
Contents

1	Preliminary Concepts	1
1.1	Introduction to Data Structures	1
1.2	Prior Concepts and Skills Needed	1
1.3	Elementary Algorithm Analysis	2
1.4	Abstraction and Interfaces	8
1.5	Java API	12
1.6	Complete Program: Implementing Interface	13
1.7	Summary	14
1.8	Exercises (Items Marked with an * Have Solutions in Appendix B)	15
2	Stacks Using Arrays	17
2.1	Introduction	17
2.2	Analysis and Design	17
2.3	Stack Class: Data Members and Methods	21
2.4	Reversing Integers	29
2.5	Generic Types	32
2.6	Prefix and Postfix Expressions	37
2.7	Complete Program: Checking for Palindromes in Strings	41
2.8	Summary	45
2.9	Exercises (Items Marked with an * Have Solutions in Appendix B)	46
3	Queues Using Arrays	49
3.1	Introduction	49
3.2	Analysis and Design	52
3.3	Queue Class: Data Members and Methods	58
3.4	Program Using the <code>QueueArray</code> Class	62
3.5	Complete Program: Simulating a Scheduling Algorithm	67
3.6	Summary	73
3.7	Exercises (Items Marked with an * Have Solutions in Appendix B)	73

4	Lists Using Arrays	77
4.1	Introduction	77
4.2	Analysis and Design	78
4.3	ListArray Class: Data Members and Methods	82
4.3.1	The Insert Method	83
4.3.2	The Delete Method	86
4.3.3	The Search Method	89
4.3.4	Complete ListArray Class	92
4.4	Simple Test Program	92
4.5	Complete Program: Checking Opcodes in Assembly Language	97
4.6	Summary	105
4.7	Exercises (Items Marked with an * Have Solutions in Appendix B)	106
5	Lists Using Objects and References	109
5.1	Introduction	109
5.2	References	110
5.3	Objects	111
5.4	The Node Class	114
5.5	Creating Links	116
5.6	Output	121
5.7	Test Program	124
5.8	Output Using Recursion	126
5.9	Complete Program: Outputting Items in the List Using Recursion	128
5.10	Summary	130
5.11	Exercises (Items Marked with an * Have Solutions in Appendix B)	131
6	Ordered Linked Lists	135
6.1	Creating the Insert Method	135
6.1.1	Inserting in the Middle of the List	135
6.1.2	Inserting at the End of the List	142
6.1.3	Inserting at the Beginning of a List	146
6.1.4	Inserting into an Empty List	148
6.1.5	The Insert Method	150
6.2	Creating the Delete Method	151
6.3	The LinkedList Class	154
6.4	Test Program	154
6.5	Doubly Linked Lists	157
6.6	The Node Class as an Inner Class	160
6.7	Complete Programs: List of User Defined Objects	163
6.7.1	Generic LinkedList Class with External NodeGeneric Class	163

6.7.2	Generic LinkedList Class with Internal NodeGeneric Class	167
6.8	Summary	170
6.9	Exercises (Items Marked with an * Have Solutions in Appendix B).....	171
7	Stacks and Queues Using References.....	173
7.1	Stack.....	173
7.2	Stack Test Program: Reversing Integers	177
7.3	Queues	178
7.4	Queue Test Program.....	183
7.5	Comparison of Arrays Verses References.....	186
7.6	Complete Program: Undo Button	186
7.6.1	Graphical User Interface	187
7.6.2	User Defined Frame.....	187
7.6.3	Placing Buttons	189
7.6.4	Handling Button Events.....	192
7.6.5	Generic Linked Stack	193
7.6.6	The UndoButton Class.....	194
7.7	Summary	196
7.8	Exercises (Items Marked with an * Have Solutions in Appendix B).....	197
8	Binary Trees	199
8.1	Introduction	199
8.2	Binary Expression Trees.....	201
8.2.1	General Concepts	201
8.2.2	Output in Prefix Form.....	203
8.2.3	Output in Infix Form.....	211
8.3	Creating a Binary Expression Tree from a Prefix Expression	216
8.4	Generic if Statement for Recursive Algorithms.....	220
8.5	Binary Search Trees.....	221
8.5.1	General Concepts	221
8.5.2	Inorder Traversal.....	222
8.5.3	Creating a Binary Search Tree.....	226
8.5.4	Finding the Minimum and Maximum	230
8.5.5	Removing an Item from a BST.....	231
8.6	The Complete BinarySearchTree Class	240
8.7	Test Program	242
8.8	Complete Program: Implementing a Binary Expression Tree ...	244
8.9	Summary	247
8.10	Exercises (Items Marked with an * Have Solutions in Appendix B).....	247

9	Sorting	249
9.1	Introduction	249
9.2	The Insertion Sort	249
9.3	The Quick Sort	255
9.3.1	General Concepts	255
9.3.2	Potential Problems	256
9.3.3	The Sort3 and SwapElements Methods	257
9.3.4	The QuickSort Method	258
9.4	The Radix Sort	273
9.4.1	General Concepts	273
9.4.2	More Pockets	275
9.4.3	Fewer Pockets	275
9.4.4	Memory	277
9.4.5	Radix Sort Program	280
9.5	Complete Program: Sorting String Items	286
9.6	Summary	290
9.7	Exercises (Items Marked with an * Have Solutions in Appendix B)	290
10	Heaps	293
10.1	General Concepts	293
10.2	Creating Heaps	295
10.3	The insertItem and moveUp Methods	300
10.3.1	The insertItem Method	300
10.3.2	The moveUp Method	301
10.4	Priority Queues	306
10.5	The removeMin and moveDown Methods	309
10.5.1	The removeMin Method	310
10.5.2	The moveDown Method	311
10.5.3	Walk-Through	314
10.6	Test Program	316
10.7	The Heap Sort	319
10.7.1	The Simplified Heap Sort	319
10.7.2	The Modified Heap Sort	320
10.7.3	The makeHeap Method	321
10.7.4	The heapSort Method	324
10.8	Test Program and Output	328
10.9	Complete Program: Priority Queues for Printers	329
10.10	Summary	335
10.11	Exercises (Items Marked with an * Have Solutions in Appendix B)	335

11 Hashing	339
11.1 General Concepts	339
11.2 Mid-square Method	341
11.3 Hash Function for Strings.	342
11.4 Collision Resolution	343
11.5 Implementation of a Hash Table.	344
11.6 Implementation of Hash Tables Using the Java API.	347
11.6.1 The HashMap Class	348
11.6.2 The HashSet Class.	352
11.6.3 The hashCode and equals Methods	355
11.7 Complete Program: Hash Table Using the HashMap Class	357
11.8 Summary	360
11.9 Exercises (Items Marked with an * Have Solutions in Appendix B).	361
Appendix A: Glossary.	363
Appendix B: Answers to Selected Exercises	365
References and Useful Websites.	373
Index	375

Guide to Data Structures

A Concise Introduction Using Java

Streib, J.T.; Soma, T.

2017, XIII, 376 p. 43 illus., 9 illus. in color., Softcover

ISBN: 978-3-319-70083-0