Brief Contents

Chapter 1  Introduction to Java Programming  29
Chapter 2  Primitive Data and Definite Loops  91
Chapter 3  Introduction to Parameters and Objects  164
Supplement 3G  Graphics (Optional)  222
Chapter 4  Conditional Execution  262
Chapter 5  Program Logic and Indefinite Loops  339
Chapter 6  File Processing  411
Chapter 7  Arrays  467
Chapter 8  Classes  544
Chapter 9  Inheritance and Interfaces  601
Chapter 10  ArrayLists  676
Chapter 11  Java Collections Framework  729
Chapter 12  Recursion  768
Chapter 13  Searching and Sorting  846
Chapter 14  Stacks and Queues  898
Chapter 15  Implementing a Collection Class  936
Chapter 16  Linked Lists  979
Chapter 17  Binary Trees  1031
Chapter 18  Advanced Data Structures  1085
Appendix A  Java Summary  1121
Appendix B  The Java API Specification and Javadoc Comments  1136
Appendix C  Additional Java Syntax  1142
Contents

Chapter 1 Introduction to Java Programming 29

1.1 Basic Computing Concepts 30
   Why Programming? 30
   Hardware and Software 31
   The Digital Realm 32
   The Process of Programming 34
   Why Java? 35
   The Java Programming Environment 36

1.2 And Now—Java 38
   String Literals (Strings) 42
   System.out.println 43
   Escape Sequences 43
   print versus println 45
   Identifiers and Keywords 46
   A Complex Example: DrawFigures1 48
   Comments and Readability 49

1.3 Program Errors 52
   Syntax Errors 52
   Logic Errors (Bugs) 56

1.4 Procedural Decomposition 56
   Static Methods 59
   Flow of Control 62
   Methods That Call Other Methods 64
   An Example Runtime Error 67

1.5 Case Study: DrawFigures 68
   Structured Version 69
   Final Version without Redundancy 71
   Analysis of Flow of Execution 72

Chapter 2 Primitive Data and Definite Loops 81

2.1 Basic Data Concepts 92
   Primitive Types 92

...
3.4 Case Study: Projectile Trajectory
Unstructured Solution
Structured Solution

Supplement 3G Graphics (Optional)

3G.1 Introduction to Graphics
DrawingPanel
Drawing Lines and Shapes
Colors
Drawing with Loops
Text and Fonts

3G.2 Procedural Decomposition with Graphics
A Larger Example: DrawDiamonds

3G.3 Case Study: Pyramids
Unstructured Partial Solution
Generalizing the Drawing of Pyramids
Complete Structured Solution

Chapter 4 Conditional Execution

4.1 if/else Statements
Relational Operators
Nested if/else Statements
Object Equality
Factoring if/else Statements
Testing Multiple Conditions

4.2 Cumulative Algorithms
Cumulative Sum
Min/Max Loops
Cumulative Sum with if
Roundoff Errors

4.3 Text Processing
The char Type
char versus int
Cumulative Text Algorithms
System.out.printf

4.4 Methods with Conditional Execution
Preconditions and Postconditions
Throwing Exceptions
4.5 Case Study: Body Mass Index 309
One-Person Unstructured Solution 310
Two-Person Unstructured Solution 313
Two-Person Structured Solution 315
Procedural Design Heuristics 319

Chapter 5 Program Logic and Indefinite Loops 339

5.1 The while Loop 340
A Loop to Find the Smallest Divisor 341
Random Numbers 344
Simulations 348
do/while Loop 349

5.2 Fencepost Algorithms 351
Sentinel Loops 353
Fencepost with if 354

5.3 The boolean Type 357
Logical Operators 359
Short-Circuited Evaluation 362
boolean Variables and Flags 366
Boolean Zen 368
Negating Boolean Expressions 371

5.4 User Errors 372
Scanner Lookahead 373
Handling User Errors 375

5.5 Assertions and Program Logic 377
Reasoning about Assertions 379
A Detailed Assertions Example 380

5.6 Case Study: NumberGuess 385
Initial Version without Hinting 385
Randomized Version with Hinting 387
Final Robust Version 391

Chapter 6 File Processing 411

6.1 File-Reading Basics 412
Data, Data Everywhere 412
## Contents

Accessing Inherited Fields 612  
Calling a Superclass’s Constructor 614  
DividendStock Behavior 616  
The Object Class 618  
The equals Method 619  
The instanceof Keyword 622  

### 9.3 Polymorphism 624  
Polymorphism Mechanics 627  
Interpreting Inheritance Code 629  
Interpreting Complex Calls 631  

### 9.4 Inheritance and Design 634  
A Misuse of Inheritance 634  
Is-a Versus Has-a Relationships 637  
Graphics2D 638  

### 9.5 Interfaces 640  
An Interface for Shapes 641  
Implementing an Interface 643  
Benefits of Interfaces 646  

### 9.6 Case Study: Financial Class Hierarchy 648  
Designing the Classes 649  
Redundant Implementation 653  
Abstract Classes 656  

## Chapter 10 ArrayLists 676  

### 10.1 ArrayLists 676  
Basic ArrayList Operations 678  
ArrayList Searching Methods 681  
A Complete ArrayList Program 684  
Adding to and Removing from an ArrayList 686  
Using the For-Each Loop with ArrayLists 690  
Wrapper Classes 691  

