

1

Getting Organized 1

- 1.1 Software Engineering 2
 - Software Life Cycles 3
 - Agile Methods 5
 - Goals of Quality Software 6
- 1.2 Object Orientation 8
 - The Unified Method 8
- 1.3 Classes, Objects, and Applications 9
 - Classes 9
 - Objects 16
 - Applications 17
- 1.4 Organizing Classes 20
 - Inheritance 20
 - Packages 25
- 1.5 Data Structures 28
 - Implementation-Dependent Structures 29
 - Implementation-Independent Structures 30
 - What Is a Data Structure? 33
- 1.6 Basic Structuring Mechanisms 33
 - References 33
 - Arrays 38

1.7	Comparing Algorithms: Big-O Analysis	41
	Big-O Notation	43
	Common Orders of Magnitude	45
	Example 1: Sum of Consecutive Integers	46
	Example 2: Finding a Number in a Phone Book	48
	Summary	50
	Exercises	51

2

Abstract Data Types 63

2.1	Abstraction	64
	Information Hiding	64
	Data Abstraction	65
	Data Levels	66
	Preconditions and Postconditions	67
	Java Interfaces	68
2.2	The StringLog ADT Specification	72
	Constructors	72
	Transformers	73
	Observers	73
	The StringLogInterface	73
	Using the StringLogInterface	75
2.3	Array-Based StringLog ADT Implementation	77
	Instance Variables	77
	Constructors	79
	Transformers	80
	Observers	82
2.4	Software Testing	91
	Identifying Test Cases	92
	Test Plans	94
	Testing ADT Implementations	94
2.5	Introduction to Linked Lists	102
	Array Versus Linked Lists	102
	The LLStringNode Class	103
	Operations on Linked Lists	107

2.6	Linked List StringLog ADT Implementation	113
	Instance Variables	115
	Constructors	115
	Transformers	116
	Observers	119
2.7	Software Design: Identification of Classes	124
	Brainstorm	124
	Filter	125
	Scenario Analysis	125
	Nouns and Verbs	125
	Cohesive Designs	126
	Summation of Our Approach	126
	Design Choices	128
2.8	Case Study: A Trivia Game	128
	The Source of the Trivia Game	129
	Identifying Support Classes	131
	Implementing the Support Classes	133
	The Trivia Game Application	140
	Case Study Summation	145
	Summary	146
	Exercises	146

3

The Stack ADT 159

3.1	Stacks	160
	Operations on Stacks	161
	Using Stacks	161
3.2	Collection Elements	163
	Generally Usable Collections	164
3.3	Exceptional Situations	167
	Handling Exceptional Situations	167
	Exceptions and ADTs: An Example	168
	Error Situations and ADTs	173
3.4	Formal Specification	175
	Exceptional Situations	176
	The Interfaces	179
	Example Use	182

3.5	Array-Based Implementations	185
	The ArrayStack Class	185
	Definitions of Stack Operations	188
	Test Plan	190
3.6	Application: Well-Formed Expressions	194
	The Balanced Class	194
	The Application	200
3.7	Link-Based Implementation	202
	The LLObject Node Class	203
	The LinkedStack Class	204
	The push Operation	205
	The pop Operation	209
	The Other Stack Operations	211
	Comparing Stack Implementations	213
3.8	Case Study: Postfix Expression Evaluator	214
	Discussion	214
	Evaluating Postfix Expressions	215
	Postfix Expression Evaluation Algorithm	216
	Specification: Program Postfix Evaluation	219
	Brainstorming and Filtering	220
	The PostFixEvaluator Class	222
	The PFixConsole Class	224
	Testing the Postfix Evaluator	226
	Summary	229
	Exercises	229

4

Recursion 243

4.1	Recursive Definitions, Algorithms, and Programs	244
	Recursive Definitions	244
	Recursive Algorithms	245
	Recursive Programs	248
4.2	The Three Questions	251
	Verifying Recursive Algorithms	251
	Writing Recursive Methods	253
	Debugging Recursive Methods	253
4.3	Towers of Hanoi	254
	The Algorithm	254

	The Method	256
	The Program	257
4.4	Counting Blobs	260
	Generating Blobs	261
	The Counting Algorithm	262
	The Marking Algorithm	263
	The Grid Class	264
	The Program	267
4.5	Recursive Linked-List Processing	269
	Reverse Printing	269
4.6	Removing Recursion	273
	How Recursion Works	274
	Iteration	278
	Stacking	279
4.7	Deciding Whether to Use a Recursive Solution	281
	Recursion Overhead	282
	Inefficient Algorithms	282
	Clarity	285
	Summary	285
	Exercises	286

5

The Queue ADT 297

5.1	Queues	298
	Operations on the Queues	299
	Using Queues	299
5.2	Formal Specification	300
	Example Use	303
5.3	Array-Based Implementations	305
	The ArrayBndQueue Class	305
	The ArrayUnbndQueue Class	312
5.4	Application: Palindromes	315
	The Palindrome Class	316
	The Application	318
5.5	Application: The Card Game of War	321
	The RankCardDeck Class	321
	The WarGame Class	323
	The WarGameApp Class	327

