

TURING

图灵原版计算机科学系列

PEARSON
Prentice
Hall

Java An Introduction to Problem Solving & Programming
Fourth Edition

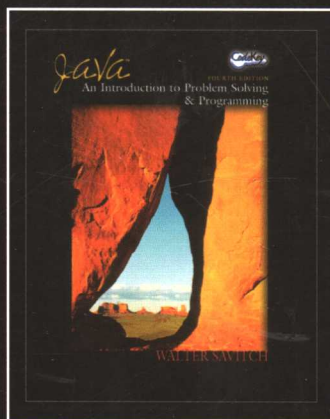
Java

程序设计与问题解决

基础篇

(英文版 · 第4版)

[美] Walter Savitch 著



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原版计算机科学系列

Java An Introduction to Problem Solving & Programming
Fourth Edition

Java

程序设计与问题解决

基础篇 苏工业学院图书馆

(英文版 · 第4版) 藏书章

[美] Walter Savitch 著



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

Java 程序设计与问题解决. 基础篇: 第 4 版 / (美) 萨维奇著. —北京: 人民邮电出版社, 2006.12
(图灵原版计算机科学系列)

ISBN 7-115-15288-8

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

中国版本图书馆 CIP 数据核字 (2006) 第 109963 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。书中以Java基础知识为主, 包括基本类型、字符串、控制流、类和方法的定义、数组、继承等, 此外, 还介绍了一些编程技巧。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。本书内容高度协调, 包含自测题、编程提示和编程项目以及关于编程缺陷和容易出错的程序语言特征的章节。

本书适合作为高等院校计算机专业本科生或研究生Java语言程序设计课程的教材, 也适合相关技术人员参考。

图灵原版计算机科学系列

Java 程序设计与问题解决: 基础篇 (英文版·第 4 版)

- ◆ 著 [美] Walter Savitch
责任编辑 杨海玲
- ◆ 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>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 800 × 1000 1/16
印张: 36.75
字数: 750 千字 2006 年 12 月第 1 版
印数: 1—3 000 册 2006 年 12 月北京第 1 次印刷

著作权合同登记号 图字: 01-2006-5776 号

ISBN 7-115-15288-8/TP · 5701

定价: 59.00 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，也适用于长到一个学年的课程。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，在这种情况下，可以将前面几章指定为学生的课外阅读材料。

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，你使用的Java必须是5.0版或更高的版本¹。

本版中的修改

如果你没有用过本书的第3版，就可以跳过本节。如果你用过第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。你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。这个版本要进行的最大的修改就是用Java版本5.0中提供的新的Scanner类取代SavitchIn类²。如果你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，你可以选择在早期就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章至第9章³，每章都以一个图形编程补充小节作为结束，这个小节涵盖了使用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9章之后（实际上是在第7章之后的任意时刻），你希望获得更多有关图形和GUI的知识，你可以直接跳到第12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与升级正文以便与Java 5.0版本相匹配有关的：为键盘输入使用了新的Scanner类。我们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

-
1. Sun公司修改了它的版本编号方式。版本5.0原来被称为版本1.5，在某些地方你可能会发现它还被称为版本1.5。
 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1章至第7章的内容，文中提到的第8章至第14章的内容包含在本书的姊妹篇《Java程序设计与问题解决：高级篇（英文版·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

型类型参数。第10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版本的最新特性。特别是，我们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版检查了所有代码。

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，你可以很容易地改变所要讲授的章节间的次序。前言后面的依赖关系图对重新组织材料的详情进行了解释，每章开始的“预备知识”(Prerequisites)小节中对其进行了更详细的解释。

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

从第1章开始，各章就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。在第7章之后，就可以跳转到讲述GUI编程的主要章节（第12章～第14章）中去了。或者继续讲到第9章的图形编程补充小节，这样你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

本书的设计目标是教学生基本的问题求解及编程技术，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。在第1章和

第2章接触了一些类的使用，第4章介绍了如何定义类。所有关于类的基本信息，包括继承在内，都在第7章结束之前进行了交待（即使略过第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10章学到向量时介绍的。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 (Self-Test Questions)

