

国外信息科学与技术经典图书系列

Fundamentals of JAVA Programming

(Fifth Edition)

JAVA编程

(第五版)

[美] Joyce Farrell 著

(英文影印版)



科学出版社



国外信息科学与技术经典图书系列

JAVA 编程 (英文影印版)

(第五版)

Fundamentals of JAVA Programming
(Fifth Edition)

〔美〕 Joyce Farrell 著

科学出版社

北 京

图字: 01-2011-6335

内 容 简 介

本书为入门级程序员提供了用 JAVA 编程语言开发应用程序的方法。JAVA 语言深受专业程序员青睐,因为它可以用来制造在视觉上有趣的图形用户界面(GUI)和互联网应用程序。本书也为学生在学习基本的结构化和面向对象程序设计技术的前提下,尽快开始程序编制提供了良好的指导。

本书可作为计算机专业的双语教材或教学参考书,也可供工程技术人员参考。

Joyce Farrell

Fundamentals of JAVA Programming, 5e

ISBN: 978-1-133-69109-9

Copyright © 2012 Cengage Learning Asia Pte Ltd.

Science Press is authorized by Cengage Learning to publish and distribute exclusively this custom reprint edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

此客户订制影印版由圣智学习出版公司授权科学出版社独家出版发行。此版本仅限在中华人民共和国境内(不包括中国香港、澳门特别行政区及中国台湾)销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可,不得以任何方式复制或发行本书的任何部分。

Cengage Learning Asia Pte. Ltd.

5 Shenton Way, # 01-01 UIC Building, Singapore 068808

北京市版权局著作权合同登记号 图字: 01-2011-6335

本书封面贴有 **Cengage Learning** 防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

JAVA 编程 = Fundamentals of JAVA Programming: 5 版: 英文/(美)法瑞尔(Farrell, J.)等著. —影印本.

—北京: 科学出版社, 2012

(国外信息科学与技术经典图书系列)

ISBN 978-7-03-032910-3

I. ①J… II. ①法… III. ①JAVA 语言-编程设计-英文 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 246556 号

责任编辑: 王鑫光 匡 敏 / 责任印制: 张克忠 / 封面设计: 迷底书装

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

保定市中西美凯印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2012 年 1 月第 一 版 开本: 787×960 1/16

2012 年 1 月第一次印刷 印张: 45 1/4

字数: 900 000

定价: 99.00 元

(如有印装质量问题, 我社负责调换)

PREFACE

Java Programming, Fifth Edition provides the beginning programmer with a guide to developing applications using the Java programming language. Java is popular among professional programmers because it can be used to build visually interesting graphical user interface (GUI) and Web-based applications. Java also provides an excellent environment for the beginning programmer—a student quickly can build useful programs while learning the basics of structured and object-oriented programming techniques.

This textbook assumes that you have little or no programming experience. This book provides a solid background in good object-oriented programming techniques and introduces object-oriented terminology using clear, familiar language. The writing is nontechnical and emphasizes good programming practices. The examples are business examples; they do not assume a mathematical background beyond high-school business math. In addition, the examples illustrate only one or two major points; they do not contain so many features that you become lost following irrelevant and extraneous details. The explanations in this textbook are written clearly in straightforward sentences so that native and non-native English speakers alike can master the programming concepts. Complete, working code examples appear frequently in each chapter; these examples help the student make the transition from the theoretical to the practical. The code presented in each chapter is also provided on disk, so that students can easily run the programs and experiment with changes to them.

ORGANIZATION AND COVERAGE

Java Programming, Fifth Edition presents Java programming concepts, enforcing good style, logical thinking, and the object-oriented paradigm. Objects are covered right from the beginning, earlier than in many other textbooks. You create your first Java program in Chapter 1. Chapters 2, 3, and 4 increase your understanding of how data, classes, objects, and methods interact in an object-oriented environment.

Chapters 5 and 6 explore input and repetition structures, which are the backbone of programming logic and essential to creating useful programs in any language. You learn the special considerations of string and array manipulation in Chapters 7 and 8.

Chapters 9, 10, and 11 thoroughly cover inheritance (the object-oriented concept that allows you to develop new objects quickly by adapting the features of existing ones) and exception handling (the object-oriented approach to handling errors). Both are important concepts in object-oriented design. Chapter 12 provides information on handling files so you can permanently store and retrieve program output. Chapter 13 introduces Swing components.

