Data Structures and the Java Collections Framework

Third Edition

William J. Collins
Lafayette College
CONTENTS

Preface xvii

CHAPTER 0
Introduction to Java 1

Chapter Objectives 1

0.1 Java Fundamentals 1
  0.1.1 Primitive Types 2
  0.1.2 The char Type 2

0.2 Classes 3
  0.2.1 The String Class 4
  0.2.2 Using javadoc Notation for Method Specifications 5
  0.2.3 Equality of References and Equality of Objects 7
  0.2.4 Local Variables 9
  0.2.5 The Scanner Class 12

0.3 Arrays 17

0.4 Arguments and Parameters 19

0.5 Output Formatting 22

Crossword Puzzle 24
Programming Exercises 25

CHAPTER 1
Object-Oriented Concepts 27

Chapter Objectives 27

1.1 Data Abstraction 27

1.2 Abstract Methods and Interfaces 28
  1.2.1 Abstract Data Types and Data Structures 31
  1.2.2 An Interface and a Class that Implements the Interface 31
  1.2.3 Using the FullTimeEmployee Class 35

1.3 Inheritance 37
  1.3.1 The protected Visibility Modifier 39
  1.3.2 Inheritance and Constructors 43
  1.3.3 The Subclass Substitution Rule 43
  1.3.4 Is-a versus Has-a 47

1.4 Information Hiding 48

1.5 Polymorphism 48

1.6 The Unified Modeling Language 49

Summary 52
Crossword Puzzle 54
Concept Exercises 55
Programming Exercises 56
Programming Project 1.1: A CalendarDate Class 58

CHAPTER 2
Additional Features of Programming and Java 59

Chapter Objectives 59

2.1 Static Variables, Constants and Methods 59

2.2 Method Testing 61
  2.2.1 More Details on Unit Testing 64

2.3 Exception Handling 68
  2.3.1 Propagating Exceptions 71
  2.3.2 Unit Testing and Propagated Exceptions 74
  2.3.3 Checked Exceptions 77
  2.3.4 The finally Block 81

2.4 File Output 81

2.5 System Testing 91

2.6 The Java Virtual Machine 92
  2.6.1 Pre-Initialization of Fields 92
  2.6.2 Garbage Collection 92

2.7 Packages 93
2.8 Overriding the Object Class’s equals Method 94
Summary 97
Crossword Puzzle 98
Concept Exercises 99
Programming Exercises 100
Programming Project 2.1: An Integrated Web Browser and Search Engine, Part 1 102

CHAPTER 3
Analysis of Algorithms 105

Chapter Objectives 105
3.1 Estimating the Efficiency of Methods 105
3.1.1 Big-O Notation 106
3.1.2 Getting Big-O Estimates Quickly 110
3.1.3 Big-Omega, Big-Theta and Plain English 116
3.1.4 Growth Rates 117
3.1.5 Trade-Offs 119
3.2 Run-Time Analysis 121
3.2.1 Timing 121
3.2.2 Overview of the Random Class 122
Summary 126
Crossword Puzzle 127
Concept Exercises 128
Programming Exercises 130
Programming Project 3.1: Let’s Make a Deal! 131

CHAPTER 4
The Java Collections Framework 133

Chapter Objectives 133
4.1 Collections 133
4.1.1 Collection Classes 134
4.1.2 Storage Structures for Collection Classes 136
4.2 Some Details of the Java Collections Framework 136
4.2.1 Abstract Classes 137
4.2.2 Parameterized Types 140
4.2.3 The Collection Interface 141
4.2.4 The List Interface 147
Summary 150
Crossword Puzzle 151
Concept Exercises 152
Programming Exercises 152
Programming Project 4.1: Let’s Make a Developer’s Hat and a User’s Hat 153

CHAPTER 5
Recursion 155

Chapter Objectives 155
5.1 Introduction 155
5.2 Factorials 156
5.2.1 Execution Frames 159
5.3 Decimal to Binary 162
5.4 Towers of Hanoi 167
5.4.1 Analysis of the move Method 177
5.5 Searching an Array 179
5.6 Backtracking 191
5.6.1 An A-maze-ing Application 195
5.7 Indirect Recursion 208
5.8 The Cost of Recursion 209
Summary 210
Crossword Puzzle 211
Concept Exercises 212
Programming Exercises 214
Programming Project 5.1: Iterative Version of the Towers of Hanoi 219
Programming Project 5.2: Eight Queens 221
Programming Project 5.3: A Knight’s Tour 222
Programming Project 5.4: Sudoku 225
Programming Project 5.5: Numbrix 227
CHAPTER 6
Array-Based Lists 233

