LEARNING MADE EASY

7th Edition

Java®

ALL-IN-ONE

for dummies®

A Wiley Brand

8 Books in one!

Doug Lowe

Wrote his first program on a computer
with less than 1K of memory

Java®

ALL-IN-ONE

7th Edition

by Doug Lowe

for dummies®
A Wiley Brand

Java® All-in-One For Dummies®, 7th Edition

Published by: **John Wiley & Sons, Inc.,** 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

Copyright © 2023 by John Wiley & Sons, Inc., Hoboken, New Jersey

Media and software compilation copyright © 2023 by John Wiley & Sons, Inc. All rights reserved.

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permissions.

Trademarks: Wiley, For Dummies, the Dummies Man logo, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and may not be used without written permission. Java is a registered trademark of Oracle America, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: WHILE THE PUBLISHER AND AUTHORS HAVE USED THEIR BEST EFFORTS IN PREPARING THIS WORK, THEY MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES REPRESENTATIVES, WRITTEN SALES MATERIALS OR PROMOTIONAL STATEMENTS FOR THIS WORK. THE FACT THAT AN ORGANIZATION, WEBSITE, OR PRODUCT IS REFERRED TO IN THIS WORK AS A CITATION AND/OR POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE PUBLISHER AND AUTHORS ENDORSE THE INFORMATION OR SERVICES THE ORGANIZATION, WEBSITE, OR PRODUCT MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING PROFESSIONAL SERVICES. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR YOUR SITUATION. YOU SHOULD CONSULT WITH A SPECIALIST WHERE APPROPRIATE. FURTHER, READERS SHOULD BE AWARE THAT WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ. NEITHER THE PUBLISHER NOR AUTHORS SHALL BE LIABLE FOR ANY LOSS OF PROFIT OR ANY OTHER COMMERCIAL DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR OTHER DAMAGES.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit https://hub.wiley.com/community/support/dummies.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2022950700

ISBN 978-1-119-98664-5 (pbk); ISBN 978-1-119-98665-2 (ebk); ISBN 978-1-119-98666-9 (ebk)

Contents at a Glance

Introduction 1

Book 1: Java Basics 5

CHAPTER 1: Welcome to Java 7

CHAPTER 2: Installing and Using Java Tools 17

CHAPTER 3: Working with TextPad 31

Book 2: Programming Basics 37

CHAPTER 1: Java Programming Basics 39

CHAPTER 2: Working with Variables and Data Types 59

CHAPTER 3: Working with Numbers and Expressions 97

CHAPTER 4: Making Choices 127

CHAPTER 5: Going Around in Circles (or, Using Loops) 149

CHAPTER 6: Pulling a Switcheroo 177

CHAPTER 7: Adding Some Methods to Your Madness 195

CHAPTER 8: Handling Exceptions 215

Book 3: Object-Oriented Programming 237

CHAPTER 1: Understanding Object-Oriented Programming 239

CHAPTER 2: Making Your Own Classes 253

CHAPTER 3: Working with Statics 273

CHAPTER 4: Using Subclasses and Inheritance 283

CHAPTER 5: Using Abstract Classes and Interfaces 305

CHAPTER 6: Using the Object and Class Classes 327

CHAPTER 7: Using Inner Classes and Anonymous Classes 355

CHAPTER 8: Working with Packages and the Java Module System 365

Book 4: Strings and Other Data Types 387

CHAPTER 1: Working with Strings 389

CHAPTER 2: Using Regular Expressions 411

CHAPTER 3: Working with Dates and Times 429

CHAPTER 4: Using the BigDecimal Class 451

Book 5: Data Structures 471

CHAPTER 1: Introducing Data Structures 473

CHAPTER 2: Using Arrays 503

CHAPTER 3: Using the ArrayList Class 535

CHAPTER 4: Using the LinkedList Class. 549
CHAPTER 5: Creating Generic Collection Classes 561
CHAPTER 6: Using Maps and Trees 575

Book 6: Algorithms 595
CHAPTER 1: Introducing Algorithms 597
CHAPTER 2: Using Recursion 613
CHAPTER 3: Sorting 625
CHAPTER 4: Searching 637