In every chapter, *Java Programming, Fifth Edition* follows the text explanation with a “You Do It” section that contains step-by-step exercises to illustrate the concepts just learned, reinforcing the student’s understanding and allowing concepts to be better retained. Creating the programs in the step-by-step examples also provides students with a successful experience in the language; finishing the examples provides them with models for their own creations.

The student using *Java Programming, Fifth Edition* builds applications from the bottom up, rather than starting with existing objects. This facilitates a deeper understanding of the concepts used in object-oriented programming, and engenders appreciation for the existing objects students use as their knowledge of the language advances. When students complete this book, they will know how to modify and create simple Java programs and will have the tools to create more complex examples. They also will have a fundamental knowledge of object-oriented programming, which will serve them well in advanced Java courses or in studying other object-oriented languages such as C++, C#, and Visual Basic.

FEATURES

Java Programming, Fifth Edition is a superior textbook because it also includes the following features:

- » **Objectives:** Each chapter begins with a list of objectives so you know the topics that will be presented in the chapter. In addition to providing a quick reference to topics covered, this feature provides a useful study aid.
- » **Notes:** These highlighted tips provide additional information—for example, an alternative method of performing a procedure, another term for a concept, background information on a technique, or a common error to avoid.
- » **Figures:** Each chapter contains many figures. Code figures are most frequently 25 lines or less, illustrating one concept at a time. Frequently placed screen shots show exactly how program output appears.

- NEW!** » **Callouts in more figures:** Callouts have been added to many figures to help students focus on the points emphasized in the text. Some icons contain the words “Don’t Do It” to emphasize when an example illustrates a practice not to emulate.
- » **Color:** The code figures in each chapter contain all Java keywords in brown. This helps students identify keywords more easily, distinguishing them from programmer-selected names.
- » **Files:** The Student Disk holds more than 180 files that contain the code presented in the figures in each chapter. Students can run the code for themselves, view the output, and make changes to the code to observe the effects.
- NEW!** » **Two Truths and a Lie:** A new quiz reviews each chapter section, with answers provided. This quiz contains three statements from the preceding section of text—two statements are true and one is false. Over the years, students have requested answers to problems, but we have hesitated to distribute them in case instructors want to use problems as assignments or test questions. These true-false mini-quizzes provide students with immediate feedback as they read, without “giving away” answers to the existing multiple-choice and programming problem questions.
- » **You Do It:** In each chapter, step-by-step exercises help the student create multiple working programs that emphasize the logic a programmer uses in choosing statements to include. This section provides a means for students to achieve success on their own—even those in online or distance learning classes.
- NEW!** » **Don’t Do It:** This section at the end of each chapter summarizes common mistakes and pitfalls that plague new programmers while learning the current topic.

