

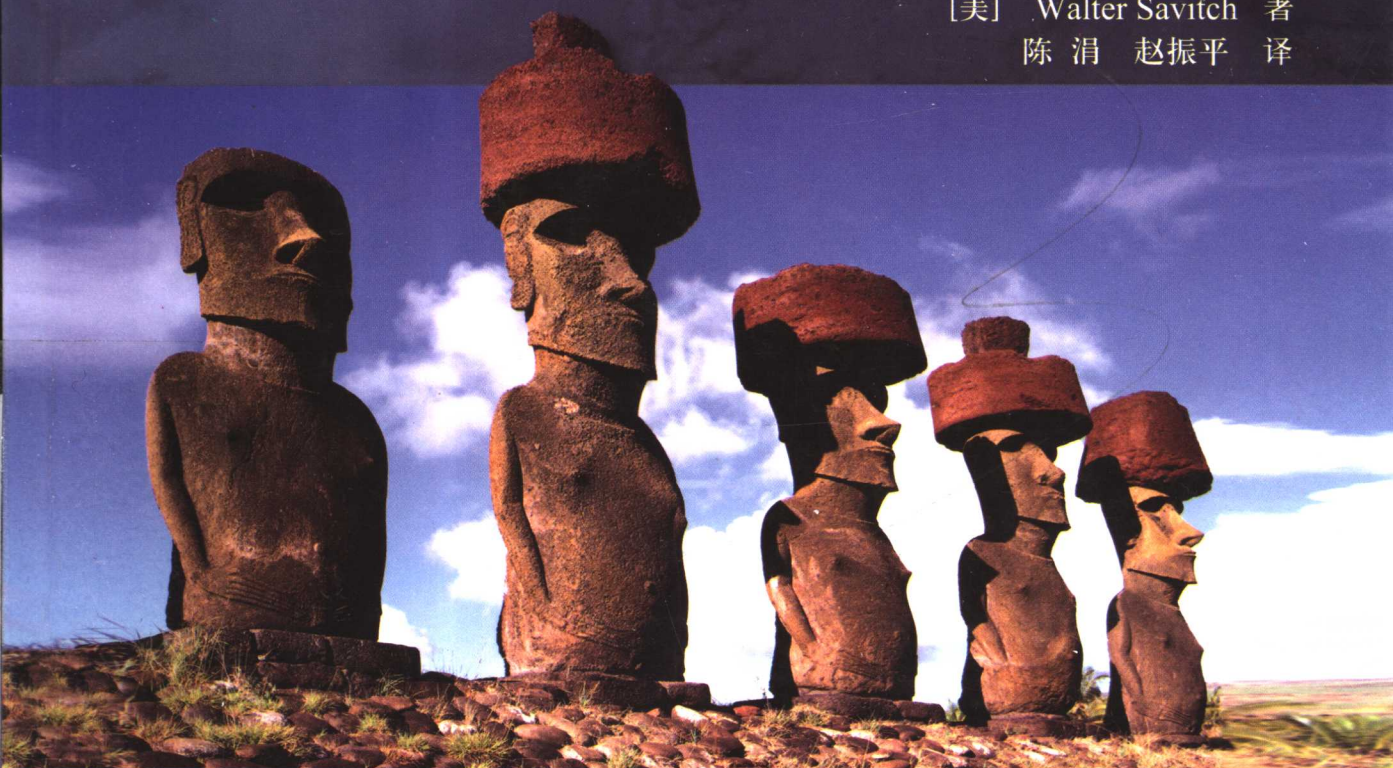
Java程序设计与问题解决

高级篇 (第4版)

Java: An Introduction to Problem Solving and Programming

Fourth Edition

[美] Walter Savitch 著
陈涓 赵振平 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵计算机科学丛书

TP312/2618

:2

2007

Java程序设计与问题解决

高级篇 (第4版)

Java: An Introduction to Problem Solving and Programming

Fourth Edition

[美] Walter Savitch 著
陈涓 赵振平 译



人民邮电出版社
北京

图书在版编目 (CIP) 数据

Java 程序设计与问题解决: 第 4 版. 高级篇 / (美) 萨维奇 (Savitch, W.) 著; 陈涓, 赵振平译. —北京: 人民邮电出版社, 2007.12

