java Applet 则直接用于网站的建立。Java 2 平台的第一次引入伴随着 JDK 1.2 版的发布，接着随着 JDK 1.3 版的发布 Java 2 平台又被升级，这样 Java 技术日臻完美。

本书按照最新公布的 Java 2 SDK 1.3 标准规范，深入地介绍了 Java 技术的基本概念和基本设计思想、Java 核心、输入/输出机制、数据库访问特性、Java Bean 等内容。该书始终围绕着工程实例，深入浅出地展示了采用 Java 语言开发应用程序的基本步骤和各种实现方法，作者以独到的眼光和深厚的功底向我们揭示了 Java 内在的多彩世界。它不仅是一本教材，还是一本工具书。无论你是初次接触 Java，还是有丰富编程经验的程序员，这本书都会使你受益匪浅。

本书是一本最新的 Java 语言参考书，内容相当丰富，包括了 Java 语言的方方面面。能够把这样一部优秀的著作准确、及时地翻译出来，奉献给中国的 Java 程序员们，这是我们的荣幸。我们也深知翻译这样一部大部头的书，对我们是一个挑战。我们全体参加翻译的同仁齐心协力，密切合作，终于完成了这一艰巨的工作。

本书由陈建荣、王俊、王乐春、齐宁等组织翻译，前导工作室全体工作人员共同完成了本书的翻译、录排、校对等工作。全书最后由肖国尊统稿。由于时间仓促，且译者的水平有限，在翻译过程中难免会出现一些错误，请读者批评指正。

但愿我们的工作会给所有 Java 程序员带来方便。

译者

2001 年 4 月

作者简介

Chuck Cavanness 是 NetVendor 公司资深的 Java 设计者，该公司是提供 B2B 产品的 Internet 公司。他的研究方向是 J2EE 的体系结构设计以及构造 B2C 和 B2B 的 Internet 解决方案。他曾经开发了多个 CORBA 和 EJB 的多线程 Internet 解决方案，还包括使用 JDBC 开发了多个关系对象框架。该作者在许多 Java 技术领域获得了认证，并且他是开发互联网 Internet Auction 组件的主要成员之一。

Geoff Friesen 他主研 Java 技术。他已经为 JavaWorld 撰写了大量的关于 Java 的文章，包括国际化到未来的 Java Plug-In。该作者还出版了介绍性的《Java 2 by Example》一书。此外，他还在大学里讲授 Java 课程。目前 Geoff 正在 Itword.com 机构的 Java Beginner Discussion Group 中从事考试监督员的工作。

Brian Keeton 是 NetVendor 公司的软件工程师。他在该公司使用 EJB 开发了支持 B2B 交易的服务方 Java 组件。他是 Sun 认证的 Java 程序员，并且具有 10 年的软件开发经验。他曾经使用 C++ 为国防工业开发了 5 年的面向对象的应用程序，然后转而使用 Swing、CORBA 和 EJB 开发 Java 商业应用。

内容提要

本书是关于 Java 语言的权威参考书，内容涵盖了 Java 语言的方方面面。全书共分为五个部分。

在第一部分中，本书首先介绍了面向对象语言的设计方法和一些基本概念，使读者对 Java 语言的编程模式有了整体性的了解；接着以一个简单的 Hello World 程序为例子，使读者在整体上对 Java 编程有所了解；之后介绍了 Java 的基本数据类型、基本数据结构、基本类、基本接口和基本方法、表达式的使用方法以及线程的概念和使用方法等。

本书的第二部分对 Java 语言提供的丰富的图形显示方法集作了详尽的介绍。首先介绍了 Java 基础类，接着介绍了 AWT 的使用方法，然后用两个创建图形用户界面的例子来具体说明，随之还介绍了 Java 语言的图像处理方法和用户与图形界面的交互。

本书第三部分介绍了 Java 语言的输入/输出方法。在本书的第四部分和第五部分分别介绍了 Java 应用程序和数据库的交互与反射机制，以及 JavaBean 等内容。