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，但更重要的是，讲授基本的编程技术。本书不要事先具备编程经验，除了一些简单的代数知识之外，也不需要更多的数学知识。但是，为了最大程度地从本书获益，在你的机器上应该装有一个可用的Java版本，这样你就可以练习书中给出的示例和技巧了。你应该安装Java 5.0版本（或更高的版本）。（Sun公司修改了它的版本编号方式。版本5.0原来被称为版本1.5，在某些地方你可能会发现它仍然被称为版本1.5。如果你有一个名为“版本1.5”的Java的副本，也应该可以。）

如果以前编写过程序

使用本书不要求你以前有编程经验。本书是为初学者设计的。如果你碰巧有过使用其他编程语言的经验，不要假设Java和你习惯使用的编程语言是一样的。所有的语言都是不同的，它们之前的区别即使很小，也足以给你带来问题。你至少应该读一下1.3节中被框起来的内容和第2章中所有被框起来的内容。当阅读到第4章的时候，阅读整章是比较明智的。

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11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，你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。对初学者来说，我们推荐使用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，一定要下载版本号为5.0的版本或更新的版本。

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 (Self-Test Questions)

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，你还可以并且应该将它当作一本参考书。当需要查找一个你已经忘记了或者你听别人提到但你还没学过的问题时，就来看看索引¹吧。很多索引条目中都给出了一个“快速参考”的页码。翻到这个快速参考页。这一页中会有一个很短的条目，这个条目通常包含在一个方框内，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。

每章中用方框框起来的部分都对那一章的主要内容进行了快速的总结。你可以根据这些方框中的内容对那一章进行复习，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和我联系：

`wsavitch@ucsd.edu`

遗憾的是，我不能为你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，你只能与你的老师联系。但是，请记住，每章的结尾都有所有自测题的答案。

Walter Savitch

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

1. 本书索引请读者到www.turingbook.com下载。——编者注

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，本书很多素材在这里得到了检验。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。这些学生的建议和那些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，并对其进行了课堂试用。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——北卡罗来纳州大学 (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——加州大学北岭分校 (California State University, Northridge)

Michael Olan——斯托克顿州 (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——加州大学 (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准备了教师指南的编程答案, Robert Burton准备了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