Chapter Objectives 233
6.1 The List Interface 234
6.2 The ArrayList Class 234
   6.2.1 Method Specifications for the ArrayList Class 235
   6.2.2 A Simple Program with an ArrayList Object 244
   6.2.3 The ArrayList Class's Heading and Fields 246
   6.2.4 Definition of the One-Parameter add Method 248
6.3 Application: High-Precision Arithmetic 251
   6.3.1 Method Specifications and Testing of the VeryLongInt Class 252
   6.3.2 Fields in the VeryLongInt Class 253
   6.3.3 Method Definitions of the VeryLongInt Class 254
Summary 257
Crossword Puzzle 258
Concept Exercises 259
Programming Exercises 259
Programming Project 6.1: Expanding the VeryLongInt Class 263
Programming Project 6.2: An Integrated Web Browser and Search Engine, Part 2 264

CHAPTER 7
Linked Lists 267

Chapter Objectives 267
7.1 What is a Linked List? 267
7.2 The SinglyLinkedList Class—A Singly-Linked, Toy Class! 268
   7.2.1 Fields and Method Definitions in the SinglyLinkedList Class 273
   7.2.2 Iterating through a SinglyLinkedList Object 276
7.3 Doubly-Linked Lists 281
   7.3.1 A User's View of the LinkedList Class 282
   7.3.2 The LinkedList Class versus the ArrayList Class 282
   7.3.3 LinkedList Iterators 285
   7.3.4 A Simple Program that uses a LinkedList Object 291
   7.3.5 Fields and Heading of the LinkedList Class 294
   7.3.6 Creating and Maintaining a LinkedList Object 296
   7.3.7 Definition of the Two-Parameter add Method 298
7.4 Application: A Line Editor 300
   7.4.1 Design and Testing of the Editor Class 304
   7.4.2 Method Definitions for the Editor Class 308
   7.4.3 Analysis of the Editor Class Methods 312
   7.4.4 Design of the EditorUser Class 312
   7.4.5 Implementation of the EditorUser Class 313
Summary 315
Crossword Puzzle 316
Concept Exercises 317
Programming Exercises 318
Programming Project 7.1: Expanding the SinglyLinkedList Class 320
Programming Project 7.2: Implementing the remove() Method in SinglyLinkedListIterator 322
Programming Project 7.3: Making a Circular Singly Linked List Class 322
Programming Project 7.4: Alternative Implementation of the LinkedList Class 323
Programming Project 7.5: Expanding the Line Editor 323
Programming Project 7.6: An Integrated Web Browser and Search Engine, Part 3 328
CHAPTER 8
Stacks and Queues 329

Chapter Objectives 329
8.1 Stacks 329
  8.1.1 The Stack Class 329
  8.1.2 A Fatal Flaw? 333
  8.1.3 Stack Application 1: How Compilers Implement Recursion 334
  8.1.4 Stack Application 2: Converting from Infix to Postfix 338
  8.1.5 Prefix Notation 343
8.2 Queues 347
  8.2.1 The Queue Interface 348
  8.2.2 Implementations of the Queue Interface 349
  8.2.3 Computer Simulation 350
  8.2.4 Queue Application: A Simulated Car Wash 351

Summary 365
Crossword Puzzle 366
Concept Exercises 367
Programming Exercises 368
Programming Project 8.1: Making the Speedo’s Car Wash Simulation More Realistic 369
Programming Project 8.2: Design, Test, and Implement a Program to Evaluate a Condition 371
Programming Project 8.3: Maze-Searching, Revisited 374
Programming Project 8.4: Fixing the Stack Class 375

CHAPTER 9
Binary Trees 377

Chapter Objectives 377
9.1 Definition of Binary Tree 377
9.2 Properties of Binary Trees 378
9.3 The Binary Tree Theorem 383
9.4 External Path Length 385
9.5 Traversals of a Binary Tree 386

Summary 393
Crossword Puzzle 394
Concept Exercises 395

CHAPTER 10
Binary Search Trees 401

Chapter Objectives 401
10.1 Binary Search Trees 402
  10.1.1 The BinarySearchTree Implementation of the Set Interface 403
  10.1.2 Implementation of the BinarySearchTree Class 411
10.2 Balanced Binary Search Trees 430
  10.2.1 AVL Trees 435
  10.2.2 The Height of an AVL Tree 436
  10.2.3 The AVLTree Class 438
  10.2.4 Runtime Estimates 441

Summary 442
Crossword Puzzle 443
Concept Exercises 444
Programming Exercises 448
Programming Project 10.1: An Alternate Implementation of the Binary-Search-Tree Data Type 449
Programming Project 10.2: Printing a BinarySearchTree Object 451
Programming Project 10.3: The fixAfterInsertion Method 451
Programming Project 10.4: The fixAfterDeletion Method 455

CHAPTER 11
Sorting 457

Chapter Objectives 457
11.1 Introduction 457
Contents xiii

11.2 Simple Sorts 458
   11.2.1 Insertion Sort 459
   11.2.2 Selection Sort 461
   11.2.3 Bubble Sort 463

11.3 The Comparator Interface 465

11.4 How Fast Can We Sort? 468
   11.4.1 Merge Sort 470
   11.4.2 The Divide-and-Conquer Design Pattern 476
   11.4.3 Quick Sort 477

11.5 Radix Sort 489
Summary 493
Crossword Puzzle 494
Concept Exercises 495
Programming Exercises 497
Programming Project 11.1: Sorting a File into Ascending Order 497