对所有 Java 程序员而言，本书都是不容错过的最新、最详尽的参考书。

本书所面向的读者

如果你刚开始接触 Java，那么本书将特别适合你。本书包含大量关于 Java 2 Standard Edition (J2SE) 的信息，各章节的设计便于用户学习，可以从头学至尾。

如果你已经是一位 Java 专家了，那么本书应当是你需要经常翻阅的参考书。本书是一本完全的参考手册，它包含 J2SE 每个方面的一整套示例。在本书涵盖的内容中，你将会找到使你的编程工作变得非常容易的例子和解释。

如果你对编程一无所知，那么对你而言，本书可能会相对过早地进入了编程环境。在你购买本书前，请先阅读第 1 章“面向对象程序设计”和第 2 章“Hello World：第一个 Java 程序”。如果你对于这些内容感到很吃力，那么你也许应该先阅读其他的 Java 指南类书籍。

本书包含 JAVA 2 SDK Standard Edition Version 1.3

Java 2 Platform 是由 JDK 1.2 首先提出和引入的，后来它升级为 JAVA SDK Version 1.3。Java 2 支持这两个版本。作者特别强调指出本书中包含 JAVA SDK Version 1.3 是为了使读者能够了解最新的 Java 2 Platform。

本书讲述的 Java 2 Standard Edition 是语言的核心。Sun 公司同时提出了 Java Enterprise Edition (J2EE) 和 Java 2 Micro Edition (J2ME)，其中 J2EE 使用了 Servlets、Java Server Pages (JSP) 和 Enterprise JavaBeans (EJB) 等技术，它支持分布式的应用开发，而 J2ME 支持为便携式设备开发精巧的嵌入式程序。

引言

本引言内容:

- 本书面向的读者
- 本书结构
- 本书约定

欢迎进入神奇而动态的 Java 世界!如果你刚开始接触 Java,那么我们已准备好为你全面介绍业界最热门的编程技术。Java 是一门内容特别丰富的语言,同时它又是一种相对简单和容易学习的语言,特别是与那些发明家创造的语言相比,该语言具有其独特的魅力。即使是最复杂的任务,Java 通过它强大的 API 也将给与空前支持。

Java 是什么?Java 是革命性的编程语言,它是由 Sun Microsystems 在 1995 年 6 月推出的。从那时起,数以万计的程序员拿起了像本书一样的 Java 工具书,并认识到该语言的强大魅力。

Java 是一种面向对象的编程语言,这就意味着程序员可以使用 Java 方便地开发复杂的、但同时又是可维护的程序,这要比程序员使用早期语言开发同样的程序容易得多。此外,Java 内部支持线程、网络和大量的工具。这些工具允许用户更关注自身程序的需求而更少地考虑低层的结构问题。

关于版本问题:本书支持 Java 2 Platform,Java 2 Platform 是第一个支持公布的 SDK1.2 的版本,SDK1.2 很快升级到 Java™ 2 SDK——版本 1.3。Java 2 同时支持这两个 SDK 版本。本书强调指出其中包含在 JAVA SDK 版本 1.3 中的新增内容,目的是为了使读者了解最新的 Java 2 Platform。

注意:

在本书中,有些参考来自于 SDK,也有一些来自于 JDK。Sun Microsystem 最初推出的平台称为 Java Development Kit-SDK。然而,在 Sun 市场的作用下,随着版本 1.2 的推出,由于有了更通用的名字 Software Development Kit-SDK, Sun 不再使用 JDK。关于该名字的详细解释以及它如何对 Java 平台的各部分产生影响,请查看 Sun 公司站点的市场信息,其 URL 为: <http://java.sun.com/products/jdk/1.2/java2.html>

