

Java™

Structures

Data Structures

in Java™ for

the Principled

Programmer

Duane A. Bailey

Contents

Preface	xi
0 Introduction	1
0.1 Read Me	1
0.2 He Can't Say That. Can He?	2
1 The Object-Oriented Method	5
1.1 Data Abstraction and Encapsulation	5
1.2 The Object Model	7
1.3 Object-Oriented Terminology	8
1.4 Sketching an Example: A Word List	10
1.5 A Special Purpose Class: A Bank Account	12
1.6 A General Purpose Class: An Association	15
1.7 Interfaces	18
1.8 Who Is the User?	19
1.9 Conclusions	20
2 Comments, Conditions, and Assertions	25
2.1 Pre- and Postconditions	26
2.2 Assertions	26
2.3 Craftsmanship	28
2.4 Conclusions	28
3 Vectors	31
3.1 Application: The Word List Revisited	33
3.2 Application: Word Frequency	34
3.3 The Interface	36
3.4 The Implementation	38
3.5 Extensibility: A Feature	41
3.6 Application: The Matrix Class	43
3.7 Conclusions	47
4 Design Fundamentals	49
4.1 Asymptotic Analysis Tools	49
4.1.1 Time and Space Complexity	50
4.1.2 Examples	53
4.1.3 The Trading of Time and Space	57
4.2 Self-Reference	58
4.2.1 Recursion	58
4.2.2 Mathematical Induction	65

4.3	Properties of Design	70
4.3.1	Symmetry	70
4.3.2	Friction	72
4.4	Conclusions	72
5	Sorting	
5.1	Approaching the Problem	77
✓ 5.2	Selection Sort	77
✓ 5.3	Insertion Sort	80
✓ 5.4	Mergesort	82
✓ 5.5	Quicksort	85
5.6	Sorting Objects	89
5.7	Vector-Based Sorting	92
5.8	Conclusions	95
6	Lists	
6.1	Example: A Unique Program	99
✓ 6.2	Example: Free-Lists	101
✓ 6.3	Implementation: Singly-Linked Lists	102
✓ 6.4	Implementation: Doubly-Linked Lists	105
✓ 6.5	Implementation: Circularly-Linked Lists	116
6.6	Conclusions	121
7	Linear Structures	
7.1	Stacks	127
7.1.1	Example: Simulating Recursion	127
✓ 7.1.2	Vector-Based Stacks	128
✓ 7.1.3	List-Based Stacks	132
7.1.4	Comparisons	134
7.2	Queues	136
7.2.1	Example: Solving a Coin Puzzle	136
✓ 7.2.2	List-Based Queues	137
✓ 7.2.3	Vector-Based Queues	140
7.2.4	Array-Based Queues	142
7.3	Example: Solving Mazes	145
7.4	Conclusions	149
8	Iterators	
8.1	Java's Enumeration Interface	152
8.2	The Iterator Interface	155
8.3	Example: Vector Iterators	157
8.4	Example: List Iterators	158
8.5	Example: Filtering Iterators	160
8.6	Conclusions	162

9	Ordered Structures	167
9.1	Comparable Objects	167
9.1.1	Example: Comparable Integers	168
9.1.2	Example: Comparable Associations	170
9.2	Keeping Structures Ordered	172
9.2.1	The OrderedStructure Interface	173
9.2.2	The Ordered Vector	173
9.2.3	Example: Sorting	179
9.2.4	The Ordered List	179
9.2.5	Example: The Modified Parking Lot	183
9.3	Conclusions	184
10	Trees	187
10.1	Terminology	187
10.2	The Interface	190
10.3	Motivating Example: Expression Trees	192
10.4	Implementation	194
10.4.1	The BinaryTreeNode Implementation	194
10.4.2	Implementation of the BinaryTree Wrapper	197
10.5	Traversals	201
10.5.1	Preorder Traversal	202
10.5.2	Inorder Traversal	204
10.5.3	Postorder Traversal	206
10.5.4	Levelorder Traversal	207
10.5.5	Recursion in Iterators	209
10.6	Property-Based Methods	210
10.7	Example: Huffman Compression	214
10.8	Conclusions	219
11	Priority Queues	223
11.1	The Interface	223
11.2	Example: Improving the Huffman Code	224
11.3	Priority Vectors	225
11.4	A Heap Implementation	227
11.4.1	Vector-Based Heaps	228
11.4.2	Example: Heapsort	236
11.4.3	Skew Heaps	237
11.5	Example: Circuit Simulation	241
11.6	Conclusions	244
12	Search Trees	249
12.1	Binary Search Trees	249
✓ 12.2	Example: Tree Sort	251
✓ 12.3	Implementation	251
12.4	Splay Trees	257
12.5	Splay Tree Implementation	260

12.6 Conclusions	284
13 Dictionaries	
13.1 The Interface	287
13.2 Unit Cost Dictionaries: Hash Tables	287
13.2.1 Open Addressing	288
13.2.2 External Chaining	289
13.2.3 Generation of Hash Codes	277
13.2.4 Analysis	279
13.3 Ordered Dictionaries and Tables	285
13.4 Example: Document Indexing	285
13.5 Conclusions	287
	291
14 Graphs	
14.1 Terminology	293
14.2 The Graph Interface	293
14.3 Implementations	294
14.3.1 Abstract Classes	298
14.3.2 Adjacency Matrices	298
14.3.3 Adjacency Lists	300
14.4 Examples: Common Graph Algorithms	306
14.4.1 Reachability	312
14.4.2 Topological Sorting	312
14.4.3 Transitive Closure	315
14.4.4 All Pairs Minimum Distance	317
14.4.5 Greedy Algorithms	318
14.5 Conclusions	319
	324
A A Sip of Java	
A.1 A First Program	329
A.2 Declarations	329
A.2.1 Primitive Types	331
A.2.2 Reference Types	331
A.3 Important Classes	333
A.3.1 The ReadStream Class	334
A.3.2 PrintStreams	334
A.3.3 Strings	335
A.4 Control Constructs	335
A.4.1 Conditional Statements	336
A.4.2 Loops	336
A.5 Methods	337
A.6 Inheritance and Subtyping	339
A.6.1 Inheritance	340
A.6.2 Subtyping	340
A.6.3 Interfaces and Abstract Classes	341
	342

B Use of the Keyword Protected	345
C Principles	349
D Structure Package Hierarchy	351
E Selected Answers	355
Index	363