P R E F A C E

- » **Key Terms:** Each chapter includes a list of newly introduced vocabulary, shown in the order of appearance in the text. The list of key terms provides a mini-review of the major concepts in the chapter.
- » **Summaries:** Following each chapter is a summary that recaps the programming concepts and techniques covered in the chapter. This feature helps students check their understanding of the main points in each chapter.
- » **Review Questions:** Each chapter includes 20 multiple-choice questions that serve as a review of chapter topics.
- » **Exercises:** Each chapter concludes with meaningful programming exercises that provide additional practice of the skills and concepts learned in the chapter. These exercises vary in difficulty and are designed to allow exploration of logical programming concepts.
- » **Game Zone:** Each chapter provides one or more exercises in which the student creates interactive games using the programming techniques learned up to that point; 70 game programs are suggested in the book. The games are fun to create and play; writing them motivates students to master the necessary programming techniques. Students might exchange completed game programs with each other, suggesting improvements and discovering alternate ways to accomplish tasks.
- » **Tough Questions:** Each chapter includes two or more fairly difficult, and often open-ended, questions that are typical of what an applicant might encounter in a technical job interview. Some questions involve coding; others might involve research. **NEW!**
- » **Up for Discussion:** Each chapter concludes with a few thought-provoking questions concerning programming in general or Java in particular. The questions can be used to start classroom or online discussions, or to develop and encourage research, writing, and language skills.
- » **Glossary:** This edition includes a glossary that contains definitions for all key terms in the book, presented in alphabetical order. **NEW!**
- » **Appendix on javadoc:** This edition includes a new appendix on creating javadoc comments. **NEW!**
- » **Other pedagogical improvements:** This edition introduces the following pedagogical improvements: **NEW!**
 - » The `Scanner` class is introduced in Chapter 2 to facilitate user keyboard entry in programs.
 - » Programming examples provide earlier and more consistent use of named constants.
 - » Clearer distinction between troublesome concepts is provided—for example, argument vs. parameter and static vs. nonstatic.
 - » The `String` chapter focuses on `StringBuilder` instead of `StringBuffer` because `StringBuilder` is more efficient. However, it is emphasized that the two classes are used in exactly the same way.
 - » The GUI chapters have been completely rewritten and moved later in the book, which makes it easier for instructors who want to cover the concepts of inheritance and polymorphism first. Similarly, applet coverage has been removed from the GUI chapters, which makes it easier for instructors who want to cover GUI topics first.
 - » Applets have been moved to the last chapter in the book, reflecting their diminished popularity as a business tool.
- » **Quality:** Every program example in the book, as well as every exercise and game solution, was tested by the author and then tested again by a Quality Assurance team using Java Standard Edition (SE) 6, the most recent version available. (The external version number used by Sun Microsystems is 6.0; the internal version number is 1.6.0. For more information on the features of the JDK, visit <http://java.sun.com>.)

» **CD-ROM included with book:** The CD that comes with this book includes the following items:

- » Sun Microsystems Java SE 6, the Java language, compiler, and runtime environment
- » The jGRASP integrated development environment for Java
- » Code files for all Java program examples contained in the text

TEACHING TOOLS

The following supplemental materials are available when this book is used in a classroom setting. All of the teaching tools available with this book are provided to the instructor on a single CD.

- » **Electronic Instructor's Manual:** The Instructor's Manual that accompanies this textbook includes additional instructional material to assist in class preparation, including items such as Sample Syllabi, Chapter Outlines, Technical Notes, Lecture Notes, Quick Quizzes, Teaching Tips, Discussion Topics, and Key Terms.
- » **ExamView®:** This textbook is accompanied by ExamView, a powerful testing software package that allows instructors to create and administer printed, computer (LAN-based), and Internet-based exams. ExamView includes hundreds of questions that correspond to the topics covered in this text, enabling students to generate detailed study guides that include page references for further review. The computer-based and Internet testing components allow students to take exams at their computers, and they save the instructor time by grading each exam automatically.
- » **PowerPoint Presentations:** This book comes with Microsoft PowerPoint slides for each chapter. These are included as a teaching aid for classroom presentation, to make available to students on the network for chapter review, or to be printed for classroom distribution. Instructors can add their own slides for additional topics they introduce to the class.
- » **Solution Files:** Solutions to "You Do It" exercises and all end-of-chapter exercises are provided on the Instructor Resources CD and on the Course Technology Web site at www.course.com. The solutions are password protected.
Annotated solutions are provided for the multiple-choice Review Questions. For example, if students are likely to debate answer choices, or not understand the choice deemed to be the correct one, a rationale is provided.
- » **Distance Learning:** Course Technology is proud to present online test banks in WebCT and Blackboard to provide the most complete and dynamic learning experience possible. Instructors are encouraged to make the most of the course, both online and offline. For more information on how to access the online test bank, contact your local Course Technology sales representative.

ACKNOWLEDGEMENTS

I would like to thank all of the people who helped to make this book a reality, especially Dan Seiter, Development Editor. Dan's suggestions and attention to detail made this a superior book, and his sense of humor made writing it practically painless.

Thanks also to Tricia Coia, Managing Editor; and Heather Furrow, Content Project Manager. I am lucky to work with Tricia and Heather; they are dedicated to producing quality instructional materials.

Thanks to Serge Palladino, John Freitas, and Chris Scriver of the Quality Assurance Department.

Thank you to Dick Grant of Seminole Community College, Sanford, Florida. He provided important technical and pedagogical suggestions based on his classroom use of this book. He possesses the rare combination of excellent teacher and programmer, and he made this book more accurate and more useful to students.