Sun 公司 Java 打包的另外一个变化是引入了语言的多版本概念。所有的版本都基于相同的核心 API,并且它们之间不是竞争的关系,它们允许开发者关注能更有力地支持其特定需求的技术。本书讲述 Java 2 Standard Edition (J2SE),它是该语言的核心。每一个 Java 开发者都要利用 J2SE 的功能,并且它定义了学习该语言的起点。Sun 公司也提出了 Java Enterprise Edition (J2EE) 和 Java 2 Micro Edition (J2ME),其中 J2EE 使用了 Servlets、Java Server Pages

(JSP) 和 Enterprise JavaBeans (EJB) 等技术, 它支持分布式的应用开发; 而 J2ME 支持为便携式设备开发精巧的嵌入式程序。

本书面向的读者

如果你刚开始接触 Java, 那么本书会特别适合你。不要被本书的厚度吓倒。其中包含大量丰富的 Standard Edition Java 编程语言的信息, 章节的设计便于用户学习, 你可以从书的开头学习 Java。如果你已经是一个 Java 专家了, 那么它也是你书架上经常要翻阅的参考书。本书是完全的参考手册, 并且包含 J2SE 每个方面的一整套示例。在本书涵盖的内容中, 你将会找到使编程工作变得非常容易的例子和解释。

本书结构

本书由六部分组成。每一部分包含大量的相关主题。当你阅读本书时, 这些组织起来的主题将大大利于你学习 Java 语言。

第一部分, “Java 语言”。该部分将讲述 Java 的语法和如何创建正确组织的类和接口。在 Java 语言语法中, 可以找到 Java 的基础内容。任何程序都是建立在语言的基础上。本部分将讲述 Java 语言的每一方面的内容。对于初学者来说, 所有章节的组织有助于熟悉 Java 编程。对于专家来说, Java 语言的每一个独立方面都有详细描述, 因此本书第一部分既是学习工具又是一部详细的参考手册。本部分还包括关于 Java 类集框架和该语言支持多线程应用的高级讨论。

第二部分, “用户界面”。该部分讲述了使用 Java 建立一个用户图形界面的详细内容。讲述内容包括 AWT 和 Swing 等, 此外还探讨了 JFC API。第二部分探讨了 JFC 的辅助功能、Java 2D 以及拖放 API。而且, 由于 Java Media Framework 与 JFC 非常接近, 所以把该部分的内容归入了第二部分。

第三部分, “I/O”。该部分包含了 Java 应用中的所有读数据和写入数据的内容。本部分以讲述基本组件、基本流式文件技术和读文件技术作为开始, 然后学习如何使用 Java 的串行化技术 (serialization) 发送并取回整个 Java 对象。本部分还包括使用 Java 建立网络应用和国际应用的信息

第四部分, “数据库”。该部分提供了建立现代商业应用的最重要方面之一——“数据库”的细节信息。数据库几乎是所有商业应用的核心, 并 Java 的 JDBC 可以非常轻松地解决 Java 应用与数据库之间的通信负担问题。在本部分中介绍了 Java 如何在较短的时间内开发出功能强大具有可移植性的访问, 以及如何使用数据库的应用。欢迎进入独立于平台和独立于 DBMS 系统的世界!

第五部分, “组件开发”。该部分讲述了如何使开发周期更迅速且更容易。基于组件的开发已经提出许多年了, 但从来没有象在 Java 中这样容易实现。在本部分中, 读者将学习如何

使用 Java 中独立于平台的组件模型 **JavaBeans**。讲述该内容以前还将学习 **Reflection API**，该内容支持 **JavaBeans**，并且支持其他的类发现，在 Java 应用运行时这些类发现也许会用到。

附录。当读者需要建立并运行 Java 开发环境以及开始编写自己的程序时，附录将给出所有细节说明。当你实地使用前面的例子时，如果编译和运行那些例子代码遇到了困难，那么你最好是先阅读附录。这部分还包括对 **SDK 1.3** 性能提高的介绍。