(图灵计算机科学丛书)

ISBN 978-7-115-16922-8

I. J… II. ①萨…②陈…③赵… III. JAVA 程序—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 151549 号

内 容 提 要

本书是国外高校采用率很高的Java程序设计经典教材。书中不但讲解了Java的知识, 还提供了一些编程技巧。本书主要讲述高级内容, 包括异常处理、流和文件I/O、动态数据结构与泛型、递归等高级主题。本书使用了J2SE 5.0的最新特性, 并且包含新的、可选讲的图形编程内容和贯穿全书的GUI材料。

本书可作为高等院校计算机及相关专业本科生或研究生程序设计课程的教材, 也适合相关技术人员作为技术提高之用。

图灵计算机科学丛书

Java 程序设计与问题解决: 高级篇 (第 4 版)

- ◆ 著 [美] Walter Savitch
- 译 陈 涓 赵振平
- 责任编辑 杨海玲 刘 静
- ◆ 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>
- 北京鸿佳印刷厂印刷
- 新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
- 印张: 21.25
- 字数: 517 千字 2007 年 12 月第 1 版
- 印数: 1-4 000 册 2007 年 12 月北京第 1 次印刷

著作权合同登记号 图字: 01-2007-2669 号

ISBN 978-7-115-16922-8/TP

定价: 49.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

写给教师的前言

本书是为程序设计及计算机科学方面的第一门课程而撰写的。书中涵盖了一些编程技巧，以及Java程序设计语言的基本概念。适用于短到半学年的课程，或长到一个学年的课程。学生不需要事先具备编程经验，除了一点代数知识外，也不需要更多的数学知识。本书也可以在为那些已经学过其他程序设计课程的学生开设的Java课程中使用。

本书只使用了Java中的标准类（这些类是Java的一部分），不需要额外的类。

本书所有代码都用Sun公司的Java JDK 5.0 beta 2版测试过。为了保持与本书的全面兼容，使用的Java必须是5.0版或更高的版本^①。

本版中的修改

如果没有用过本书的第3版，就可以跳过本节。如果用过第3版，本节会告诉你第4版和第3版有哪些不同。

对教师来说，从第3版到第4版的转换是很容易的。在讲授相同的课程时，可以以相同的顺序讲授基本上相同的主题，只是在涵盖的材料上有极少量的修改。这个版本最大的修改就是用Java 5.0版中提供的新的Scanner类取代SavitchIn类^②。如果想对课程做进一步的修改，本版中还包含了图形编程补充小节，可以选择尽早开始讲述图形编程的内容。本版还添加了对带有类型参数的泛型编程概念的介绍。

第1章和第2章^③都以一个图形编程补充小节结束，这一小节涵盖了使用applet以及JFrame的图形应用程序和GUI。这些图形编程补充小节是可选的。如果在第2章之后，希望获得更多有关图形和GUI的知识，可以直接跳到第5章去，从该章开始对Swing进行了详细的讨论。

本版中其他的重要修改都是为了与Java 5.0版相匹配：为键盘输入使用了新的Scanner类；解释并使用了自动装箱和拆箱；对向量内容进行升级，以使用泛型类型参数。第3章给出了对类型参数和泛型的介绍。

最新的Java特性

本版已经进行了更新，使用了Java 5.0版的最新特性。特别地，为键盘输入使用了新的Scanner类，涵盖并使用了自动装箱和拆箱，用类型参数使用向量，还介绍了使用类型参数的泛型编程。编写本书时，我们用Sun公司的Java 5.0 beta 2版检测了所有代码。

① Sun公司修改了它的版本编号方式。5.0版原来被称为1.5版，在某些地方可能会发现它还被称为1.5版。

② SavitchIn类在附录中提供，而且源代码也可以从网上下载。但是，在本书的正文中没有使用SavitchIn。

③ 本书主要包含Java™: *An Introduction to Problem Solving and Programming, Fourth Edition*的第8章~第14章的内容，其第1章~第7章的内容包含在本书的姊妹篇《Java程序设计与问题解决：基础篇（第4版）》中，附录及索引内容可从图灵网站（www.turingbook.com）下载。——编者注