Book 7: Programming Techniques. 657
CHAPTER 1: Programming Threads. 659
CHAPTER 2: Using Functional Programming and Lambda Expressions 689
CHAPTER 3: Consuming Web Services with HttpClient 705

Book 8: JavaFX. 727
CHAPTER 1: Hello, JavaFX! 729
CHAPTER 2: Handling Events 751
CHAPTER 3: Setting the Stage and Scene Layout 769
CHAPTER 4: Using Layout Panes to Arrange Your Scenes. 791
CHAPTER 5: Getting Input from the User 825
CHAPTER 6: Choosing from a List 841

Index 869

Table of Contents

INTRODUCTION 1

About This Book 2

Foolish Assumptions 2

Icons Used in This Book 3

Beyond the Book 4

Where to Go from Here 4

BOOK 1: JAVA BASICS 5

CHAPTER 1: Welcome to Java 7

What Is Java, and Why Is It So Great? 8

Platform independence 8

Object orientation 9

The Java API 10

Important Features of the Java Language 11

Type checking 11

Exception handling 12

Java Version Insanity 13

What's in a Name? 15

CHAPTER 2: Installing and Using Java Tools 17

Downloading and Installing the Java Development Kit 18

Downloading the JDK 18

Installing the JDK 19

Perusing the JDK folders 19

Setting the JAVA_HOME and path variables 20

Confirming Your Java Version 22

Using Java's Command-Line Tools 23

Compiling a program 23

Compiling more than one file 24

Using Java compiler options 25

Running a Java program 27

Using the javap command 28

Using Java Documentation 29

CHAPTER 3: Working with TextPad 31

Downloading and Installing TextPad 31

Editing Source Files 33

Compiling a Program 34

Running a Java Program 35

BOOK 2: PROGRAMMING BASICS 37
CHAPTER 1: **Java Programming Basics** 39
Looking at the Venerable Hello, World! Program 40
Dealing with Keywords 43
Working with Statements 46
Types of statements 46
White space 47
Working with Blocks 48
Creating Identifiers 49
Crafting Comments 50
End-of-line comments. 50
Traditional comments. 51
Introducing Object-Oriented Programming 51
Understanding classes and objects 51
Understanding static methods 52
Creating an object from a class 53
Viewing a program that uses an object 54
So what's the difference? 56
Importing Java API Classes 57

CHAPTER 2: **Working with Variables and Data Types**. 59
Declaring Variables 60
Declaring two or more variables in one statement 61
Declaring class variables 61
Declaring instance variables 62
Declaring local variables. 63
Initializing Variables 64
Initializing variables with assignment statements 65
Initializing variables with initializers 66
Using Final Variables (Constants) 66
Working with Primitive Data Types. 67
Integer types 69
Floating-point types. 70
The char type 71
The Boolean type. 73
Using wrapper classes 74
Using reference types. 74
Using inferred variable types. 76
Working with Strings. 77
Declaring and initializing strings 78
Combining strings 78
Converting primitives to strings 79
Converting strings to primitives 80

Converting and Casting Numeric Data 81
Automatic conversions 81
Type casting 82
Thinking Inside the Box 83
Understanding Scope 84
Shadowing Variables 85
Printing Data with System.out 87
Using standard input and output streams 87
Using System.out and System.err 88
Getting Input with the Scanner Class 89
Importing the Scanner class 90
Declaring and creating a Scanner object 90
Getting input 91
Getting Input with the JOptionPane Class 93
Using enum to Create Your Own Data Types 94

CHAPTER 3: **Working with Numbers and Expressions** 97
Working with Arithmetic Operators 97
Dividing Integers 100
Combining Operators 102
Using the Unary Plus and Minus Operators .103
Using Increment and Decrement Operators .104
Using the Assignment Operator 106
Using Compound Assignment Operators 108
Using the Math Class 109
Using constants of the Math class 110
Working with mathematical functions 111
Creating random numbers 114
Rounding functions 117
Formatting Numbers 119
Recognizing Weird Things about Java Math .121
Integer overflow 122
Floating-point weirdness 122
Division by zero 123

CHAPTER 4: **Making Choices** 127
Using Simple Boolean Expressions 128
Using if Statements 130
Simple if statements 130
if-else statements 132
Nested if statements 134
else-if statements 138
Using Mr. Spock's Favorite Operators (Logical Ones, of Course) 140
Using the ! operator 140
Using the & and && operators 141