本书约定

为了便于用户使用，本书使用了不同的排版格式和排版风格。

注意

本书中的“注意：”将给出一些附加信息，这些信息也许会帮助读者避免问题的发生，或是一些用来描述特征的信息。

提示

提示指出执行一个过程时简单的或者可替换的方法。提示介绍的技术是有经验的开发者简化问题时使用的技术，以及他们在开发坚固的和可维护的系统时使用的设计技术和实现技术。

警告

警告是告诉读者那些危险的做法（例如，删除文件的行为）

新特性 1.3

“新特性”用来描述在 SDK 1.3 中引入的 Java 2 特征和增强功能。

本书中使用了交叉引用，这样可以帮助读者参考该书其他部分的相关信息。

→ 参见第 9 章“接口”

故障排除向导

出口每一章的结尾都有“故障排除”一节，它给出针对关于某些特定主题中也许会产生普遍问题的解决方法。在每章正文主体部分，象这样的交叉引用可以直接地告诉读者关于该问题的解决办法在“故障排除”中的对应标题。

目 录

第一部分 Java 语言

第 1 章 面向对象程序设计

1.1 面向对象程序设计：一种新的思维方式	3
1.2 程序设计简史	3
1.2.1 过程式语言	4
1.2.2 结构化开发	4
1.2.3 面向对象程序设计	5
1.3 关于对象	6
1.3.1 传统的程序设计	6
1.3.2 OOP 方法	7
1.3.3 用继承机制扩展对象	7
1.4 多实体对象	9
1.5 模块化与代码组织	10
1.6 将对象关联到 Java 类	10
1.7 构建层次结构：OOP 设计步骤	10
1.7.1 将问题分解成多个实体	10
1.7.2 在实体间寻找共同点	11
1.7.3 在实体间寻找差异	11
1.7.4 用抽象和继承机制设计层次结构	11
1.8 OOP 与 UML	12
1.8.1 UML 简介	13
1.8.2 类图	13
1.8.3 顺序图	16
1.8.4 UML 指导思想	17
1.9 Java 是充满魔力的 OOP 语言吗？	18

第 2 章 HelloWorld：第一个 Java 程序

2.1 理解 HelloWorld 程序	19
2.1.1 创建文件	19
2.1.2 编译代码	20
2.1.3 运行程序	20
2.2 理解 HelloWorld	20
2.2.1 注释	20
2.2.2 声明类	21
2.2.3 main 方法	21
2.2.4 显示到屏幕上	22
2.2.5 System.out 和 System.in	22
2.3 作为 applet 的 HelloWorld	25
2.3.1 修改并编译源代码	26
2.3.2 创建 HTML 文件	26
2.3.3 在 AppletViewer 中运行程序	27
2.3.4 在 Internet Explorer 中运行 Hello World	27
2.3.5 导入其他类	28
2.3.6 声明一个 applet 类	28
2.3.7 Applet 的 paint 方法	28
2.3.8 applet 短暂的生命	29
2.4 保留字	29
2.5 Java 1.3 API	30
2.6 故障排除	30