I am also grateful to the many other reviewers who provided comments and encouragement during this book's development, including Karlyn Barilovits, Kaplan University; Kay Chen, Bucks County Community College; Roman Erenshteyn, Goldey-Beacom College; Jeff Hedrington, University of Phoenix-Online; and Aaron Jagers, Louisiana Technical College.

Thanks, too, to my husband, Geoff, who supports me every step of the way. Finally, this book is dedicated to our lifelong friends, George and Mary Profeta.

Joyce Farrell

READ THIS BEFORE YOU BEGIN

The following information will help you as you prepare to use this textbook.

TO THE USER OF THE DATA FILES

To complete the steps and projects in this book, you need data files that have been created specifically for this book. Your instructor will provide the data files to you. You also can obtain the files electronically from the Course Technology Web site by connecting to www.course.com and then searching for this book title. Note that you can use a computer in your school lab or your own computer to complete the exercises in this book.

USING YOUR OWN COMPUTER

To use your own computer to complete the steps and exercises, you need the following:

- » **Software:** Java SE 6, available from <http://java.sun.com>. (Although almost all of the examples in this book will work with earlier versions of Java, this book was created using Java 6.) The book clearly points out the few cases when an example does not work with earlier versions of Java. You also need a text editor, such as Notepad. A few exercises ask you to use a browser, such as Internet Explorer.
- » **Hardware:** To install Java on your computer, the Java Web site suggests at least a Pentium III 500-MHz system with 512 MB of memory and at least 850 MB of disk space. A Pentium IV 1.4-GHz system with 1 GB of memory and 1 GB of disk space is recommended.
- » **Data Files:** You cannot complete all the chapters and projects in this book using your own computer unless you have the data files. You can get the data files from your instructor, or you can obtain the data files electronically from the Course Technology Web site by connecting to www.course.com and then searching for this book title.

The following material is provided on the CD that comes with this book:

- » Sun Microsystems Java SE 6, the Java language, compiler, and runtime environment
- » Sun Microsystems Java Application Programming Interface (API) Specification, official documentation for the Java programming language
- » The jGRASP integrated development environment for Java
- » Code files for all Java program examples contained in the text

VISIT OUR WORLD WIDE WEB SITE

Additional materials designed especially for this book might be available for your course. Periodically search www.course.com for more details and updates.

CONTENTS

PREFACE	xv
READ THIS BEFORE YOU BEGIN	xx
CHAPTER 1 CREATING YOUR FIRST JAVA CLASSES	1
LEARNING ABOUT PROGRAMMING	2
INTRODUCING OBJECT-ORIENTED PROGRAMMING CONCEPTS	4
Procedural Programming	4
Object-Oriented Programming	4
Understanding Objects, Classes, and Encapsulation	5
Understanding Inheritance and Polymorphism	7
LEARNING ABOUT JAVA	8
Java Program Types	9
ANALYZING A JAVA APPLICATION THAT USES CONSOLE OUTPUT	10
Understanding the Statement That Prints the Output	10
Understanding the <code>First</code> Class	11
Understanding the <code>main()</code> Method	14
ADDING COMMENTS TO A JAVA CLASS	16
SAVING, COMPILING, RUNNING, AND MODIFYING A JAVA APPLICATION	18
Saving a Java Class	18
Compiling a Java Class	18
Running a Java Application	19
Modifying a Java Class	19
CREATING A JAVA APPLICATION USING GUI OUTPUT	21
CORRECTING ERRORS AND FINDING HELP	23
YOU DO IT	26
Your First Application	26
Adding Comments to a Class	27
Modifying a Class	28
Creating a Dialog Box	29
DON'T DO IT	30
KEY TERMS	31
CHAPTER SUMMARY	34
REVIEW QUESTIONS	35
EXERCISES	37