灵活性

如果你是一名教师，本书会适应你的教学方法，而不需要你来适应本书。本书没有严格规定所授课程必须涵盖的主题之间的次序，可以很容易地改变所要讲授的章节间的次序。每章开始的“预备知识”部分对重新组织材料的详情进行了解释。

本书没有使用专门的库，只使用标准Java库中的标准类。

更早的图形编程

本书前两章以可选的图形编程补充小节作为章节的结尾。这样，从课程一开始，你就可以选择是否讲授图形和GUI编程内容。图形编程补充小节的重点是applet，但也涵盖了用JFrame类构建的GUI。你也可以一开始就跳转到讲述GUI编程的主要章节（第5章～第7章）。或者继续讲到第2章的图形编程补充小节，这样就可以继续进行一些结合了图形编程方式和传统编程方式的练习。希望推迟讲授图形编程内容的教师可以推迟讲授或跳过这些图形编程补充小节。

问题求解及编程技术的内容

本书以讲解基本的问题求解方法及编程技术为设计目标，而不仅仅是一本有关Java语法的书。书中包含了大量的案例学习和编程提示，以及很多讲解了重要问题求解方法及编程技术的小节，比如循环设计技术、调试技术、编程风格、抽象数据类型和基本的面向对象编程技术，包括UML和事件驱动编程，以及使用类型参数的泛型编程。

面向对象技术与传统技术

任何实际讲授Java的课程都必须很早就讲授类的概念，因为Java中所有的概念都涉及了类。一个Java程序是一个类，字符串数据类型是一个类，甚至等于运算符（==）的行为都取决于它是在与类中的对象进行比较，还是在与一个较简单的数据进行比较。除非利用那些极长且复杂的“魔法公式”，否则，类是无法避免的。本书很早就引入了类的概念。所有关于类的基本信息，包括继承在内，都在《Java程序设计与问题解决：基础篇》的第7章结束之前进行了交代（即使略过该书的第6章也是如此）。但是，可以将某些与类有关的主题，包括继承，推迟到课程的后期或高级课程中介绍。

尽管本书很早就引入了类的概念，但它并没有忽略自顶向下的设计方法和循环设计技术这样的传统编程技术。这些较老的主题可能不再那么吸引人，但仍然是所有初学者必须掌握的内容。

泛型编程

类型参数的概念是在第3章学习向量时介绍的。在基本向量类型中包含了类型参数的就是最新版本的向量。然后，向学生们展示了如何定义自己的、包含了类型参数的类。

语言细节及示例代码

本书讲授的是编程技术，而不仅仅是Java语言。但是，对于一门不讲授编程语言的入门性编程课程，无论是学生还是教师都不会感到满意的。在你帮助学生克服对语言细节的恐惧之前，通常是无法将他们的注意力集中到更大的问题上的。因此，本书给出了对Java语言特性的完整解释，以及大量的示例代码。书中提供了完整的程序，以及作为示例的输入和输出。在很多情况下，除了正文中那些完整的示例之外，还可以从因特网上获得一些额外的完整示例。

自测题

每章都配有自测题。这些问题的难度级别跨度很大。有些只需要用一句话来回答，而有些则需要读者编写一个完整的、重要的程序。在每章的结尾都给出了所有自测题的完整答案，包括那些需要完整程序的题目。

经过课堂检验

书中所有资料都经过完整的课堂检验。根据检验结果对很多资料和表达方式进行了修改。

配套资料

可以从出版商或者因特网上获取以下的配套资料。

配套资源

在本书的配套网站<http://www.prenhall.com/savitch>上提供了本书的源代码、额外的编程示例以及下载Java编译器和编程环境的链接。

教师资源指南

教师工具中包含各章的教师资源指南，指南中包含了大量的授课提示、带有答案的测试题、很多编程练习的答案、PowerPoint幻灯片以及其他一些授课资源。教师可以与Prentice Hall销售代表联系，以获取访问教师网站的相关信息。可以通过网址<http://www.prenhall.com>与Prentice Hall联系，获取销售代表的名字和编号。