Using the | and || operators 143
Using the ^ operator 144
Combining logical operators 145
Using the Conditional Operator 146
Comparing Strings. 147

CHAPTER 5: **Going Around in Circles (or, Using Loops)** 149
Using Your Basic while Loop 150
The while statement 150
A counting loop 150
Breaking Out of a Loop. 152
Looping Forever. 152
Letting the user decide when to quit 154
Letting the user decide in another way 155
Using the continue Statement. 156
Running do-while Loops. 157
Validating Input from the User 159
Using the Famous for Loop 161
Understanding the formal format of the for loop 162
Scoping out the counter variable 164
Counting even numbers. 165
Counting backward 166
Using for loops without bodies 167
Ganging up your expressions 168
Omitting expressions 169
Breaking and continuing your for loops. 170
Nesting Your Loops. 171
A simple nested for loop 171
A guessing game 172

CHAPTER 6: **Pulling a Switcheroo** 177
Battling else-if Monstrosities 177
Viewing an example else-if program 178
Creating a better version of the example program 179
Using the switch Statement 181
Viewing a boring switch example, complete with flowchart 182
Putting if statements inside switch statements 184
Creating Character Cases 185
Matching Two or More Values in a Single Case Group 186
Intentionally Leaving Out a Break Statement .187
Switching with Strings. 190
Assigning a Value with a Switch Statement 191

CHAPTER 7: **Adding Some Methods to Your Madness** 195
The Joy of Methods 196
The Basics of Making Methods 196
An example. 198
Another example 199
Methods That Return Values 201
Declaring the method's return type 201
Using the return statement to return the value 202
Using a method that returns a type 203
You gotta have a proper return statement 203
Trying another version of the guessing-game program 205
Methods That Take Parameters 208
Declaring parameters 209
Scoping out parameters 210
Understanding pass-by-value 210
Trying yet another version of the guessing-game program 211

CHAPTER 8: **Handling Exceptions** 215
Understanding Exceptions. 216
Witnessing an exception. 217
Finding the culprit 218
Catching Exceptions 219
A simple example 220
Another example 220
Handling Exceptions with a Preemptive Strike 222
Catching All Exceptions at Once 224
Displaying the Exception Message 225
Using a finally Block 226
Handling Checked Exceptions 229
Viewing the catch-or-throw compiler error 229
Catching FileNotFoundException 230
Throwing the FileNotFoundException 231
Throwing an exception from main 232
Swallowing exceptions 232
Throwing Your Own Exceptions 234
Using the try-with-resources Statement 235

BOOK 3: OBJECT-ORIENTED PROGRAMMING 237

CHAPTER 1: **Understanding Object-Oriented Programming** . . . 239
What Is Object-Oriented Programming? 240
Understanding Objects. 241
Objects have identity 241
Objects have type 242

Objects have state 243
Objects have behavior 244
Understanding the Life Cycle of an Object 245
Working with Related Classes 246
Inheritance 246
Interfaces 247
Designing a Program with Objects 248
Diagramming Classes with UML 249
Drawing classes 250
Drawing arrows 251

CHAPTER 2: **Making Your Own Classes** 253
Declaring a Class 254
Picking class names 254
Knowing what goes in the class body 255
Seeing where classes live 256
Working with Members 257
Understanding fields 257
Understanding instance methods 258
Understanding visibility 259
Using Getters and Setters 260
Overloading Methods 262
Creating Constructors 264
Creating basic constructors 264
Creating default constructors 265
Calling other constructors 267
Finding More Uses for the this Keyword 268
Using Initializers 270
Using Records 271

CHAPTER 3: **Working with Statics** 273
Understanding Static Fields and Methods 273
Working with Static Fields 275
Using Static Methods 276
Counting Instances 277
Preventing Instances 280
Using Static Initializers 281

CHAPTER 4: **Using Subclasses and Inheritance** 283
Introducing Inheritance 283
Motorcycles, trains, and automobiles 285
Game play 285
A businesslike example 286
Inheritance hierarchies 286
Creating Subclasses 287