### 10.2 The Comparable Interface 694  
Natural Ordering and compareTo 696  
Implementing the Comparable Interface 700  

### 10.3 Case Study: Vocabulary Comparison 706  
Some Efficiency Considerations 706  
Version 1: Compute Vocabulary 709  
Version 2: Compute Overlap 712  
Version 3: Complete Program 717
12.6 Case Study: Prefix Evaluator
Infix, Prefix, and Postfix Notation
Evaluating Prefix Expressions
Complete Program

Chapter 13 Searching and Sorting

13.1 Searching and Sorting in the Java Class Libraries
Binary Search
Sorting
Shuffling
Custom Ordering with Comparators

13.2 Program Complexity
Empirical Analysis
Complexity Classes

13.3 Implementing Searching and Sorting Algorithms
Sequential Search
Binary Search
Recursive Binary Search
Searching Objects
Selection Sort

13.4 Case Study: Implementing Merge Sort
Splitting and Merging Arrays
Recursive Merge Sort
Complete Program

Chapter 14 Stacks and Queues

14.1 Stack/Queue Basics
Stack Concepts
Queue Concepts

14.2 Common Stack/Queue Operations
Transferring Between Stacks and Queues
Sum of a Queue
Sum of a Stack

14.3 Complex Stack/Queue Operations
Removing Values from a Queue
Comparing Two Stacks for Similarity

14.4 Case Study: Expression Evaluator
Splitting into Tokens
The Evaluator
Chapter 15 Implementing a Collection Class  

15.1 Simple `ArrayIntList`  
Adding and Printing  
Thinking about Encapsulation  
Dealing with the Middle of the List  
Another Constructor and a Constant  
Preconditions and Postconditions  

15.2 A More Complete `ArrayIntList`  
Throwing Exceptions  
Convenience Methods  

15.3 Advanced Features  
Resizing When Necessary  
Adding an Iterator  

15.4 `ArrayList<E>`  

Chapter 16 Linked Lists  

16.1 Working with Nodes  
Constructing a List  
List Basics  
Manipulating Nodes  
Traversing a List  

16.2 A Linked List Class  
Simple `LinkedIntList`  
Appending add  
The Middle of the List  

16.3 A Complex List Operation  
Inchworm Approach  

16.4 An `IntList` Interface  

16.5 `LinkedList<E>`  
Linked List Variations  
Linked List Iterators  
Other Code Details  

Chapter 17 Binary Trees  

17.1 Binary Tree Basics  
Node and Tree Classes  

### Contents

<table>
<thead>
<tr>
<th>Chapter 15 Implementing a Collection Class</th>
<th>936</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.1 Simple <code>ArrayIntList</code></td>
<td>937</td>
</tr>
<tr>
<td>Adding and Printing</td>
<td>937</td>
</tr>
<tr>
<td>Thinking about Encapsulation</td>
<td>943</td>
</tr>
<tr>
<td>Dealing with the Middle of the List</td>
<td>944</td>
</tr>
<tr>
<td>Another Constructor and a Constant</td>
<td>949</td>
</tr>
<tr>
<td>Preconditions and Postconditions</td>
<td>950</td>
</tr>
<tr>
<td>15.2 A More Complete <code>ArrayIntList</code></td>
<td>954</td>
</tr>
<tr>
<td>Throwing Exceptions</td>
<td>954</td>
</tr>
<tr>
<td>Convenience Methods</td>
<td>957</td>
</tr>
<tr>
<td>15.3 Advanced Features</td>
<td>960</td>
</tr>
<tr>
<td>Resizing When Necessary</td>
<td>960</td>
</tr>
<tr>
<td>Adding an Iterator</td>
<td>962</td>
</tr>
<tr>
<td>15.4 <code>ArrayList&lt;E&gt;</code></td>
<td>968</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 16 Linked Lists</th>
<th>979</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.1 Working with Nodes</td>
<td>980</td>
</tr>
<tr>
<td>Constructing a List</td>
<td>981</td>
</tr>
<tr>
<td>List Basics</td>
<td>983</td>
</tr>
<tr>
<td>Manipulating Nodes</td>
<td>986</td>
</tr>
<tr>
<td>Traversing a List</td>
<td>989</td>
</tr>
<tr>
<td>16.2 A Linked List Class</td>
<td>993</td>
</tr>
<tr>
<td>Simple <code>LinkedIntList</code></td>
<td>993</td>
</tr>
<tr>
<td>Appending add</td>
<td>995</td>
</tr>
<tr>
<td>The Middle of the List</td>
<td>999</td>
</tr>
<tr>
<td>16.3 A Complex List Operation</td>
<td>1006</td>
</tr>
<tr>
<td>Inchworm Approach</td>
<td>1011</td>
</tr>
<tr>
<td>16.4 An <code>IntList</code> Interface</td>
<td>1012</td>
</tr>
<tr>
<td>16.5 <code>LinkedList&lt;E&gt;</code></td>
<td>1015</td>
</tr>
<tr>
<td>Linked List Variations</td>
<td>1016</td>
</tr>
<tr>
<td>Linked List Iterators</td>
<td>1019</td>
</tr>
<tr>
<td>Other Code Details</td>
<td>1021</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 17 Binary Trees</th>
<th>1031</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.1 Binary Tree Basics</td>
<td>1032</td>
</tr>
<tr>
<td>Node and Tree Classes</td>
<td>1035</td>
</tr>
</tbody>
</table>