Walter Savitch

wsavitch@ucsd.edu

<http://www.cse.ucsd.edu/users/savitch>

写给学生的前言

本书的目的是讲授Java程序设计语言，不要求学生事先具备编程经验，除了一些简单的代数知识之外，也不需要更多的数学知识。为了最大程度地从本书获益，在机器上应该装有一个可用的Java版本，这样就可以练习书中给出的示例和技巧了。应该安装Java 5.0版（或更高的版本）。（Sun公司修改了它的版本编号方式。5.0版原来被称为1.5版，在某些地方可能会发现它仍然被称为1.5版。如果有一个名为“1.5版”的Java的副本，也应该可以。）

如果以前编写过程序

使用本书时，不要假设Java和习惯使用的编程语言是一样的。所有的语言都是不同的，它们之间的区别即使很小，也足以给你带来一些问题。

如果你以前曾经用C或C++编写过程序，那么，向Java的转换可能会比较麻烦。乍一看，Java好像和C或C++基本上是一样的。但是，Java和这些语言有很大的不同，你必须认识到这些区别。附录K对Java和C++进行了比较，以帮助你了解它们之间的区别。

正文中的程序代码

本书中所有的程序和其他软件示例都可以从本书的网站上下载，这样，不用将这些示例输入计算机，就可以用它们进行练习了。

获取Java环境

在本书的配套网站上有可以下载Java编译器和编程环境的链接。对于初学者来说，推荐使用Sun公司的Java SDK作为Java编译器及相关软件，推荐TextPad作为简单的编辑环境来编写Java代码。下载Java SDK时，一定要下载5.0版或更新的版本。

本书的配套网站

在本书的配套网站<http://www.prenhall.com/savitch>上提供了本书中的源代码、额外的编程示例以及下载Java编译器和编程环境的链接。

自测题

每章中都配有大量的自测题。在每章的结尾都给出了这些问题的完整答案。要对所学的知识进行练习，最好的方法就是在看答案之前做这些自测题。

本书也是一本参考书

除了将本书作为教材之外，还可以并且应该将它当作一本参考书。“快速参考”中会有一个很短的条目，给出了所有与那个主题有关的基本内容。可以用这种方法来查看Java语言及编程技术方面的细节。

每章的“小结”都对该章的主要内容进行了简要的总结。可以根据这些小结中的内容对该章进行复习，或者根据这些内容来查看Java语言的一些细节。

我们期待读者的意见

本书是为你编写的，我们期望听到你对本书的任何评论。你可以用电子邮件通过下面的地址与我联系：

wsavitch@ucsd.edu

遗憾的是，不能提供编程练习的答案。只有采用本书作为教材的教师才可以从出版商那里获取部分答案。要想获取编程练习方面的帮助，只能与你的老师联系。但是，每章的结尾都有所有自测题的答案。

Walter Savitch

<http://www.cse.ucsd.edu/users/savitch>

致 谢

我要感谢我所在的加州大学圣迭戈分校的计算机科学与工程系，本书的很多素材在那里得到了检验。在我授课的班级中，很多学生都帮助我对本书的初稿进行了校对。这些学生的建议和在课堂上试用本书的教师的建议对形成本书的最终版本有着极大的帮助。我要特别感谢加州大学萨克拉门托分校的Carole McNamee和加州大学圣迭戈分校的Paul Kube，他们对本书的早期版本提出了详细的反馈意见，并在课堂进行试用。我还要特别感谢杨百翰大学的Robert Burton，他为本版的草稿准备了详细的分析材料。他们的建议对本书的最终成型提供了巨大的帮助。