Overriding Methods 289
Protecting Your Members 290
Using the this and super Keywords in Your Subclasses 291
Understanding Inheritance and Constructors 292
Using the final Keyword 294
Final methods 294
Final classes 294
Casting Up and Down 295
Determining an Object's Type 297
Poly What? 298
Creating Custom Exceptions 301
Tracing the Throwable hierarchy 301
Creating an exception class 302
Throwing a custom exception 303

CHAPTER 5: Using Abstract Classes and Interfaces 305
Using Abstract Classes 305
Using Interfaces 308
Creating a basic interface 309
Implementing an interface 310
Using an interface as a type 311
More Things You Can Do with Interfaces 312
Adding fields to an interface 312
Extending interfaces 313
Using interfaces for callbacks 314
Using Additional Interface Method Types 318
Two Interfaces That Enable Java Language Features 320
Sealing Your Classes 321

CHAPTER 6: Using the Object and Class Classes 327
The Mother of All Classes: Object 327
Every object is an Object 328
Object as a type 328
Methods of the Object class. 329
Primitives aren't objects 330
The toString Method 331
Using toString. 331
Overriding toString 332
The equals Method 334
Using equals 335
Overriding the equals method 336
The clone Method 340
Implementing the clone method. 340
Using clone to create a shallow copy 343
Creating deep copies 345

The Class Class 350
Using the instanceof Operator 352

CHAPTER 7: **Using Inner Classes and Anonymous Classes** 355
Declaring Inner Classes 356
Understanding inner classes 356
Viewing an example 357
Using Static Inner Classes 359
Using Anonymous Inner Classes 361
Creating an anonymous class 361
Creating a program with an anonymous class 363

CHAPTER 8: **Working with Packages and the Java Module System** 365
Working with Packages 366
Importing classes and packages 366
Creating your own packages 367
An example 369
Putting Your Classes in a JAR File 370
jar command-line options 371
Archiving a package 372
Adding a jar to your classpath 373
Running a program directly from an archive 374
Using Javadoc to Document Your Classes 375
Adding Javadoc comments 375
Using the javadoc command 378
Viewing Javadoc pages 379
Using the Java Module System 381
Understanding modules 382
The module-info.java file 383
Setting up folders for a module 384
Compiling a module 385
Creating a modular JAR file 385
Running a modular JAR file 386

BOOK 4: STRINGS AND OTHER DATA TYPES 387

CHAPTER 1: **Working with Strings** 389
Reviewing Strings 390
Using the String Class 392
Finding the length of a string 395
Making simple string modifications 395
Extracting characters from a string 396
Extracting substrings from a string 397

Splitting a string 399
Replacing parts of a string 400
Stripping a string 401
Determining Whether a String Is Empty 402
Using the StringBuilder and StringBuffer Classes 404
Creating a StringBuilder object 405
Using StringBuilder methods 405
Viewing a StringBuilder example 407
Using the CharSequence Interface 408
Using Text Blocks 408

CHAPTER 2: **Using Regular Expressions** 411
Creating a Program for Experimenting with Regular Expressions . . . 412
Performing Basic Character Matching 415
Matching single characters 415
Using predefined character classes 415
Using custom character classes 418
Using ranges 419
Using negation 420
Matching multiple characters 420
Using escapes 422
Using parentheses to group characters 422
Using the pipe symbol 424
Using Regular Expressions in Java Programs 425
Understanding the String problem 425
Using regular expressions with the String class 426
Using the Pattern and Matcher classes 427

CHAPTER 3: **Working with Dates and Times** 429
Pondering How Time Is Represented 430
Picking the Right Date and Time Class for Your Application 431
Using the now Method to Create a Date-Time Object 432
Using the parse Method to Create a Date-Time Object 434
Using the of Method to Create a Date-Time Object 435
Using the Month enumeration 436
Using the ZoneId class 437
Using the ZoneOffset class 437
Looking Closer at the LocalDate Class 438
Extracting Information about a Date 440
Comparing Dates 441
Calculating with Dates 442
Formatting Dates 444
Looking at a Fun Birthday Calculator 446

CHAPTER 4: **Using the BigDecimal Class** 451

Seeing Why Java Can't Add 451
    Double trouble 453
    Another example 454
BigDecimal to the Rescue! 456
Creating BigDecimal Objects 456
Doing BigDecimal Arithmetic 458
Rounding BigDecimal Values 461
    The fundamentals of scale 462
    How to round 463
Comparing BigDecimal Values. 465
Converting BigDecimals to Strings 467
Revisiting Sales Tax 467

