

Contents

PREFACE	xvii
1 OBJECT-ORIENTED PROGRAMMING	1
1.1 Object-Oriented and Procedural Programming	2
<i>Top-Down Design and Procedural Programming</i> ,	3
<i>Problems with Top-Down Design</i> ,	3
<i>Classes and Objects</i> ,	4
<i>Fields and Methods</i> ,	5
<i>Instance versus Class Members</i> ,	6
<i>Class Design</i> ,	7
<i>Class and Object Relationships</i> ,	8
1.2 Classes and Abstract Data Types	9
<i>Information Hiding</i> ,	9
<i>Encapsulation</i> ,	10
<i>Abstract Data Types</i> ,	10

1.3	The Client/Server Model and Message Passing	12
	<i>The Client/Server Model</i> , 13	
	<i>Message Passing and Method Invocation</i> , 14	
1.4	Inheritance and Polymorphism	15
	<i>Inheritance</i> , 16	
	<i>Polymorphism</i> , 17	
	<i>Polymorphism and Recursion</i> , 18	
1.5	Interfaces and Components	21
	<i>Component Technology</i> , 23	
1.6	Object-Oriented Modeling and UML	24
	<i>Sample Models</i> , 25	
	<i>From UML Models to Code</i> , 28	
2	INTRODUCTORY PROGRAMS	32
2.1	A First Program	34
	<i>Source Files and Class Definitions</i> , 38	
2.2	Basic Programming Constructs	41
	<i>Finding the Smallest and Largest Integers</i> , 41	
	<i>The Convenience of the <code>import</code> Statement</i> , 43	
	<i>Generating Random Integers</i> , 44	
	<i>The <code>while</code> Loop</i> , 44	
	<i>The <code>if</code> Statement</i> , 45	
	<i>Compiling and Running the <code>BigAndSmall</code> Program</i> , 46	
	<i>Reading Input from a Disk File</i> , 46	
	<i>Filling and Sorting an Array</i> , 49	
	<i>Ensuring that a Program is Invoked Properly</i> , 50	
	<i>Constructing, Filling, and Sorting an Array</i> , 51	
	<i>Populating, Printing, and Sorting the Array</i> , 52	
	<i>Writing Output to a Disk File</i> , 53	
	<i>Determining the Best Strategy for a Game</i> , 55	
	<i>The <code>for</code> Loop</i> , 59	
	<i>Absolute Values and Remainders</i> , 60	
2.3	Strings	63
	<i>The <code>toString</code> Method</i> , 64	
	<i>A Test Client for the <code>String</code> Class</i> , 65	
	<i>String Conversion Methods</i> , 68	
	<i>Wrapper Classes and Type Conversions</i> , 69	
	<i>The <code>StringBuffer</code> Class</i> , 70	
2.4	Programmer-Defined Classes	72
	<i>Properties and <code>Get/Set</code> Methods</i> , 74	
2.5	Input and Output	75
	<i>Stream Input/Output</i> , 76	
	<i>The <code>IOException</code></i> , 77	

Binary Input: Counting the Bytes in a File, 78
Distinguishing Letters and Digits, 79
Echoing the Standard Input to the Standard Output, 81

- 2.6 Utility Classes 84
The StringTokenizer Class, 84
The Vector Class, 86
The Hashtable Class, 88
Options to the javac Compilation Command, 91

3 PROGRAMS AND PACKAGES 98

- 3.1 Program Types 99
The Java Virtual Machine, 99
Applications, 100
Applets, 100
Servlets, 101
Beans, 101
Summary of Program Types, 102
- 3.2 Review of Source Files and Class Files 103
Source Files with a public Class, 104
- 3.3 Packages 105
The import Statement, 105
Use of the import Statement, 108
Packages and Name Conflicts, 109
Default and Named Packages, 109
The package Statement, 110
The CLASSPATH Environment Variable, 112
Subdirectories as Subpackages, 113
- 3.4 Sample Application: Generating Nonnegative Integers 116
Problem, 116
Output, 116
Solution, 117
Java Implementation, 117
Discussion, 117
Review of Constructors and Methods, 118
Compiling and Running the Sample Application, 119

4 LANGUAGE FUNDAMENTALS 130

- 4.1 Identifiers, Variables, and Values 131
Variables and Values, 132
Fields and Default Values, 134
The null Default Value for Object References, 135
Local Variables, 135
final Variables, 136