我要感谢所有花时间阅读本书早期版本书稿的审阅者。他们提供了非常宝贵的、详细的评论和建议，这个新的版本也继续从这些评论和建议中受益。按照字母顺序，这些审阅者是：

Jim Buffenbarger——爱达荷州立大学 (Idaho State University)
Robert P. Burton——杨百翰大学 (Brigham Young University)
Steve Cater——凯特林大学 (Kettering University)
Martin Chelten——穆尔帕克社区学院 (Moorpark Community College)
Michael Clancy——加州大学伯克利分校 (University of California, Berkeley)
Tom Cortina——纽约州立大学石溪分校 (SUNY, Stony Brook)
Prasun Dewan——北卡罗来纳大学 (University of North Carolina)
Laird Dornan——Sun公司 (Sun Microsystems, Inc.)
H.E. Dunsmore——普度大学拉斐特分校 (Purdue University, Lafayette)
Adel Elmaghraby——路易斯维尔大学 (University of Louisville)
Gobi Gopinath——萨福克县社区学院 (Suffolk County Community College)
Le Gruenwald——俄克拉何马大学 (University of Oklahoma)
Gopal Gupta——得克萨斯大学达拉斯分校 (University of Texas, Dallas)
Ricci Heishman——北弗吉尼亚社区学院 (North Virginia Community College)
Robert Herrmann——Sun公司 (Sun Microsystems, Inc., Java Soft)
Robert Holloway——威斯康星大学麦迪逊分校 (University of Wisconsin, Madison)
Lily Hou——卡内基-梅隆大学 (Carnegie Mellon University)
Rob Kelly——纽约州立大学石溪分校 (SUNY, Stony Brook)
Michele Kleckner——埃隆学院 (Elon College)
Mike Litman——西伊利诺伊大学 (Western Illinois University)

Blayne Mayfield——俄克拉何马州立大学 (Oklahoma State University)

John Motil——加利福尼亚州立大学北岭分校 (California State University, Northridge)

Michael Olan——斯托克顿州立大学 (Stockton State)

James Roberts——卡内基-梅隆大学 (Carnegie Mellon University)

Alan Saleski——罗耀拉大学芝加哥分校 (Loyola University, Chicago)

Nan C. Schaller——罗切斯特理工学院 (Rochester Institute of Technology)

Ryan Shoemaker——Sun公司 (Sun Microsystems, Inc.)

Ken Slonneger——艾奥瓦大学 (University of Iowa)

Donald E. Smith——罗杰斯大学 (Rutgers University)

Boyd Trolinger——巴特学院 (Butte College)

Subramanian Vijayarangam——麻省大学洛厄尔分校 (University of Massachusetts, Lowell)

我还要感谢本版的审阅者们:

Robert Burton——杨百翰大学 (Brigham Young University)

Ed Gellenbeck——华盛顿大学 (Central Washington University)

Anthony Larrain——德保大学 (Depaul University)

Michael Long——加利福尼亚州立大学 (California State University)

Drew McDermott——耶鲁大学 (Yale University)

Ken Slonneger——艾奥瓦大学 (University of Iowa)

Navabi Tadayon——亚利桑那州立大学 (Arizona State University)

Richard Whitehouse——亚利桑那州立大学 (Arizona State University)

Michael Young——俄勒冈大学 (University of Oregon)

我还要感谢Prentice Hall所有负责组织本书审阅和出版工作的人员。我尤其要感谢Toni Holm、Patrick Lindner和Irwin Zucker。我要特别感谢我的出版人Alan Apt, 感谢他在本书的编写和出版过程中提供的非常宝贵的支持和建议。这些优秀的人都做得很棒!

Lew Rakocy编写了教师指南的编程答案, Robert Burton制作了PowerPoint幻灯片。我要感谢他们细心的工作。

我要感谢Sun公司允许我在很多GUI例子中使用Duke图标。

Walter Savitch

版 权 声 明

Authorized translation from the English language edition, entitled *Java™: An Introduction to Problem Solving and Programming, Fourth Edition*, 0131492020 by Walter Savitch, published by Pearson Education, Inc., publishing as Prentice Hall, Copyright © 1999, 2001, 2004, 2005 by Pearson Education, Inc.

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 Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD. and POSTS & TELECOM PRESS Copyright © 2007.