BOOK 5: DATA STRUCTURES 471

CHAPTER 1: **Introducing Data Structures**. 473

Imagining Models of Computation 474
    Musing about word RAM memory 475
    Considering the CPU in the word RAM model. 476
Defining Data Structures 478
Quantifying Performance. 480
Analyzing Arrays 482
Looking at Linked Lists 485
Doubling Down with Doubly Linked Lists 492
Surmising Stacks, Queues, and Deques. 494
Musing about Maps 495
Checking Out Hash Maps. 496
Tackling Trees 498
Checking Out Binary Trees. 501

CHAPTER 2: **Using Arrays** 503

Understanding Arrays. 503
Creating Arrays 504
Initializing an Array 506
Using for Loops with Arrays. 506
Solving Homework Problems with Arrays 507
Using the Enhanced for Loop 510
Using Arrays with Methods 511
Using Varargs. 512
Using Two-Dimensional Arrays 513
    Creating a two-dimensional array. 514
    Accessing two-dimensional array elements 515
    Initializing a two-dimensional array 516

Using jagged arrays 517
Going beyond two dimensions 519
Working with a Fun but Complicated Example: A Chessboard. 520
Using the Arrays Class 527
Filling an array 529
Copying an array 529
Sorting an array 531
Searching an array. 531
Comparing arrays 532
Converting arrays to strings. 532

CHAPTER 3: **Using the ArrayList Class** 535
Understanding the ArrayList Class 536
Creating an ArrayList Object 539
Adding Elements 540
Accessing Elements 541
Printing an ArrayList 542
Using an Iterator 543
Updating Elements 545
Deleting Elements 546

CHAPTER 4: **Using the LinkedList Class** 549
Understanding the LinkedList Class 550
Creating a LinkedList. 554
Adding Items to a LinkedList 555
Retrieving Items from a LinkedList 557
Updating LinkedList Items 558
Removing LinkedList Items 559

CHAPTER 5: **Creating Generic Collection Classes** 561
Why Generics? 561
Creating a Generic Class. 563
A Generic Stack Class 565
Using Wildcard-Type Parameters 568
A Generic Queue Class 570

CHAPTER 6: **Using Maps and Trees**. 575
Mastering Maps 575
Using the HashMap Class. 576
Creating a hash map 580
Adding elements to a hash map 580
Retrieving elements from a hash map 582
Replacing entries from a hash map 583

Removing Entries from a Hash Map 584
Treating a hash map as a collection 585
Using lambda expressions with maps 587
Using the TreeMap Class 587
Introducing red-black trees 588
Looking at TreeMap methods 590
Creating and using a TreeMap. 592
More ways to play with a TreeMap 593

BOOK 6: ALGORITHMS

BOOK 6: ALGORITHMS 595
CHAPTER 1: Introducing Algorithms. 597
What Is an Algorithm?. 597
Classifying Algorithms. 599
Brute-force algorithms 599
Divide-and-conquer algorithms. 601
Backtracking algorithms 602
Dynamic algorithms 604
Some Fun Algorithm Challenges 606
Fizz buzz 606
Roman numerals 607
Spelling out dollar amounts. 609
Determining if a tic-tac-toe game is over .609
Scheduling a tournament. 610
CHAPTER 2: Using Recursion. 613
Calculating the Classic Factorial Example 613
The nonrecursive solution 614
The recursive solution. 614
Displaying Directories. 616
Playing the Towers of Hanoi 620
CHAPTER 3: Sorting. 625
Looking at the Bubble Sort Algorithm 625
Introducing the Quicksort Algorithm 629
Using the sort method 631
Using the partition method 631
Putting it all together 634
CHAPTER 4: Searching. 637
Creating a Data Structure Worth Searching 638
Looking at Linear Searching. 643
Using a Binary Search 647