CONTENTS

DEBUGGING EXERCISES	39
GAME ZONE	39
TOUGH QUESTIONS	40
UP FOR DISCUSSION	41
CHAPTER 2 USING DATA WITHIN A PROGRAM	43
USING CONSTANTS AND VARIABLES	44
Declaring Variables	45
Declaring Named Constants	46
Pitfall: Forgetting That a Variable Holds One Value at a Time	48
LEARNING ABOUT THE <code>int</code> DATA TYPE	48
DISPLAYING DATA	50
WRITING ARITHMETIC STATEMENTS	51
Writing Arithmetic Statements Efficiently	53
USING THE <code>BOOLEAN</code> DATA TYPE	54
LEARNING ABOUT FLOATING-POINT DATA TYPES	55
UNDERSTANDING NUMERIC-TYPE CONVERSION	56
WORKING WITH THE <code>char</code> DATA TYPE	58
USING THE <code>Scanner</code> CLASS FOR KEYBOARD INPUT	61
Pitfall: Using <code>nextLine()</code> Following One of the Other <code>Scanner</code>	
Input Methods	63
USING THE <code>JOptionPane</code> CLASS FOR GUI INPUT	66
Using Input Dialog Boxes	66
Using Confirm Dialog Boxes	70
YOU DO IT	72
Working with Numeric Values	72
Accepting User Data	73
Performing Arithmetic	74
Experimenting with Java Programs	75
DON'T DO IT	76
KEY TERMS	77
CHAPTER SUMMARY	80
REVIEW QUESTIONS	81
EXERCISES	83
DEBUGGING EXERCISES	86
GAME ZONE	86
TOUGH QUESTIONS	86
UP FOR DISCUSSION	87

CONTENTS

CHAPTER 3 USING METHODS, CLASSES, AND OBJECTS	89
CREATING METHODS WITH ZERO, ONE, AND MULTIPLE PARAMETERS	90
Creating Methods That Require a Single Parameter	94
Creating Methods That Require Multiple Parameters	97
CREATING METHODS THAT RETURN VALUES	99
Calling a Method from Another Method	101
LEARNING ABOUT CLASS CONCEPTS	102
CREATING A CLASS	104
CREATING INSTANCE METHODS IN A CLASS	106
DECLARING OBJECTS AND USING THEIR METHODS	109
Understanding Data Hiding	110
ORGANIZING CLASSES	112
AN INTRODUCTION TO USING CONSTRUCTORS	114
UNDERSTANDING THAT CLASSES ARE DATA TYPES	116
YOU DO IT	118
Creating a Static Method That Requires No Arguments and Returns No Values	118
Calling a Static Method from Another Class	119
Creating a Static Method That Accepts Arguments and Returns Values	120
Creating a Class That Contains Instance Fields and Methods	122
Creating a Class That Instantiates Objects of Another Class	123
Adding a Constructor to a Class	124
Creating a More Complete Class	124
DON'T DO IT	125
KEY TERMS	125
CHAPTER SUMMARY	127
REVIEW QUESTIONS	128
EXERCISES	131
DEBUGGING EXERCISES	133
GAME ZONE	133
TOUGH QUESTIONS	134
UP FOR DISCUSSION	134
 CHAPTER 4 MORE OBJECT CONCEPTS	 135
UNDERSTANDING BLOCKS AND SCOPE	136
OVERLOADING A METHOD	142
LEARNING ABOUT AMBIGUITY	144
SENDING ARGUMENTS TO CONSTRUCTORS	147
OVERLOADING CONSTRUCTORS	148