本书中文简体字版由Pearson Education Asia Ltd. 授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

目 录

第1章 异常处理	1	(选读)	82
1.1 基本异常处理	2	2.4.3 用ObjectInputStream从 二进制文件读出	82
1.1.1 Java中的异常	2	2.4.4 EOFException类	87
1.1.2 预定义的异常类	12	2.4.5 FileInputStream和 FileOutputStream类	89
1.1.3 ArrayIndexOutOfBoundsException- Exception (另一种阅读顺序)	12	2.5 用对象流进行对象I/O	93
1.2 定义异常类	13	2.5.1 类对象的二进制I/O	93
1.3 使用异常类	21	2.5.2 Serializable接口	96
1.3.1 声明异常(推脱责任)	21	2.5.3 二进制文件中的数组对象	97
1.3.2 不需要被捕获的异常	24	2.6 图形编程补充(选读)	99
1.3.3 AssertionError类(选读)	25	小结	104
1.3.4 多重抛出和捕获	25	自测题答案	104
1.3.5 finally块(选读)	31	编程项目	108
1.3.6 重新抛出异常(选读)	32	第3章 动态数据结构与泛型	110
1.4 图形编程补充(选读)	43	3.1 向量	111
小结	46	3.1.1 使用向量	111
自测题答案	47	3.1.2 参数化的类与泛型	120
编程项目	50	3.2 链式数据结构	121
第2章 流和文件I/O	53	3.2.1 链表	121
2.1 流和文件I/O概览	54	3.2.2 内部类	129
2.1.1 流的概念	54	3.2.3 节点内部类	130
2.1.2 为什么要用文件进行I/O	54	3.2.4 迭代器	131
2.1.3 文本文件和二进制文件的区别	54	3.2.5 对链表的异常处理	141
2.2 文本文件I/O	55	3.2.6 链表的变体	144
2.2.1 用PrintWriter进行的文本文件 输出	55	3.2.7 其他链式数据结构	145
2.2.2 带有BufferedReader的文本 文件输入	64	3.3 泛型	146
2.2.3 StringTokenizer类	70	小结	152
2.2.4 FileReader和FileOutputStream 类	73	自测题答案	152
2.3 File类	74	编程项目	159
2.4 基本的二进制文件I/O	77	第4章 递归	161
2.4.1 用ObjectOutputStream向 二进制文件输出	78	4.1 递归基础	161
2.4.2 一些有关writeUTF的细节		4.1.1 递归是如何工作的	165
		4.1.2 递归及迭代定义	170
		4.1.3 返回值的递归方法	171
		4.2 用递归编程	175

小结	185	6.3.3 applet与安全	271
自测题答案	185	小结	271
编程项目	187	自测题答案	271
第5章 使用Swing的视窗界面	189	编程项目	272
5.1 背景知识	190	第7章 更多有关Swing的内容	274
5.1.1 GUI——图形用户界面	190	7.1 菜单	274
5.1.2 事件驱动编程	190	7.1.1 菜单条、菜单和菜单条目	275
5.2 基本的Swing细节	192	7.1.2 嵌套式菜单	278
5.2.1 更多关于窗口侦听器的内容	197	7.2 使GUI更漂亮(功能也更强大)	280
5.2.2 屏幕对象的尺寸单位	198	7.2.1 添加图标	280
5.2.3 更多关于setVisible的内容	199	7.2.2 向菜单条目添加图标	285
5.2.4 JFrame类的一些方法	207	7.2.3 滚动条的JScrollPane类	286
5.2.5 布局管理器	208	7.2.4 添加边界	290
5.3 按钮和动作侦听器	214	7.3 更多布局管理器	294
5.3.1 按钮	216	7.3.1 BoxLayout管理器类	295
5.3.2 动作侦听器和动作事件	217	7.3.2 strut与glue	297
5.3.3 接口	221	7.3.3 设置组件间的间隔	299
5.3.4 模型-视图-控制器模式	223	7.3.4 Box容器类	299
5.4 容器类	225	7.3.5 CardLayout管理器	302
5.4.1 JPanel类	225	7.4 内部类	306
5.4.2 Container类	228	7.5 更多关于事件和侦听器的内容	308
5.5 GUI的文本I/O	231	7.5.1 WindowListener接口	308
5.5.1 文本区和文本域	231	7.5.2 更多与更新GUI有关的细节	318
5.5.2 输入和输出数字	238	7.6 Swing类的层次关系	318
5.5.3 捕获一个NumberFormatException- Exception	243	小结	320
小结	246	自测题答案	321
自测题答案	246	编程项目	323
编程项目	251	附录A 关键字(图灵网站下载)	
第6章 applet与HTML	253	附录B 优先规则(图灵网站下载)	
6.1 applet	254	附录C 统一代码(Unicode)字符集 (图灵网站下载)	
6.1.1 applet基础	254	附录D SavitchIn(图灵网站下载)	
6.1.2 运行一个applet	256	附录E 保护(protected)修饰符与包 (package)修饰符(图灵网站下载)	
6.1.3 向applet添加图标	259	附录F DecimalFormat类(图灵网站下载)	
6.2 HTML简介	261	附录G Iterator接口(图灵网站下载)	
6.2.1 HTML基础	261	附录H 克隆(图灵网站下载)	
6.2.2 插入超链接	263	附录I javadoc(图灵网站下载)	
6.2.3 显示一张图片	267	附录J JOptionPane类(图灵网站下载)	
6.3 HTML中的applet	267	附录K C++与Java的区别(图灵网站下载)	
6.3.1 将applet放入HTML文档中	268		
6.3.2 较老的applet类(选读)	270		