BOOK 7: PROGRAMMING TECHNIQUES 657

CHAPTER 1: **Programming Threads** 659

Understanding Threads 660
Creating a Thread 661
Understanding the Thread class 661
Extending the Thread class 662
Creating and starting a thread 664
Implementing the Runnable Interface 664
Using the Runnable interface 665
Creating a class that implements Runnable 665
Using the CountDownApp class 667
Creating Threads That Work Together 669
Using an Executor 673
Synchronizing Methods 675
Creating a Lock 680
Coping with Threadus Interruptus 681
Finding out whether you've been interrupted 681
Aborting the countdown 683

CHAPTER 2: **Using Functional Programming and Lambda Expressions** 689

Introducing Functional Programming 690
Introducing Functional Interfaces 691
Using Lambda Expressions 693
Passing Parameters via Lambda Expressions .694
Using Block Lambda Expressions 697
Using the java.util.function Package 698

CHAPTER 3: **Consuming Web Services with HttpClient** 705

Introducing Web Services 705
Understanding HTTP 708
Diving into HTTP 709
Looking at a simple HTTP exchange 713
Getting Started with Java's HTTP Client Library 714
HttpClient 715
HttpRequest 716
HttpResponse 717
Using the send method 720
Putting It All Together 721
The HTTP Tester Program 723

BOOK 8: JAVAFX 727

CHAPTER 1: **Hello, JavaFX!** 729

Perusing the Possibilities of JavaFX. 730
Getting Ready to Run JavaFX 732
Looking at a Simple JavaFX Program 734
Importing JavaFX Packages 736
Extending the Application Class 737
Launching the Application 738
Overriding the start Method 739
Creating a Button 740
Handling an Action Event 741
Creating a Layout Pane. 743
Making a Scene 744
Setting the Stage 745
Examining the Click Counter Program 746

CHAPTER 2: **Handling Events**. 751

Examining Events 752
Handling Events. 753
Implementing the EventHandler Interface 755
Handling Events with Inner Classes 758
Handling Events with Anonymous Inner Classes 761
Using Lambda Expressions to Handle Events. 763

CHAPTER 3: **Setting the Stage and Scene Layout**. 769

Examining the Stage Class 770
Examining the Scene Class. 773
Switching Scenes 775
Creating an Alert Box 780
Exit, Stage Right 784
Creating a Close button 785
Handling the CloseRequest event. 787
Putting it all together 788

CHAPTER 4: **Using Layout Panes to Arrange Your Scenes** 791

Working with Layout Panes 792
Introducing five JavaFX layout panes 792
Creating layout panes. 792
Combining layout panes. 794
Using the HBox Layout 794

Spacing Things Out 796
Adding Space with Margins 798
Adding Space by Growing Nodes 799
Using the VBox Layout 801
Aligning Nodes in a Layout Pane. 803
Using the Flow Layout. 804
Using the Border Layout 808
Using the GridPane Layout 810
- Sketching out a plan 811
- Creating a grid pane 812
- Working with grid pane constraints 815
- Examining a grid pane example 817

CHAPTER 5: **Getting Input from the User** 825

Using Text Fields 826
Validating Numeric Data. 833
Using Check Boxes 835
Using Radio Buttons 837

CHAPTER 6: **Choosing from a List** 841

Using Choice Boxes. 841
- Creating a choice box 842
- Setting a default value 844
- Getting the selected item 845

Working with Observable Lists 845
Listening for Selection Changes 848
Using Combo Boxes 850
- Creating combo boxes 851
- Getting the selected item 852
- Handling combo box events 853

Using List Views 854
- Creating a list view. 856
- Getting the selected items 856

Using Tree Views 857
- Building a tree 858
- Creating a TreeView control. 861
- Getting the selected node 863
- Looking at a complete program that uses a tree view. 864

INDEX. 869

Introduction

Welcome to *Java All-in-One For Dummies,* 7th Edition — the one Java book that's designed to replace an entire shelf full of the dull, tedious titles you'd otherwise have to buy. This book contains all the basic information you need to know to get going with Java programming, starting with writing statements and using variables and ending with techniques for writing programs that use advanced Java features. Along the way, you find plenty of not-so-basic information about programming user interfaces, working with classes and objects, data structures, and algorithms.

You can (and probably should, eventually) buy separate books on each of these topics. It won't take long before your bookshelf is bulging with 10,000 or more pages of detailed information about every imaginable nuance of Java programming. But before you're ready to tackle each of those topics in depth, you need to get a bird's-eye picture. That's what this book is about.