第 3 章 数据类型与其他标记

3.1 Java 数据类型	33
3.1.1 原始类型	33
3.1.2 引用类型	34

3.2 使用变量.....35	4.2.2 作用域.....67
3.2.1 声明变量.....35	4.2.3 局部变量初始化.....67
3.2.2 标识符：对变量的命名.....35	4.3 方法和 UML 顺序图.....67
3.3 布尔原始类型.....36	4.4 捕获并抛出异常.....68
3.4 整数类型.....37	4.4.1 使用 try/catch/finally.....68
3.4.1 整数值界限.....38	4.4.2 抛出异常.....70
3.4.2 声明整型变量.....38	4.4.3 组合方法.....71
3.4.3 整数运算.....39	4.4.4 异常的类型.....71
3.5 运算符.....39	4.4.5 处理多个异常.....73
3.5.1 算术运算符.....39	4.4.6 创建自己的异常类.....74
3.5.2 赋值运算符.....40	4.4.7 异常与返回值.....75
3.5.3 增量/减量运算符.....41	4.5 Java 的错误类.....75
3.6 字符变量.....42	4.6 故障排除.....76
3.7 浮点变量.....42	第 5 章 使用表达式
3.8 直接量：赋值.....43	5.1 什么是表达式？.....79
3.9 整型直接量.....44	5.2 表达式怎样求值.....80
3.10 字符型直接量.....45	5.2.1 运算符的结合性.....80
3.11 浮点型直接量.....47	5.2.2 运算符的优先顺序.....81
3.12 字符串直接量.....48	5.2.3 求值顺序.....82
3.13 数组.....48	5.3 C 程序员应特别注意的地方.....83
3.14 不使用标记的输入元素.....51	5.4 按位运算符.....84
3.14.1 空格（whitespace）.....51	5.5 移位运算符.....87
3.14.2 注释.....52	5.5.1 左移.....87
3.15 故障排除.....54	5.5.2 有符号右移.....88
第 4 章 方法与异常	5.5.3 无符号右移.....88
4.1 方法声明.....55	5.6 类型转换.....88
4.1.1 访问修饰符.....56	5.6.1 隐式转换.....89
4.1.2 修饰符（modifiers）.....58	5.6.2 使用类型转换运算符的显式转换.....90
4.1.3 返回信息.....61	5.6.3 字符转换.....91
4.1.4 方法名.....61	5.7 支持字符串的特殊运算符.....91
4.1.5 参数.....61	5.8 故障排除.....91
4.1.6 异常.....65	第 6 章 控制流
4.2 块与语句.....66	6.1 控制运行流程.....93
4.2.1 标号语句.....66	

6.2	布尔运算符	93
6.2.1	关系运算符	94
6.2.2	等价运算符	96
6.3	逻辑表达式	97
6.3.1	条件 AND 和条件 OR 运算符	98
6.3.2	逻辑非运算符	99
6.4	条件运算符	99
6.5	控制流语句中的布尔运算	100
6.6	控制流功能	101
6.6.1	if 语句	101
6.6.2	if-else 语句	102
6.7	Switch 语句	103
6.8	循环语句	105
6.8.1	while 循环	106
6.8.2	do 循环	106
6.8.3	for 循环	106
6.9	跳转语句	107
6.9.1	break 语句	108
6.9.2	continue 语句	109
6.9.3	return 语句	109
6.10	故障排除	109

第7章 类

7.1	什么是类?	111
7.2	为什么使用类?	112
7.3	Java 中的类	115
7.4	声明一个类	117
7.4.1	访问修饰符	117
7.4.2	修饰符	118
7.4.3	类名	120
7.4.4	超类——扩展另一个类	120
7.5	变量——定义状态	121
7.5.1	访问修饰符	125
7.5.2	修饰符	126

7.5.3	实例域	127
7.5.4	类域	128
7.5.5	声明一个常量	128
7.5.6	实现一个枚举类型	129
7.6	方法——定义行为	130
7.6.1	构造器	130
7.6.2	重载方法	132
7.6.3	用方法来提供 guarded 访问	134
7.6.4	类方法	134
7.7	创建类的一个实例	135
7.7.1	new 运算符	135
7.7.2	垃圾回收	135
7.8	引用类元素	136
7.9	Object 类	136
7.9.1	对象相等	137
7.9.2	拷贝一个类实例	137
7.9.3	在 finalize() 方法中进行清扫工作	138
7.10	类型转换与转换引用类型	139
7.10.1	编译时与运行时的类型检查	139
7.10.2	使用 instanceof 运算符	140
7.11	类属类	140
7.11.1	定义类属类	140
7.11.2	使用类属类的理由	140
7.11.3	类属类如何工作	142
7.12	包	142
7.12.1	使用包来组织代码	142
7.12.2	包与你的文件系统	143
7.12.3	从另一个包中导入一个类	143
7.12.4	导入整个包	144
7.12.5	不经导入而使用类	144
7.12.6	Java.lang 的隐式导入	144