异常处理

异常能证明规则。

——常见格言（可能是类似“异常能够检验规则”的谬传）

写程序的一种方法是首先假设不会发生错误或者不正常。例如，如果程序访问列表的条目，就可能假设该列表不为空。当程序能够在所有事情都按计划发展的核心状况下工作后，再考虑增加代码来处理例外的情况。在Java中，有一种方式反映了这种编程方法。起初，写代码时假设没有什么不寻常的事情会发生。随后，使用Java的异常处理设施为不寻常的情况添加处理代码。异常处理允许将程序或者方法的定义分成一些独立的部分，分别表示正常情况和异常情况，这样就把一个大的编程任务分解为两个更小更容易实现的编程任务。本章将向读者介绍Java中的异常处理。

异常处理最主要的用处可能是处理具有特殊情况的方法，这种特殊情况在方法的不同应用场合有不同的处理方式。方法可能会被很多程序使用，其中有些程序会对这种特殊情况采取一种处理办法，而其他程序会对这种特殊情况采取别的办法。例如，如果方法中有除以0的情况，那么对于某些调用该方法的情形，程序应当结束；而对于其他情形，会有别的处理方式。读者会看到，在这种特殊情况发生时，定义这样的方法来抛出一个异常。这种处置使得这种特殊情况在方法外部才被处理。这样一来，这种特殊情况在不同的调用中就可以做不同的处理。

目标

- 熟悉异常处理的各种记号。
- 学习Java的异常处理语法。
- 培养在自己的程序和类中有效地使用异常处理的技能。

预备知识

除了1.1.3节之外，本章需要读者先读《Java程序设计与问题解决：基础篇》的第1章~第5章以及第7章。学习1.1.3节则需要读者掌握该书第6章有关数组的内容。这一小节可以选读。

（除了1.1.3节之外）1.1节可以在读过该书的第1章~第5章之后阅读，它不需要其中第6章和第7章中讲到的知识。

1.1 基本异常处理

杰克，机灵些，
杰克，快一些，
杰克，跳过那个蜡烛。

——佚名

在Java中，异常处理（exception handling）如下进行：Java语言本身或者读者的代码提供一种机制，当有不正常的情况发生时，可以发出信号。这种发出信号的过程被称为抛出异常（throwing an exception）。在程序的另一部分——位于一个单独的类或方法中，甚至就在该程序的另外一部分代码中——可以放置代码来处理这种异常情况。这部分代码完成的任务就称为处理异常（handling the exception）。当然，我们还是需要详细解释Java中异常处理的各种细节。

1.1.1 Java中的异常

绝大多数小程序基本上用不到异常处理，因此异常处理不容易在这样的代码中出现。为了给出一个简单的例子，我们将用一个没什么价值的程序作为第一个示例。首先不使用Java的异常处理设施来写这个程序，然后再用异常处理重写它。