- 4.2 Constructors and Methods 138
 - Constructor and Method Overloading*, 139
 - The Syntax of Methods*, 140
 - Distinguishing Methods from One Another*, 141
- 4.3 Primitive Data Types and Operators 143
 - Integer Types*, 144
 - Floating-Point Types*, 146
 - Cast Operations*, 147
 - Arithmetic Operators*, 148
 - Shift and Bit Operators*, 149
 - Assignment, Increment, and Decrement Operators*, 150
 - Arithmetic Operators for Floating-Point Numbers*, 150
 - The char Type*, 151
 - The Problem with Mixing char Type and Integer Types*, 152
 - Arithmetic Operations on the char Type*, 152
 - String Literals*, 153
 - The boolean Type*, 153
 - Relational Operators*, 154
 - Cautionary Notes on the Equality Operator ==*, 155
 - The Equality Operator == and Object References*, 156
 - Logical Operators*, 157
 - The instanceof Operator*, 159
- 4.4 Arrays 161
 - Bounds Checking*, 163
 - Arrays of Arrays*, 163
 - Arrays as Return Types*, 165
- 4.5 Control Structures 166
 - Blocks*, 166
 - Fields, Parameters, and Local Variables with the Same Name*, 169
 - Nested Blocks*, 170
 - The if Statement*, 170
 - The switch Statement*, 172
 - The while and do while Loops*, 174
 - The for Loop*, 176
- 4.6 Sample Application: Generating Nonduplicate Integers 179
 - Problem*, 179
 - Output*, 179
 - Solution*, 180
 - Java Implementation*, 180
 - Discussion*, 182
 - The Set Interface and HashSet Implementation*, 183
- 4.7 Exceptions 184
 - The finally Clause*, 186

Deliberately Throwing an Exception in a Program, 187
The throws Clause, 188
Exceptions and Debugging, 190
2's Complement Representation of Integers, 191
Garbage Collection, 192
The Bitwise/Logical Operators, 193
The Conditional Operator, 194

5 CLASSES

204

- 5.1 Class and Member Scope 205
 - Class Scope*, 205
 - Member Scope*, 208
 - Summary of Class and Member Scope*, 211
- 5.2 Constructors, Methods, and Fields 212
 - Constructors*, 213
 - The No-Argument Constructor*, 215
 - Restricting Object Creation through Constructors*, 217
 - Constructors and Unreferenced Objects*, 218
 - Methods*, 219
 - Properties and Get/Set Methods*, 219
 - Methods and Miscellaneous Functionality*, 220
 - Invoking Methods from Constructors*, 221
 - Returning Values from Methods with Return Types*, 221
 - Object Construction through Factory Methods*, 222
 - Fields*, 223
 - Instance Fields and an Object's State*, 223
 - Field Initialization*, 224
 - static Methods and Fields*, 226
 - A Workaround for Accessing static Members*, 227
 - Uses for static Members*, 228
 - Deprecated Features*, 230
- 5.3 Learning Library Classes 233
 - Test Clients*, 234
- 5.4 Sample Application: Basic Input and Output Classes 238
 - Problem*, 238
 - Input/Output*, 238
 - Solution*, 240
 - Java Implementation*, 241
 - Discussion*, 244
 - Stream Input/Output*, 244
 - The BasicInput Class*, 244
 - End of File*, 246
 - Buffered Input and Output*, 246
 - Error Checking*, 248
 - The BasicOutput Class*, 248

Buffer Flushing, 249
Summary of the Basic Input/Output Classes, 250

5.5 Sample Program: A Utility Class for File Copying 251

Problem, 251
Sample Output, 251
Solution, 252
Java Implementation, 252
Discussion, 254
The Abstract Window Toolkit and the Swing Set, 254
The FileDialog Component, 255
The CopyUtil Class, 256
Program Development, 257
Inner Classes, 258

6 INHERITANCE, INTERFACES, AND ABSTRACT CLASSES

270

6.1 Inheritance Basics 272

The Class Object, 273
Some Important Object Methods, 274
Scope and Inheritance, 276
Scope and Inheritance, 278
Changing the Scope of an Inherited Member, 280
Constructors under Inheritance, 280
Name Hiding, 286
Disabling Inheritance with the final Modifier, 287
Casting and Type Safety, 288
The Danger of Down Casts, 289