5.6	Link-Based Implementations	331
	The Enqueue Operation	332
	The Dequeue Operation	334
	The Queue Implementation	336
	A Circular Linked Queue Design	337
	Comparing Queue Implementations	338
5.7	Concurrency, Interference, and Synchronization	340
	The counter Class	340
	Java Threads	342
	Interference	345
	Synchronization	346
	A Synchronized Queue	348
5.8	Case Study: Average Waiting Time	353
	Problem Discussion	354
	Program Design	356
	Program Details	360
	Testing Considerations	369
	Summary	370
	Exercises	372

6 The List ADT 383

6.1	Comparing Objects Revisited	384
	The equals Method	384
	The Comparable Interface	386
6.2	Lists	388
	Varieties of Lists	389
	Assumptions for Our Lists	389
6.3	Formal Specification	390
	The ListInterface	390
	The Indexed List Interface	393
	Example Use	394
6.4	Array-Based Implementations	397
	The ArrayUnsortedList Class	398
	The ArraySortedList Class	403
	The ArrayIndexedList Class	411
6.5	Applications: Poker, Golf, and Music	414
	Poker	414

	Golf	418
	Music	421
6.6	The Binary Search Algorithm	425
	Improving Linear Search in a Sorted List	426
	Binary Search Algorithm	426
	Recursive Binary Search	431
	Efficiency Analysis	432
6.7	Reference-Based Implementations	433
	The RefUnsortedList Class	434
	The RefSortedList Class	440
6.8	Storing Objects and Structures in Files	445
	Saving Object Data in Text Files	445
	Serialization of Objects	447
	Serializing Structures	450
	Application: Song Lists	451
	Summary	458
	Exercises	459

7

More Lists 473

7.1	Circular Linked Lists	474
	An Unsorted Circular List	475
	The CRefUnsortedList Class	476
	Circular Versus Linear Linked Lists	482
7.2	Doubly Linked Lists	483
	The Add and Remove Operations	484
7.3	Linked Lists with Headers and Trailers	487
7.4	A Linked List as an Array of Nodes	488
	Why Use an Array?	488
	How Is an Array Used?	489
7.5	A Specialized List ADT	497
	The Specification	498
	The Implementation	499
7.6	Case Study: Large Integers	504
	The LargeInt Class	508
	Addition and Subtraction	510

Test Plan	519
The LargeIntApp Program	519
Summary	523
Exercises	523

8

Binary Search Trees 531

8.1	Trees	532
	Binary Trees	534
	Binary Search Trees	536
	Binary Tree Traversals	538
8.2	The Logical Level	540
	Tree Elements	540
	The Binary Search Tree Specification	541
8.3	The Application Level	543
8.4	The Implementation Level: Basics	545
8.5	Iterative Versus Recursive Method Implementations	548
	Recursive Approach to the size Method	549
	Iterative Approach to the size Method	553
	Recursion or Iteration?	555
8.6	The Implementation Level: Remaining Operations	555
	The contains and get Operations	555
	The add Operation	559
	The remove Operation	564
	Iteration	571
	Testing Binary Search Tree Operations	575
8.7	Comparing Binary Search Tree and Linear Lists	577
	Big-O Comparisons	577
8.8	Balancing a Binary Search Tree	578
8.9	A Nonlinked Representation of Binary Trees	584
8.10	Case Study: Word Frequency Generator	588
	Problem	588
	Discussion	588
	Brainstorming	588
	Filtering	589
	The User Interface	589
	Error Handling	590
	Scenario Analysis	590

The WordFreq Class	592
The Word Frequency Generator Program	594
Testing	596
Summary	598
Exercises	598

9 Priority Queues, Heaps, and Graphs 613

9.1	Priority Queues	614
	Logical Level	614
	Application Level	616
	Implementation Level	616
9.2	Heaps	617
	Heap Implementation	621
	The enqueue Method	623
	The dequeue Method	626
	A Sample Use	630
	Heaps Versus Other Representations of Priority Queues	632
9.3	Introduction to Graphs	633
9.4	Formal Specification of a Graph ADT	637
9.5	Implementations of Graphs	640
	Array-Based Implementation	640
	Linked Implementation	645
9.6	Graph Applications	646
	Depth-First Searching	647
	Breadth-First Searching	651
	The Single-Source Shortest-Paths Problem	654
	Summary	661
	Exercises	662

10 Sorting and Searching Algorithms 671

10.1	Sorting	672
	A Test Harness	673
10.2	Simple Sorts	676
	Straight Selection Sort	676
	Bubble Sort	682
	Insertion Sort	686

10.3	$O(N \log_2 N)$ Sorts	690
	Merge Sort	690
	Quick Sort	698
	Heap Sort	704
10.4	More Sorting Considerations	710
	Testing	710
	Efficiency	710
	Objects and References	712
	Using the Comparable Interface	712
	Using the Comparator Interface	713
	Stability	719
10.5	Searching	720
	Linear Searching	720
	High Probability Ordering	721
	Sorted Lists	722
10.6	Hashing	722
	Collisions	726
	Choosing a Good Hash Function	734
	Complexity	737
	Summary	738
	Exercises	739
	Appendix A Java Reserved Words	749
	Appendix B Operator Precedence	750
	Appendix C Primitive Data Types	751
	Appendix D ASCII Subset of Unicode	752
	Appendix E Application of Programmer Interfaces for the Java Classes and Interfaces Used in This Book	753
	Index	769