TABLE OF CONTENTS



CHAPTER 1 Introduction to Computers and Java 3

Objectives 4

Prerequisites 4

1.1 Computer Basics 5

Hardware and Memory 5

Programs 7

Programming Languages and Compilers 9

Java Byte-Code 10

Class Loader 12

1.2 Designing Programs 13

Object-Oriented Programming 13

Encapsulation 14

Polymorphism 15

Inheritance 16

If You Know Some Other Programming Language 17

Algorithms 17

Reusable Components 18

Testing and Debugging 19

1.3 A Sip of Java 21

History of the Java Language 21

Applets 22

A First Java Application Program 22

Compiling a Java Program or Class 27

Running a Java Program 29

1.4 Graphics Supplement (*Optional*) 29

Objects and Methods 30

A Sample Graphics Applet 30

Size and Positions of Figures 33

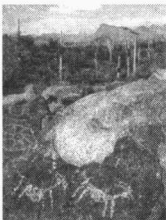
Drawing Arcs 36

Running an Applet 36

Chapter Summary 39

Answers to Self-Test Questions 40

Programming Projects 42



CHAPTER 2 Primitive Types, Strings, and Console I/O 45

Objectives 46

Prerequisites 46

2.1 Primitive Types and Expressions 47

Variables 47

Java Identifiers 51

Primitive Types 52

Assignment Statements 53

Specialized Assignment Operators 55

Simple Screen Output 56

Simple Input 56

Number Constants	58
Assignment Compatibilities	59
Type Casting	61
Programming Tip: Initialize Variables	63
Arithmetic Operators	66
Parentheses and Precedence Rules	67
Case Study: Vending Machine Change	69
Increment and Decrement Operators	74
More about the Increment and Decrement Operators	75
2.2 The Class <code>String</code>	76
String Constants and Variables	76
Concatenation of Strings	77
Classes	78
String Methods	79
String Processing	83
Escape Characters	84
The Unicode Character Set	85
2.3 Keyboard and Screen I/O	87
Screen Output	87
Keyboard Input	90
2.4 Documentation and Style	98
Programming Tip: Use Meaningful Names for Variables	99
Documentation and Comments	99
Indenting	102
Named Constants	102
2.5 Graphics Supplement (Optional)	106
Programming Example: Style Rules Applied to a Graphics Applet	106
<code>JOptionPane</code>	106
Inputting Other Numeric Types	115
Programming Example: Change-Making Program with	
Windowing I/O	117
Chapter Summary	120
Answers to Self-Test Questions	120
Programming Projects	125

CHAPTER 3 Flow of Control 129

Objectives	130
Prerequisites	130
3.1 Branching Statements	130
The <code>if-else</code> Statement	130
Introduction to Boolean Expressions	134
Programming Tip: Alphabetical Order	139
Nested Statements and Compound Statements	142
Multibranch <code>if-else</code> Statements	145



4 Table of Contents

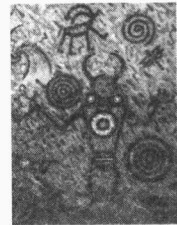
Programming Example: Assigning Letter Grades	146
The <code>switch</code> Statement	149
The Conditional Operator (<i>Optional</i>)	153
3.2 Java Loop Statements	155
<code>while</code> Statements	156
The <code>do-while</code> Statement	160
Programming Example: Bug Infestation	164
The <code>for</code> Statement	168
The Comma in <code>for</code> Statements (<i>Optional</i>)	172
The <code>break</code> Statement in Loops	175
The <code>exit</code> Method	177
3.3 Programming with Loops	179
The Loop Body	179
Initializing Statements	181
Ending a Loop	182
Programming Example: Nested Loops	183
Programming Tip: Avoid Declaring Variables in a Loop Body	184
Loop Bugs	184
Tracing Variables	188
3.4 The Type <code>boolean</code>	189
Boolean Expressions and Boolean Variables	190
Programming Tip: Naming Boolean Variables	191
Precedence Rules	192
Input and Output of Boolean Values	195
Case Study: Using a Boolean Variable to End a Loop	195
3.5 Graphics Supplement (<i>Optional</i>)	199
Specifying a Drawing Color	199
Programming Example: A Multiface Applet	203
The <code>drawString</code> Method	208
A <code>JOptionPane</code> Yes/No Window	209
Chapter Summary	211
Answers to Self-Test Questions	211
Programming Projects	217



CHAPTER 4 Defining Classes and Methods 221

Objectives	222
Prerequisites	222
4.1 Class and Method Definitions	222
Class Files and Separate Compilation	224
Instance Variables	225
Using Methods	228
<code>void</code> Method Definitions	231
Methods That Return a Value	233
The <code>this</code> Parameter	238
Local Variables	240

Blocks	242
Parameters of a Primitive Type	243
Summary of Class and Method Definition Syntax	249
4.2 Information Hiding and Encapsulation	250
Information Hiding	250
Programming Tip: Parameter Names Are Local to the Method	251
Precondition and Postcondition Comments	251
The <code>public</code> and <code>private</code> Modifiers	254
Programming Tip: Instance Variables Should Be <code>private</code>	257
Programming Example: A Purchase Class	261
Encapsulation	265
Automatic Documentation with <code>javadoc</code>	268
UML Class Diagrams	268
4.3 Objects and Reference	270
Variables of a Class Type and Objects	270
Programming Example: A Species Class	279
Boolean-Valued Methods	282
Class Parameters	284
Comparing Class Parameters and Primitive-Type Parameters	287
4.4 Graphics Supplement (Optional)	291
The <code>Graphics</code> Class	291
Programming Example: Multiple Faces Redone with a Helping Method	294
The <code>init</code> Method	298
Adding Labels to an Applet	298
Chapter Summary	302
Answers to Self-Test Questions	303
Programming Projects	310
 CHAPTER 5 More About Objects and Methods	 315
Objectives	316
Prerequisites	316
5.1 Programming with Methods	317
Methods Calling Methods	317
Programming Tip: Make Helping Methods <code>Private</code>	323
5.2 Static Methods and Static Variables	326
Static Methods	326
Static Variables (Optional)	332
The <code>Math</code> Class	334
<code>Integer</code> , <code>Double</code> , and Other Wrapper Classes	337
5.3 Designing Methods	343
Case Study: Formatting Output	343
Top-Down Design	348
Testing Methods	350