6.2 Polymorphism 291

Overriding the toString Method, 295
Overriding and Name Hiding, 297
Method Overrides Invoking the Superclass Method, 298
Disabling Overrides with the final Modifier, 298
Overriding versus Overloading, 300

6.3 Sample Application: Polymorphic Input and Output 303

Problem, 303
Sample Input/Output, 303
Solution, 303
Java Implementation, 304
Discussion, 313

6.4 Interfaces 315

Interfaces and Inheritance, 317
Interfaces as Reference Data Types, 318
Nested Interfaces, 320
Application Program Interfaces, 320

- 6.5 Abstract Classes 322
 - Three Ways to Make a Class Abstract*, 323
 - Summary of Concrete Classes, Abstract Classes, and Interfaces*, 324
 - Interface Types, Anonymous Classes, and Unreferenced Objects*, 325

7 GRAPHICS AND EVENT HANDLING

333

- 7.1 Overview of the AWT and the Swing Set 334
 - The Model-View-Controller Architecture*, 335
 - Common Features in the AWT and the Swing Set*, 336
- 7.2 Event-Driven Programming 337
 - The Event-Delegation Model*, 339
 - The Action Interface*, 344
- 7.3 Components and Containers 346
 - The JFrame Window*, 348
 - Terminating an Application by Closing Its Top-Level Window*, 353
 - The Action Interface and AbstractAction Class*, 355
 - Menus, Popup Menus, and Submenus*, 358
 - Keyboard Shortcuts*, 358
 - Popup Menus*, 361
 - Tool Bars*, 364
 - Dialog Windows*, 364
- 7.4 Sample Application: Directory Assistance 369
 - Problem*, 369
 - Sample Output*, 369
 - Solution*, 369
 - Java Implementation*, 369
 - Discussion*, 374
 - Building Tree Structures*, 375
 - Event Handling*, 376
- 7.5 The Model-View-Controller Architecture 377
 - Integration of a Component's View and Controller*, 381
 - Component Look and Feel*, 382
 - Painting and Repainting*, 387
 - Validating and Invalidating*, 389
- 7.6 Sample Application: A Graphical Table Editor 392
 - Problem*, 392
 - Sample Input*, 392
 - Solution*, 392
 - Java Implementation*, 393
 - Discussion*, 395
 - Editing Employee Records*, 396

8 **THREE INTERFACES: CLONEABLE, SERIALIZABLE, AND RUNNABLE** **402**

- 8.1 Cloning Objects 403
 - The Default Implementation of the clone Method*, 404
 - Problems with the Default clone Method*, 405
 - Cloning Arrays*, 408
 - Disabling Cloning*, 409
- 8.2 Serialization 410
 - Serialization Basics*, 411
 - Rules for Serialization*, 411
 - Serialization and Object Graphs*, 412
 - Nonserializable Superclasses with Serializable Subclasses*, 412
 - The Object Input and Output Streams*, 413
 - Serialization, Arrays, and Primitive Types*, 414
 - Primitive Types and Object Streams*, 415
 - Serialization and Strings*, 416
 - Serialization and static and transient Fields*, 416
 - Customizing Serialization*, 418
 - Cautionary Notes on Serialization*, 421
 - The Serial Version Number*, 423
 - The Externalizable Interface*, 424
 - Disabling Serialization*, 424
- 8.3 Sample Application: A Serializable Time Card 425
 - Problem*, 425
 - Sample Output*, 425
 - Solution*, 426
 - Java Implementation*, 426
 - Discussion*, 429
- 8.4 Multithreaded Programs 430
 - From Single-Threaded to Multithreaded Execution*, 430
 - Interleaved Thread Execution*, 432
 - Summary of Thread Execution*, 433
 - Benefits of Multithreading*, 433
 - Two Ways to Make a Program Multithreaded*, 441
 - Multithreading and Program Termination*, 441
 - User and Daemon Threads*, 443
 - Thread Priorities*, 444
 - Thread States*, 446
 - Thread Groups*, 447
 - Thread Synchronization*, 449
 - The join Method*, 451
 - The Need for Thread Synchronization*, 451
 - Critical Sections and Mutual Exclusion*, 452
 - Communication among Synchronized Threads*, 454
 - The wait Method and Notification*, 455
 - The notify and notifyAll Methods*, 457