7.13 将原始类型包装在类中	144
7.14 使用标准的数学函数	145
7.15 建立 UML 类图	145
7.16 故障排除	146

第 8 章 字符串和文本的使用

8.1 字符串介绍	147
8.2 使用字符串类	149
8.3 获得字符串对象信息	150
8.4 字符串比较	153
8.5 字符串提取	156
8.6 字符串操作	159
8.7 字符串连接	159
8.8 将对象转换成字符串	160
8.9 将原始类型转换成字符串	162
8.10 使用 StringBuffer 类	162
8.10.1 创建一个 StringBuffer 对 象	163
8.10.2 在 StringBuffer 后面附加 内容	163
8.10.3 转换一个 StringBuffer 对 象为字符串	163
8.10.4 对 StringBuffer 对象进行 操作	164
8.11 使用 StringTokenizer 类	165
从文件中获得所有标记符	166
8.12 字符串和编译器	168
8.13 故障排除	169

第 9 章 接口

9.1 接口是什么?	171
9.1.1 并不是每一个类本身都有 一个接口	172
9.1.2 多接口	172
9.2 定义一个接口	172
9.2.1 声明	174

9.2.2 接口程序体	175
9.2.3 标记接口	177
9.3 实现一个接口	178
9.3.1 方法重载	179
9.3.2 修饰符	179
9.3.3 参数列表	180
9.3.4 程序体	180
9.3.5 异常	180
9.3.6 接口方法冲突	181
9.4 接口引用	182
9.4.1 访问常量	182
9.4.2 设计一个接口胜于实现 一个接口	183
9.5 在一个接口和一个抽象类之 间选择	184
9.6 在 UML 中描述接口	185
9.7 故障排除	185

第 10 章 数据结构与 Java 设施

10.1 Collection 和 Utility 类	187
10.2 Collection 结构	188
10.3 Collection 结构的接口	188
10.3.1 Collection 接口	189
10.3.2 List 接口	191
10.3.3 Set 接口和 SortedSet 接 口	193
10.3.4 Map 接口和 SortedMap 接口	193
10.4 基本类和接口	195
10.4.1 Vector 类	196
10.4.2 ENUMERATION 接口	198
10.4.3 STACK	199
10.4.4 Hashtable	199
10.5 通用实现类	201
10.5.1 ArrayList 类	202

10.5.2	LinkedList 类	203
10.5.3	HashMap	206
10.5.4	TreeMap	208
10.5.5	WeakHashMap	208
10.5.6	HashSet	209
10.5.7	TreeSet	209
10.6	Collection 排序	210
10.6.1	COMPARABLE 接口	210
10.6.2	Comparator 接口	210
10.6.3	Collections.sort()的使用	211
10.7	Collection 结构重述	213
10.7.1	Iterator 接口	213
10.7.2	ListIterator 接口	214
10.7.3	有效搜索	215
10.8	Collection 功能和约束	215
10.8.1	方法和域	215
10.8.2	Singleton 方法	216
10.8.3	同步化的 Collection 结构	216
10.8.4	不可变 Collection 结构	216
10.9	使用哪种 Collection 类?	217
10.10	Array 的功能	217
10.10.1	Array 类	217
10.10.2	搜索 Array	218
10.10.3	两个 Array 的比较	218
10.10.4	填充 Array	218
10.10.5	排序 Array	218
10.11	使用 Properties 类配置应用	
	程序属性	218
10.11.1	设置 Properties 对象	219
10.11.2	查询 Properties 对象	219
10.11.3	存储和获取 Properties 对象	219
10.12	Date 类型	220
10.12.1	Date 类	220

10.12.2	Date 对象比较	221
10.12.3	将 Date 类型转换为 String 类型	221
10.12.4	改变 Date 属性	221
10.13	BitSet 类	221
10.14	随机数的产生	223
10.15	使用 Observer/Observable	
	观察状态变化	224
10.15.1	Observable 类	224
10.15.2	Observer 接口	226
10.16	故障排除	226