And if you already *own* 10,000 pages or more of Java information, you may be overwhelmed by the amount of detail and wonder, "Do I really need to *read* 1,200 pages about JSP just to create a simple web page? And do I really *need* a six-pound book on JavaFX?" Truth is, most 1,200-page programming books have about 200 pages of really useful information — the kind you use every day — and about 1,000 pages of excruciating details that apply mostly if you're writing guidance-control programs for nuclear missiles or trading systems for the New York Stock Exchange.

The basic idea here is that I've tried to wring out the 100-or-so most useful pages of information on these different Java programming topics: setup and configuration, basic programming, object-oriented programming, advanced programming techniques, JavaFX, file and database programming, web programming, and animation and game programming. Thus you get a nice, trim book.

So whether you're just getting started with Java programming or you're a seasoned pro, you've found the right book.

About This Book

Java All-in-One For Dummies, 7th Edition, is a reference for all the great things (and maybe a few not-so-great things) that you may need to know when you're writing Java programs. You can, of course, buy a huge 1,200-page book on each of the programming topics covered in this book. But then, who would carry them home from the bookstore for you? And where would you find the shelf space to store them? And when will you find the time to read them?

In this book, all the information you need is conveniently packaged for you in-between one set of covers. And all of the information is current for the newest release of Java, known as JDK 19. This book doesn't pretend to be a comprehensive reference for every detail on every possible topic related to Java programming. Instead, it shows you how to get up and running fast so that you have more time to do the things you really want to do. Designed using the easy-to-follow *For Dummies* format, this book helps you get the information you need without laboring to find it.

Java All-in-One For Dummies, 7th Edition, is a big book made up of eight smaller books — minibooks, if you will. Each of these minibooks covers the basics of one key element of programming, such as installing Java and compiling and running programs, or using basic Java statements, or using JavaFX to write GUI applications.

Whenever one big thing is made up of several smaller things, confusion is always a possibility. That's why this book has multiple access points. At the beginning is a detailed table of contents that covers the entire book. Then each minibook begins with a minitable of contents that shows you at a miniglance what chapters are included in that minibook. Useful running heads appear at the top of each page to point out the topic discussed on that page. And handy thumb tabs run down the side of the pages to help you find each minibook quickly. Finally, a comprehensive index lets you find information anywhere in the entire book.

Foolish Assumptions

You and I have never met, so it is difficult for me to make any assumptions about why you are interested in this book. However, let's start with a few basic assumptions:

- **You own or have access to a relatively modern computer.** The examples were created on a Windows computer, but you can learn to program in Java just as easily on a Mac or Linux computer.

- **You're an experienced computer user.** In other words, I assume that you know the basics of using your computer, such as starting programs and working with the file system.
- **You're interested in learning how to write programs in the Java language.** Since that's what this book teaches, it's a fair assumption.

I do *not* make any assumptions about any previous programming experience in Java or in any other programming language. Nor do I make any assumptions about *why* you want to learn about Java programming. There are all sorts of valid reasons for learning Java. Some want to learn Java for professional reasons; maybe you want to become a professional Java programmer, or maybe you are a C# or C++ programmer who occasionally needs to work in Java. On the other hand, maybe you think programming in Java would make an interesting hobby.

Regardless of your motivation, I *do* assume that you are a reasonably intelligent person. You don't have to have a degree in advanced physics, or a degree in anything at all for that matter, to master Java programming. All you have to be is someone who wants to learn and isn't afraid to try.

Icons Used in This Book

Like any *For Dummies* book, this book is chock-full of helpful icons that draw your attention to items of particular importance. You find the following icons throughout this book:

WARNING

Danger, Will Robinson! This icon highlights information that may help you avert disaster.

REMEMBER

Did I tell you about the memory course I took?

TIP

Pay special attention to this icon; it lets you know that some particularly useful tidbit is at hand.

TECHNICAL STUFF

Hold it — overly technical stuff is just around the corner. Obviously, because this is a programming book, almost every paragraph of the next 900 or so pages could get this icon. So I reserve it for those paragraphs that go into greater depth, down into explaining how something works under the covers — probably deeper than you really need to know to use a feature, but often enlightening.