5.4 Overloading 352

Overloading Basics 352

Programming Example: A Pet Class 355

Programming Example: A Class for Money 363

5.5 Constructors 369

Defining Constructors 370

Programming Tip: You Can Use Other Methods in a Constructor 378

5.6 Information Hiding Revisited 380

5.7 Packages 384

Packages and Importing 384

Package Names and Directories 385

Name Clashes 388

5.8 Graphics Supplement (*Optional*) 389

Programming Tip: Code Appearance and Actions Separately 389

Adding Buttons 390

Event-Driven Programming 391

Programming Example: A Complete Button Applet 391

Programming Buttons 392

Icons 399

Changing Visibility 401

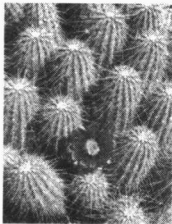
Programming Example: An Example of Changing
Visibility 401

What's Next 405

Chapter Summary 405

Answers to Self-Test Questions 406

Programming Projects 412



CHAPTER 6 Arrays 417

Objectives 418

Prerequisites 418

6.1 Array Basics 419

Creating and Accessing Arrays 419

Array Details 420

Programming Tip: Use Singular Array Names 424

The Length Instance Variable 425

Programming Tip: Use a for Loop to Step Through an Array 427

Initializing Arrays 429

6.2 Arrays in Classes and Methods 430

Case Study: Sales Report 430

Indexed Variables as Method Arguments 438

Entire Arrays as Method Arguments 440

Arguments for the Method main 441

Methods That Return Arrays 446

6.3 Programming with Arrays and Classes 449

Programming Example: A Specialized List Class 450

Partially Filled Arrays 457

Searching an Array 457

6.4 Sorting Arrays 460

Selection Sort 460

Other Sorting Algorithms 465

6.5 Multidimensional Arrays 465

Multidimensional-Array Basics 467

Multidimensional-Array Parameters and Returned Values 470

Implementation of Multidimensional Arrays 471

Ragged Arrays (*Optional*) 475

Programming Example: Employee Time Records 476

6.6 Graphics Supplement (*Optional*) 482

Graphics Supplement Part 1: Text Areas and Text Fields 483

Programming Example: A Question-and-Answer Applet 483

JTextArea and JTextField Objects 483

Graphics Supplement Part 2: Drawing Polygons 487

Drawing Polygons 487

Chapter Summary 491

Answers to Self-Test Questions 492

Programming Projects 497

CHAPTER 7 Inheritance 501

Objectives 502

Prerequisites 502

7.1 Inheritance Basics 502

Programming Example: A Person Class 503

Derived Classes 504

Overriding Method Definitions 507

Overriding Versus Overloading 508

The `final` Modifier 508

Programming Tip: Assume That Your Coworkers Are Malicious 510

UML Inheritance Diagrams 510

7.2 Programming with Inheritance 513

Constructors in Derived Classes 513

The `this` Method (*Optional*) 514

Call to an Overridden Method 515

Programming Example: Multilevel Derived Classes 516

A Subtle Point About Overloading and Overriding (*Optional*) 521

Programming Tip: An Object Can Have More than One Type 522

Programming Tip: “Is a” and “Has a” Relations 525

The Class Object 526

Case Study: Character Graphics 528