第 11 章 线程

11.1	什么是线程?	229
11.2	为什么使用线程?	231
11.3	如何制作线程类?	232
11.4	扩展 Thread	232
11.5	实现 Runnable	233
11.6	线程的生命周期	236
11.7	使用 Thread 定位变量	238
11.8	改变线程优先级	240
11.9	线程同步	243
	线程间通信	247
11.10	改变线程运行状态	249
11.11	线程组	250
11.12	获取当前运行线程数量	253
11.13	线程让位	253
11.14	守护线程	255
11.15	使用 Timer 类和 Timer Task 类	256
11.16	在程序中安全使用线程	258
11.17	故障排除	259

第二部分 用户界面

第 12 章 Java 基础类介绍

12.1 什么是 JFC	263	13.2.5 实用方法	313
12.1.1 AWT	263	13.3 字体和颜色	317
12.1.2 Swing	264	13.3.1 认识字体	317
12.1.3 辅助功能	264	13.3.2 创建字体	318
12.1.4 Java 2D	264	13.3.3 字体度量	320
12.1.5 拖放	265	13.3.4 颜色描述	324
12.2 JFC 的历史	265	13.3.5 系统颜色	326
12.2.1 AWT 1.0	265	13.4 图像	327
12.2.2 Internet 基础类	265	13.4.1 载入和绘制图像	330
12.2.3 应用基础类	266	13.4.2 动画	335
12.2.4 Java 基础类	266	13.4.3 产生图像	346
12.2.5 将来的展望	266	13.4.4 消费图像	350
12.3 JFC 的应用程序	266	13.4.5 过滤图像	357
12.3.1 尝试 JFC 的 AWT 应用 程序	266	13.5 故障排除	362
12.3.2 尝试 JFC 的 Swing 应用 程序	268	第 14 章 使用 AWT 方式创建 GUI	
12.4 Applet	270	14.1 窗口工具集介绍	365
12.4.1 什么是 Applet	270	14.2 使用组件、容器和布置管 理器创建 GUI	366
12.4.2 Applet 和 World Wide Web	270	14.2.1 系统工具集	366
12.4.3 AWT applet	271	14.2.2 对等实体、重量组件和 轻量组件	368
12.4.4 AWT Applet HTML	275	14.2.3 组件研究	369
12.4.5 Swing applet	276	14.2.4 容器研究	388
12.4.6 Swing Applet HTML	279	14.2.5 布局管理器研究	401
12.4.7 体系结构	281	14.3 使用事件和监听器创建 GUI	411
12.4.8 Applet 和应用程序	285	14.3.1 事件研究	411
12.5 故障排除	292	14.3.2 监听器研究	415
第 13 章 AWT 基础		14.4 定制窗口工具集	431
13.1 什么是 AWT	295	14.4.1 创建自己的容器	432
13.2 图形	296	14.4.2 创建自己的布局管理器	432
13.2.1 画图的基础	296	14.5 故障排除	435
13.2.2 显示文本	297	第 15 章 Swing 简介	
13.2.3 绘制形状	299	15.1 什么是 Swing	439
13.2.4 实用类和接口	309		