在本例中，假设牛奶是社会中的一种重要物资，人们几乎不能面对它断货的情况，不过我们还是要让程序适应极少见的牛奶短缺的情况。下面假设我们不会用完牛奶，基本代码如下所示：

```
System.out.println("Enter number of donuts:");
donutCount = keyboard.nextInt();

System.out.println("Enter number of glasses of milk:");
milkCount = keyboard.nextInt();

donutsPerGlass = donutCount/(double)milkCount;
System.out.println(donutCount + " donuts.");
System.out.println(milkCount + " glasses of milk.");
System.out.println("You have " + donutsPerGlass
    + " donuts for each glass of milk.");
```

如果没有牛奶了（也就是说输入的牛奶的瓶数是0），这段代码会导致除以0，这是一个错误。为了妥善处理牛奶短缺的罕见情况，可以为这种异常的情形添加一个测试。图1-1中展示了增加测试此种特殊情况后的代码的完整程序。现在我们来查看如何用Java的异常处理设施来重写这个程序。

```
import java.util.*;

public class GotMilk
{
    public static void main(String[] args)
    {
```

图1-1 处理问题情形的一种方法

```
int donutCount, milkCount;
double donutsPerGlass;
Scanner keyboard = new Scanner(System.in);

System.out.println("Enter number of donuts:");
donutCount = keyboard.nextInt();

System.out.println("Enter number of glasses of milk:");
milkCount = keyboard.nextInt();

if (milkCount < 1)
{
    System.out.println("No Milk!");
    System.out.println("Go buy some milk.");
}
else
{
    donutsPerGlass = donutCount / (double)milkCount;
    System.out.println(donutCount + " donuts.");
    System.out.println(milkCount + " glasses of milk.");
    System.out.println("You have " + donutsPerGlass
        + " donuts for each glass of milk.");
}
System.out.println("End of program.");
}
```

屏幕对话示例

```
Enter number of donuts:
2
Enter number of glasses of milk:
0
No Milk!
Go buy some milk.
End of program.
```

图1-1 (续)

在图1-2中，我们使用异常重写了图1-1中的程序。因为这个例子太简单，在此情况下，读者可能不愿意使用异常。但通过这个例子可以学习Java的异常处理语法。尽管整个程序没有因为使用异常而更加简单，但是在关键字try和catch之间的部分比原先的程序更简洁，这暗示了使用异常的好处。

让我们看一看try和catch之间的代码。这段代码基本上和图1-1中的代码相同，只是原先很庞杂的if-else语句（在图1-1中被高亮显示）被下面的短小的if语句所替代：

```
if (milkCount < 1)
    throw new Exception("Exception: No Milk!");
```

```

import java.util.*;

public class ExceptionDemo
{
    public static void main(String[] args)
    {
        int donutCount, milkCount;
        double donutsPerGlass;
        Scanner keyboard = new Scanner(System.in);

        try
        {
            System.out.println("Enter number of donuts:");
            donutCount = keyboard.nextInt();

            System.out.println("Enter number of glasses of milk:");
            milkCount = keyboard.nextInt();

            if (milkCount < 1)
                throw new Exception("Exception: No Milk!");

            donutsPerGlass = donutCount/(double)milkCount;
            System.out.println(donutCount + " donuts.");
            System.out.println(milkCount + " glasses of milk.");
            System.out.println("You have " + donutsPerGlass
                               + " donuts for each glass of milk.");
        }
        catch(Exception e)
        {
            System.out.println(e.getMessage());
            System.out.println("Go buy some milk.");
        }
        System.out.println("End of program.");
    }
}

```

这个程序只是用于学习异常处理基本语法的一个简单例子。

try块

throw语句

catch块

屏幕对话示例 1

```

Enter number of donuts:
3
Enter number of glasses of milk:
2
3 donuts.
2 glasses of milk.
You have 1.5 donuts for each glass of milk.
End of program.

```

屏幕对话示例 2

```

Enter number of donuts:
2
Enter number of glasses of milk:
0
Exception: No Milk!
Go buy some milk.
End of program.

```

图1-2 异常处理示例