Beyond the Book

In addition to the material in the print or e-book you're reading right now, this product also comes with some access-anywhere goodies on the web. Check out the free Cheat Sheet for more on Java. To get this Cheat Sheet, simply go to `www.dummies.com` and type **Java All-in-One For Dummies Cheat Sheet** in the Search box.

Visit `www.dummies.com/go/javaaiofd7e` to dive even deeper into Java. You can find and download the code used in the book at that link. You can also download a bonus minibook covering how to use Java with files and databases.

Where to Go from Here

This isn't the kind of book you pick up and read from start to finish, as if it were a cheap novel. If I ever see you reading it at the beach, I'll kick sand in your face. Beaches are for reading romance novels or murder mysteries, not programming books. Although you could read straight through from start to finish, this book is a reference book, the kind you can pick up, open to just about any page, and start reading. You don't have to memorize anything in this book. It's a "need-to-know" book: You pick it up when you need to know something. Need a reminder on the constructors for the `ArrayList` class? Pick up the book. Can't remember the goofy syntax for anonymous inner classes? Pick up the book. After you find what you need, put the book down and get on with your life.

1

Java Basics

Contents at a Glance

CHAPTER 1: **Welcome to Java** 7

CHAPTER 2: **Installing and Using Java Tools** 17

CHAPTER 3: **Working with TextPad** 31

IN THIS CHAPTER

- » Finding out about programming
- » Scoping out Java
- » Understanding Java's incomprehensible version numbers

Chapter 1

Welcome to Java

This chapter is a gentle introduction to the world of Java. In the next few pages, you find out what Java is, where it came from, and where it's going. You also discover some of the unique strengths of Java, as well as some of its weaknesses. Also, you see how Java compares with other popular programming languages such as C, C++, and C#.

By the way, I assume in this chapter that you have at least enough background to know what computer programming is all about. That doesn't mean that I assume you're an expert or professional programmer. It just means that I don't take the time to explain such basics as what a computer program is, what a programming language is, and so on. If you have absolutely no programming experience, I suggest that you pick up a copy of *Java For Dummies,* 7th Edition, or *Beginning Programming with Java For Dummies,* 5th Edition, both by Barry Burd (Wiley).

Throughout this chapter, you find little snippets of Java program code, plus a few snippets of code written in other languages, including C, C++, and Basic. If you don't have a clue what this code means or does, don't panic. I just want to give you a feel for what Java programming looks like and how it compares with programming in other languages.

TIP

All the code listings used in this book are available for download at `www.dummies.com/go/javaaiofd7e`.

What Is Java, and Why Is It So Great?

Java is a programming language in the tradition of C and C++. As a result, if you have any experience with C or C++, you'll often find yourself in familiar territory as you discover the various features of Java. (For more information about the similarities and differences between Java and C or C++, see the section "Java versus Other Languages," later in this chapter.)

Java differs from other programming languages in a couple of significant ways, however. I point out the most important differences in the following sections.

Platform independence

One of the main reasons Java is so popular is its *platform independence,* which simply means that Java programs can be run on many types of computers.

Before Java, other programming languages promised platform independence by providing compatible compilers for different platforms. (A *compiler* is the program that translates programs written in a programming language into a form that can actually run on a computer.) The idea was that you could compile different versions of the programs for each platform. Unfortunately, this idea never really worked. The compilers were never identical on each platform; each had its own little nuances. As a result, you had to maintain a different version of your program for each platform you wanted to support.

Java's platform independence isn't based on providing compatible compilers for different platforms. Instead, Java is based on the concept of a *virtual machine* called the Java Virtual Machine (JVM). Think of the JVM as a hypothetical computer platform — a design for a computer that doesn't exist as actual hardware. Instead, the JVM simulates the operation of a hypothetical computer that is designed to run Java programs.

The Java compiler doesn't translate Java into the machine language of the computer that the program is running on. Instead, the compiler translates Java into the machine language of the JVM, which is called *bytecode.* Then the JVM runs the bytecode in the JVM.

When you compile a Java program, the runtime environment that simulates the JVM for the targeted computer type (Windows, Linux, macOS, and so on) is included with your compiled Java programs.

That's how Java provides platform independence — and believe it or not, it works pretty well. The programs you write run just as well on a PC running any version