CONTENTS

LEARNING ABOUT THE <code>this</code> REFERENCE	149
Using the <code>this</code> Reference to Make Overloaded Constructors More Efficient	152
USING <code>static</code> VARIABLES	154
USING CONSTANT FIELDS	156
USING AUTOMATICALLY IMPORTED, PREWRITTEN CONSTANTS AND METHODS	157
USING AN EXPLICITLY IMPORTED PREWRITTEN CLASS AND ITS METHODS	160
UNDERSTANDING COMPOSITION	164
A BRIEF LOOK AT NESTED AND INNER CLASSES	166
YOU DO IT	168
Demonstrating Scope	168
Overloading Methods	170
Creating a Constructor That Requires an Argument	171
Using an Explicitly Imported Prewritten Class	172
Creating an Interactive Application with a Timer	174
DON'T DO IT	176
KEY TERMS	176
CHAPTER SUMMARY	177
REVIEW QUESTIONS	178
EXERCISES	181
DEBUGGING EXERCISES	184
GAME ZONE	184
TOUGH QUESTIONS	185
UP FOR DISCUSSION	185
CHAPTER 5 MAKING DECISIONS	187
UNDERSTANDING DECISION MAKING	188
MAKING DECISIONS WITH THE <code>if</code> AND <code>if...else</code> STRUCTURES	190
Pitfall: Misplacing a Semicolon in an <code>if</code> Statement	191
Pitfall: Using the Assignment Operator Instead of the Equivalency Operator	192
Pitfall: Attempting to Compare Objects Using the Relational Operators	192
The <code>if...else</code> Structure	193
USING MULTIPLE STATEMENTS IN AN <code>if</code> OR <code>if...else</code> STRUCTURE	194
NESTING <code>if</code> AND <code>if...else</code> STATEMENTS	197
USING LOGICAL AND AND OR OPERATORS	199
MAKING ACCURATE AND EFFICIENT DECISIONS	202
Using AND and OR Appropriately	205
USING THE <code>switch</code> STATEMENT	206
USING THE CONDITIONAL AND NOT OPERATORS	209
Using the NOT Operator	210
UNDERSTANDING PRECEDENCE	211

CONTENTS

YOU DO IT	213
Using an <code>if...else</code>	213
Creating an Event Class to Use in a Decision-Making Application	215
Writing an Application that Uses the Event class	216
Using the <code>switch</code> Statement	218
DON'T DO IT	219
KEY TERMS	220
CHAPTER SUMMARY	221
REVIEW QUESTIONS	221
EXERCISES	224
DEBUGGING EXERCISES	229
GAME ZONE	229
TOUGH QUESTIONS	231
UP FOR DISCUSSION	232
 CHAPTER 6 LOOPING	 233
LEARNING ABOUT THE LOOP STRUCTURE	234
USING A <code>while</code> LOOP TO CREATE A DEFINITE LOOP	235
USING A <code>while</code> LOOP TO CREATE AN INDEFINITE LOOP	239
USING SHORTCUT ARITHMETIC OPERATORS	243
USING A <code>for</code> LOOP	246
LEARNING HOW AND WHEN TO USE A <code>do...while</code> LOOP	248
LEARNING ABOUT NESTED LOOPS	250
IMPROVING LOOP PERFORMANCE	252
Avoiding Unnecessary Operations	253
Considering the Order of Evaluation of Short-Circuit Operators	253
Comparing to Zero	254
Employing Loop Fusion	255
YOU DO IT	256
Writing a Loop to Validate Data Entries	256
Working with Prefix and Postfix Increment Operators	257
Working with Definite Loops	259
Working with Nested Loops	260
DON'T DO IT	261
KEY TERMS	262
CHAPTER SUMMARY	263
REVIEW QUESTIONS	264
EXERCISES	267
DEBUGGING EXERCISES	269

CONTENTS

GAME ZONE	269
TOUGH QUESTIONS	270
UP FOR DISCUSSION	271
CHAPTER 7 CHARACTERS, STRINGS, AND THE STRINGBUILDER	273
IDENTIFYING PROBLEMS THAT CAN OCCUR WHEN YOU MANIPULATE STRING DATA	274
MANIPULATING CHARACTERS	276
DECLARING A <code>String</code> OBJECT	278
COMPARING <code>String</code> VALUES	279
USING OTHER <code>String</code> METHODS	283
CONVERTING <code>Strings</code> TO NUMBERS	286
LEARNING ABOUT THE <code>StringBuilder</code> AND <code>StringBuffer</code> CLASSES	288
YOU DO IT	293
Using <code>String</code> Class Methods	293
Converting a <code>String</code> to an Integer	295
Using <code>StringBuilder</code> Methods	296
DON'T DO IT	297
KEY TERMS	297
CHAPTER SUMMARY	299
REVIEW QUESTIONS	300
EXERCISES	302
DEBUGGING EXERCISES	305
GAME ZONE	305
TOUGH QUESTIONS	307
UP FOR DISCUSSION	308
CHAPTER 8 ARRAYS	309
DECLARING AND INITIALIZING AN ARRAY	310
Initializing an Array	312
USING SUBSCRIPTS WITH AN ARRAY	313
DECLARING AN ARRAY OF OBJECTS	316
SEARCHING AN ARRAY FOR AN EXACT MATCH	318
SEARCHING AN ARRAY FOR A RANGE MATCH	321
PASSING ARRAYS TO AND RETURNING ARRAYS FROM METHODS	323
Returning an Array from a Method	326
MANIPULATING ARRAYS OF <code>Strings</code>	326
SORTING ARRAY ELEMENTS	328
Sorting Arrays of Objects	332