Chapter 12
Tree Maps and Tree Sets 501

12.1 Red-Black Trees 501
   12.1.1 The Height of a Red Black Tree 503

12.2 The Map Interface 504

12.3 The TreeMap Implementation of the SortedMap Interface 509
   12.3.1 The TreeMap Class’s Fields and Embedded Entry Class 512
   12.3.2 Method Definitions in the TreeMap Class 513

12.4 Application of the TreeMap Class: a Simple Thesaurus 517
   12.4.1 Design, Testing, and Implementation of the Thesaurus Class 518
   12.4.2 Design and Testing of the ThesaurusUser Class 521
   12.4.3 Implementation of the ThesaurusUser Class 523

12.5 The TreeSet Class 525
   12.5.1 Implementation of the TreeSet Class 528
   12.5.2 Application: A Simple Spell Checker 530

Summary 536
Crossword Puzzle 537
Concept Exercises 538
Programming Exercises 539
Programming Project 12.1: Spell Check, Revisited 540
Programming Project 12.2: Word Frequencies 542
Programming Project 12.3: Building a Concordance 543
Programming Project 12.4: Approval Voting 545
Programming Project 12.5: An Integrated Web Browser and Search Engine, Part 4 548

Chapter 13
Priority Queues 551

13.1 Introduction 551

13.2 The PriorityQueue Class 552

13.3 Implementation Details of the PriorityQueue Class 553
   13.3.1 Fields and Method Definitions in the PriorityQueue Class 557

13.4 The heapSort Method 567
   13.4.1 Analysis of heapSort 572

13.5 Application: Huffman Codes 573
   13.5.1 Huffman Trees 575
   13.5.2 Greedy Algorithm Design Pattern 578
   13.5.3 The Huffman Encoding Project 578

Summary 590
Crossword Puzzle 591
Concept Exercises 592
Programming Project 13.2: An Integrated Web Browser and Search Engine, Part 5 597

CHAPTER 14
Hashing 599

Chapter Objectives 599
14.1 A Framework to Analyze Searching 599
14.2 Review of Searching 600
14.2.1 Sequential Search 600
14.2.2 Binary Search 601
14.2.3 Red-Black-Tree Search 602
14.3 The HashMap Implementation of the Map Interface 603
14.3.1 Hashing 604
14.3.2 The Uniform Hashing Assumption 609
14.3.3 Chaining 609
14.3.4 Implementation of the HashMap Class 612
14.3.5 Analysis of the containsKey Method 614
14.3.6 The HashIterator Class 615
14.3.7 Creating a Symbol Table by Hashing 617
14.4 The HashSet Class 625
14.5 Open-Address Hashing (optional) 626
14.5.1 The remove Method 627
14.5.2 Primary Clustering 631
14.5.3 Double Hashing 632

Summary 635
Crossword Puzzle 636
Concept Exercises 637
Programming Exercises 639
Programming Project 14.1: The Double Hashing Implementation of the HashMap Class 640
Programming Project 14.2: An Integrated Web Browser and Search Engine, Part 6 640

CHAPTER 15
Graphs, Trees, and Networks 643

Chapter Objectives 643
15.1 Undirected Graphs 643
15.2 Directed Graphs 647
15.3 Trees 648
15.4 Networks 649
15.5 Graph Algorithms 650
15.5.1 Iterators 650
15.5.2 Connectedness 658
15.5.3 Generating a Minimum Spanning Tree 659
15.5.4 Finding the Shortest Path through a Network 663
15.5.5 Finding the Longest Path through a Network? 667
15.6 A Network Class 669
15.6.1 Method Specifications and Testing of the Network Class 671
15.6.2 Fields in the Network Class 680
15.6.3 Method Definitions in the Network Class 681
15.7 Backtracking Through A Network 686

Summary 689
Crossword Puzzle 690
Concept Exercises 691
Programming Exercises 693
Programming Project 15.1: The Traveling Salesperson Problem 694
Programming Project 15.2: Backtracking through a Network 695
Programming Project 15.3: Determining Critical Activities in a Project Network 697
Programming Project 15.4: An Integrated Web Browser and Search Engine, Part 7 698
# Appendix 1
## Additional Features of the JAVA Collections Framework 701

A1.1 Introduction 701  
A1.2 Serialization 701  
A1.3 Fail-Fast Iterators 703

# Appendix 2
## Mathematical Background 705

A2.1 Introduction 705  
A2.2 Functions and Sequences 705  
A2.3 Sums and Products 706  
A2.4 Logarithms 707  
A2.5 Mathematical Induction 708  
A2.6 Induction and Recursion 719  
Concept Exercises 719

# Appendix 3
## Choosing a Data Structure 721

A3.1 Introduction 721  
A3.2 Time-Based Ordering 721  
A3.3 Index-Based Ordering 721  
A3.4 Comparison-Based Ordering 722  
A3.5 Hash-Based Ordering 723  
A3.6 Space Considerations 723  
A3.7 The Best Data Structure? 724

References 725  
Index 727