Deadlock, 458
Summary of Nondeprecated Thread Constructors and Methods, 460

- 8.5 Sample Application: The Dining Philosophers Problem 463
Problem, 463
Sample Output, 464
Solution, 464
Java Implementation, 465
Discussion, 469
Solutions to Critical Section Problems, 470
Deprecated Thread Methods, 472
Threads, Compiler Optimization, and the volatile Modifier, 472

9 NETWORK PROGRAMMING

481

- 9.1 Basic Concepts 482
IP Addresses, 483
Packet Structure, 484
Port Numbers, 484
Sockets, 485
Reliable versus Best-Try Transport, 485
Firewalls and Proxy Servers, 486
- 9.2 Sockets 487
Client Sockets, 487
Server Sockets, 491
Datagram Sockets, 492
Serialization and Sockets, 497
Multicast Sockets, 499
Java Secure Sockets Extension, 500
Testing Distributed Applications on a Standalone Machine, 501
- 9.3 Sample Application: A Multithreaded Cliche Server 502
Problem, 502
Sample Input/Output, 502
Solution, 502
Java Implementation, 503
Discussion, 505
The Client, 506
- 9.4 Applets 506
The Applet and JApplet Classes, 507
Initializing, Starting, and Stopping Applets, 512
The Appletviewer Utility, 515
Communicating Applets, 515
JAR Files, 517

- Applet Security and the Sandbox*, 517
- Java Applications as Host Programs for Applets*, 518
- 9.5 Sample Application: MACE as a Socketed Applet 522
 - Problem*, 522
 - Sample Input/Output*, 523
 - Solution*, 523
 - Java Implementation*, 523
 - Discussion*, 527
- 9.6 Remote Method Invocation 528
 - A Sample RMI Server and Client*, 529
 - The RMI Client*, 529
 - Security Permissions*, 531
 - The RMI Server*, 532
 - RMI Activation*, 533
 - RMI and Jini*, 533
- 9.7 Matrix Algebra Operations 534
 - Problem*, 534
 - Sample Input/Output*, 534
 - Solution*, 536
 - Java Implementation*, 537
 - Discussion*, 538
 - The Time Complexity of Matrix Addition and Multiplication*, 538
- 9.8 Object Request Brokers and CORBA 539
 - A Sample CORBA Application*, 540
 - The IDL File*, 540
 - The Server*, 542
 - The CORBA Naming Service*, 542
 - The Client*, 545
 - Running the Application*, 546
 - The Dynamic Invocation Interface*, 547
 - Summary of Networking Technologies*, 547
 - The jar Utility*, 548
 - Security Issues for Applets as CORBA Clients*, 548

10 SELECTED TOPICS

554

- 10.1 Beans 555
 - Dynamic Editing of Component Properties*, 556
 - Property Change Events*, 561
 - The Bean Box*, 564
 - Enterprise Java Beans*, 567
- 10.2 Security and Cryptography 568
 - Basic Security Constructs*, 569
 - The Security Manager*, 570

	<i>The Access Controller</i> , 572	
	<i>Permissions</i> , 573	
	<i>Security Policy Files</i> , 578	
	<i>Cryptography</i> , 579	
	<i>Message Digests and Digital Signatures</i> , 580	
	<i>Private and Public Key Systems</i> , 582	
	<i>Generating and Verifying a Digital Signature</i> , 583	
	<i>The Java Cryptography Extension</i> , 588	
10.3	Reflection 589	
	<i>Testing for Serializability</i> , 589	
	<i>Obtaining Run-Time Class Information</i> , 590	
	<i>Reflection and Beans</i> , 594	
10.4	Servlets and Database 595	
	<i>Servlet Basics</i> , 596	
	<i>The Action Tag</i> , 596	
	<i>The doGet and doPost Callbacks</i> , 597	
	<i>JDBC Basics</i> , 599	
10.5	Sample Application: Database Webification 602	
	<i>Problem</i> , 602	
	<i>Sample Input/Output</i> , 603	
	<i>Solution</i> , 603	
	<i>Java Implementation</i> , 603	
	<i>Discussion</i> , 609	
	<i>The Products Servlet</i> , 609	
	<i>The SalesServlet</i> , 610	
	<i>The Bean Box Utility</i> , 611	
	<i>The Java Servlet Development Kit</i> , 612	
	<i>Setting Up the Northwind Database for the JDBC-to-ODBC Bridge</i> , 613	
	HINTS AND SOLUTIONS TO ODD-NUMBERED EXERCISES	617
	INDEX	636