CONTENTS

USING TWO-DIMENSIONAL AND MULTIDIMENSIONAL ARRAYS	334
Using the <code>length</code> Field with a Two-Dimensional Array	336
Understanding Ragged Arrays	336
Using Multidimensional Arrays	336
USING THE <code>Arrays</code> CLASS	337
USING THE <code>ArrayList</code> CLASS	341
Understanding the Limitations of the <code>ArrayList</code> Class	343
YOU DO IT	344
Creating and Populating an Array	344
Initializing an Array	345
Using a <code>for</code> Loop to Access Array Elements	346
Creating Parallel Arrays to Eliminate Nested <code>if</code> Statements	346
Creating an Application with an Array of Objects	347
Creating an Interactive Application That Creates an Array of Objects	348
Passing an Array to a Method	350
Using <code>Arrays</code> Class Methods	351
DON'T DO IT	353
KEY TERMS	354
CHAPTER SUMMARY	355
REVIEW QUESTIONS	356
EXERCISES	359
DEBUGGING EXERCISES	363
GAME ZONE	363
TOUGH QUESTIONS	367
UP FOR DISCUSSION	367
 CHAPTER 9 INTRODUCTION TO INHERITANCE	 369
LEARNING ABOUT THE CONCEPT OF INHERITANCE	370
EXTENDING CLASSES	374
OVERRIDING SUPERCLASS METHODS	376
UNDERSTANDING HOW CONSTRUCTORS ARE CALLED DURING INHERITANCE	377
USING SUPERCLASS CONSTRUCTORS THAT REQUIRE ARGUMENTS	379
ACCESSING SUPERCLASS METHODS	380
Comparing <code>this</code> and <code>super</code>	382
LEARNING ABOUT INFORMATION HIDING	382
METHODS YOU CANNOT OVERRIDE	385
A Subclass Cannot Override <code>static</code> Methods in Its Superclass	385
A Subclass Cannot Override <code>final</code> Methods in Its Superclass	388
A Subclass Cannot Override Methods in a <code>final</code> Superclass	390

CONTENTS

YOU DO IT	391
Creating a Superclass and an Application to Use It	391
Creating a Subclass and an Application to Use It	393
Creating a Subclass Method That Overrides a Superclass Method	395
Understanding the Role of Constructors in Inheritance	397
Inheritance When the Superclass Requires Constructor Arguments	398
Accessing an Overridden Superclass Method from Within a Subclass	401
DON'T DO IT	402
KEY TERMS	402
CHAPTER SUMMARY	403
REVIEW QUESTIONS	404
EXERCISES	407
DEBUGGING EXERCISES	410
GAME ZONE	410
TOUGH QUESTIONS	411
UP FOR DISCUSSION	412
 CHAPTER 10 ADVANCED INHERITANCE CONCEPTS	 413
CREATING AND USING ABSTRACT CLASSES	414
USING DYNAMIC METHOD BINDING	418
Using a Superclass as a Method Parameter Type	419
CREATING ARRAYS OF SUBCLASS OBJECTS	420
USING THE <code>Object</code> CLASS AND ITS METHODS	422
Using the <code>toString()</code> Method	423
Using the <code>equals()</code> Method	425
USING INHERITANCE TO ACHIEVE GOOD SOFTWARE DESIGN	428
CREATING AND USING INTERFACES	429
Creating Interfaces to Store Related Constants	434
CREATING AND USING PACKAGES	435
YOU DO IT	437
Creating an Abstract Class	437
Extending an Abstract Class	438
Extending an Abstract Class with a Second Subclass	440
Instantiating Objects from Subclasses	441
Using Object References	442
Overriding the <code>Object</code> Class <code>equals()</code> Method	444
Eliminating Duplicate User Entries	445
Creating a Package	446
DON'T DO IT	449
KEY TERMS	449