15.2	比较 Swing 和 AWT.....	441	17.3.6	Monkey	549
15.3	混合重量和轻量的组件	444	17.4	Robot.....	550
15.4	从 AWT 到 Swing 的转换.....	450	17.5	故障排除	552
15.5	故障排除	451	第 18 章 Java 2D		
第 16 章 使用 Swing 方式创建 GUI			18.1	什么是 Java 2D	553
16.1	Swing 工具集	453	18.1.1	Graphics2D	553
16.1.1	模型视图控件器 (Model-View-Controller) 结构	453	18.1.2	坐标系统	553
16.1.2	可插入的外观和感觉	455	18.1.3	环境、设备和配置	554
16.2	使用组件、容器和布置管理器创建 GUI	459	18.1.4	多屏幕环境 (Multiple-Screen Environments)	555
16.2.1	研究组件	459	18.1.5	API	558
16.2.2	研究容器	492	18.2	着色 101	560
16.2.3	研究布局管理器	496	18.2.1	着色提示属性	560
16.3	使用事件和监听器创建 GUI	504	18.2.2	Stroke 属性	562
16.4	附加功能	504	18.2.3	Paint 属性	567
16.4.1	行为	504	18.2.4	Transform 属性	569
16.4.2	边	508	18.2.5	裁剪路径属性	573
16.4.3	键盘输入和捆绑	510	18.2.6	合成属性	574
16.4.4	绘制	515	18.3	图像、文本和带缓冲的图像 (Shapes、Text And Buffered Images)	578
16.4.5	属性	516	18.3.1	图形 (Shapes)	578
16.4.6	线程问题	516	18.3.2	文本 (Text)	591
16.4.7	定时器	516	18.3.3	带缓冲区的图像 (Buffered Image)	594
16.5	故障排除	521	18.4	打印	604
第 17 章 辅助功能			18.4.1	打印结构 (The Printing Framework)	605
17.1	什么是辅助功能	523	18.4.2	工作控制和提交 (Job Control And Rendering)	605
17.2	辅助功能是如何工作的	524	18.4.3	操纵 Printable 类型打印 (Playing With Printable)	608
17.3	实用工具 (Utilities)	546	18.4.4	操纵 Pageable 类型打印 (Playing With Pageable)	610
17.3.1	AccessibilityMonitor	546	18.4.5	快速打印和脏打印 (Quick	
17.3.2	AWTMonitor	547			
17.3.3	Ferret	547			
17.3.4	JavaMonitor	548			
17.3.5	Linker	549			

And Dirty Printing)	613
18.5 故障排除	614
第 19 章 拖放	
19.1 什么是拖放	617
19.2 数据传送	617
19.2.1 Data Flavors	618
19.2.2 Transferables	622
19.2.3 剪贴板拥有者	623
19.2.4 剪贴板	624
19.3 拖放	638
19.4 故障排除	649
第 20 章 Java 媒体框架	
20.1 什么是 JMF	651
20.1.1 JMF 的下载与安装	651
20.1.2 API	652
20.1.3 分层的体系结构	652
20.1.4 媒体流	653
20.1.5 时基和时钟	653
20.1.6 管理器	654
20.2 播放媒体	655
20.3 处理媒体	664
20.4 捕捉媒体	671
20.5 故障排除	678

第三部分 I/O

第 21 章 流、文件和串行化

21.1 什么是流	681
21.2 使用流类	682
21.3 用字节流读写数据	682
21.3.1 InputStream 类	683
21.3.2 OutputStream 类	685
21.3.3 读写字节数组	686
21.3.4 读写文件	686

21.3.5 流缓冲	686
21.3.6 过滤流	687
21.3.7 打印流	687
21.4 重定向标准输入 / 输出	688
21.5 使用 Reader 和 Writer	689
21.5.1 使用 BufferedReader 和 BufferedWriter	689
21.5.2 使用 LineNumberReader	690
21.5.3 使用 InputStreamReader 和 OutputStreamReader	691
21.5.4 PrintWriter 类	691
21.6 有关文件的工作	692
21.7 创建文件	693
创建临时文件	694
21.8 读写文件	695
21.8.1 随机文件访问	696
21.8.2 文件的安全性	698
21.9 目录操作	698
退出时删除文件	699
21.10 对象流	700
21.11 故障排除	703

第 22 章 对象串行化

22.1 什么是对象串行化	705
22.1.1 对象串行化如何工作	706
22.1.2 用对象引用处理对象	706
22.2 对象串行化的例子	707
22.3 读写自己的对象	710
22.4 定制对象串行化	712
使用 Transient 关键字	716
22.5 使用 Externalizable 接口	719
22.6 故障排除	719

第 23 章 通信和组网

23.1 网络通信概述	721
23.2 TCP/IP 体系结构	722