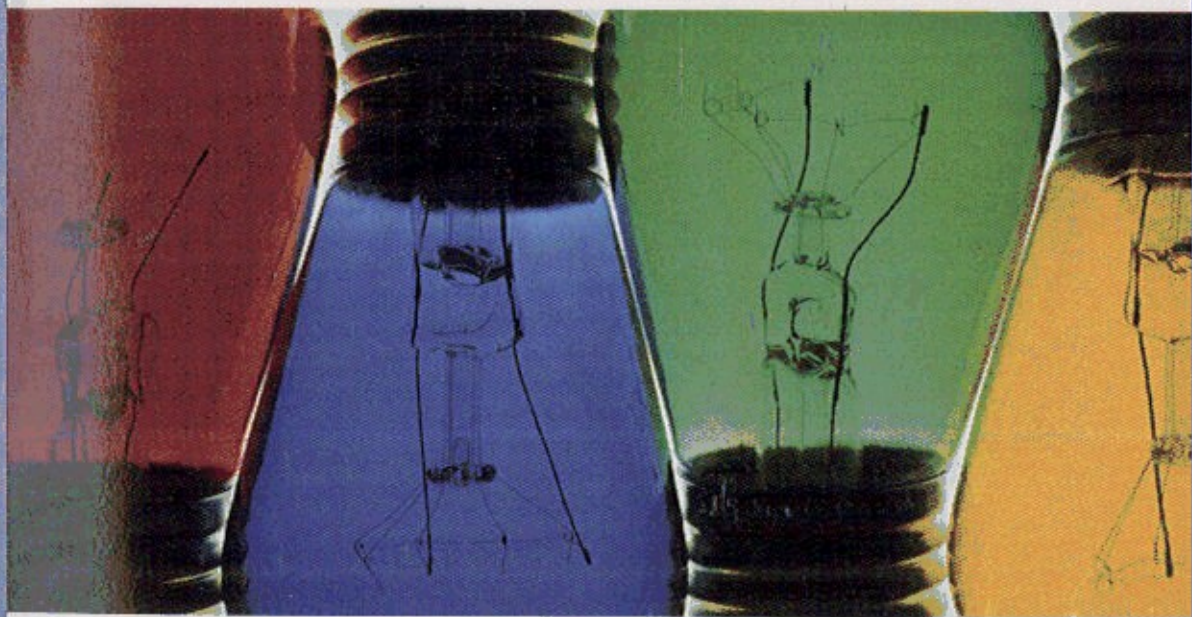


INTERNATIONAL EDITION

THIRD
EDITION

**Data Structures
& Problem
Solving Using**

Java™



MARK ALLEN WEISS

contents

part one **Tour of Java**

chapter 1 **primitive java** 3

- 1.1 the general environment** 4
- 1.2 the first program** 5
 - 1.2.1 comments 5
 - 1.2.2 main 6
 - 1.2.3 terminal output 6
- 1.3 primitive types** 6
 - 1.3.1 the primitive types 6
 - 1.3.2 constants 7
 - 1.3.3 declaration and initialization of primitive types 7
 - 1.3.4 terminal input and output 8
- 1.4 basic operators** 8
 - 1.4.1 assignment operators 9
 - 1.4.2 binary arithmetic operators 10
 - 1.4.3 unary operators 10
 - 1.4.4 type conversions 10
- 1.5 conditional statements** 11
 - 1.5.1 relational and equality operators 11
 - 1.5.2 logical operators 12
 - 1.5.3 the if statement 13
 - 1.5.4 the while statement 14
 - 1.5.5 the for statement 14
 - 1.5.6 the do statement 15

1.5.7	break and continue	16
1.5.8	the switch statement	17
1.5.9	the conditional operator	17
1.6	methods	18
1.6.1	overloading of method names	19
1.6.2	storage classes	20
	summary	20
	key concepts	20
	common errors	22
	on the internet	23
	exercises	23
	references	25

chapter 2 reference types

27

2.1	what is a reference?	27
2.2	basics of objects and references	30
2.2.1	the dot operator (.)	30
2.2.2	declaration of objects	30
2.2.3	garbage collection	31
2.2.4	the meaning of =	32
2.2.5	parameter passing	33
2.2.6	the meaning of ==	33
2.2.7	no operator overloading for objects	34
2.3	strings	35
2.3.1	basics of string manipulation	35
2.3.2	string concatenation	35
2.3.3	comparing strings	36
2.3.4	other String methods	36
2.3.5	converting other types to strings	37
2.4	arrays	37
2.4.1	declaration, assignment, and methods	38
2.4.2	dynamic array expansion	40
2.4.3	ArrayList	43
2.4.4	multidimensional arrays	45
2.4.5	command-line arguments	46
2.4.6	enhanced for loop	46

2.5	exception handling	48
2.5.1	processing exceptions	48
2.5.2	the finally clause	48
2.5.3	common exceptions	50
2.5.4	the throw and throws clauses	51
2.6	input and output	51
2.6.1	basic stream operations	52
2.6.2	the StringTokenizer type	53
2.6.3	sequential files	55
	summary	58
	key concepts	58
	common errors	60
	on the internet	60
	exercises	61
	references	62

chapter 3 objects and classes

63

3.1	what is object-oriented programming?	63
3.2	a simple example	65
3.3	javadoc	67
3.4	basic methods	70
3.4.1	constructors	70
3.4.2	mutators and accessors	70
3.4.3	output and toString	72
3.4.4	equals	72
3.4.5	main	72
3.4.6	static fields and methods	72
3.5	additional constructs	73
3.5.1	the this reference	73
3.5.2	the this shorthand for constructors	74
3.5.3	the instanceof operator	74
3.5.4	instance members versus static members	75
3.5.5	static fields and methods	75
3.5.6	static initializers	77

3.6	packages	78
3.6.1	the import directive	78
3.6.2	the package statement	80
3.6.3	the CLASSPATH environment variable	81
3.6.4	package visibility rules	82
3.7	a design pattern: composite (pair)	82
	summary	83
	key concepts	84
	common errors	87
	on the Internet	87
	exercises	88
	references	91

chapter 4 inheritance 93

4.1	what is inheritance?	94
4.1.1	creating new classes	94
4.1.2	type compatibility	99
4.1.3	dynamic dispatch and polymorphism	100
4.1.4	inheritance hierarchies	101
4.1.5	visibility rules	101
4.1.6	the constructor and super	102
4.1.7	final methods and classes	103
4.1.8	overriding a method	105
4.1.9	type compatibility revisited	105
4.1.10	compatibility of array types	108
4.1.11	covariant return types	108
4.2	designing hierarchies	109
4.2.1	abstract methods and classes	110
4.3	multiple inheritance	114
4.4	the interface	115
4.4.1	specifying an interface	115
4.4.2	implementing an interface	116
4.4.3	multiple interfaces	117
4.4.4	interfaces are abstract classes	117
4.5	fundamental inheritance in Java	118
4.5.1	the Object class	118

4.5.2	the hierarchy of exceptions	119
4.5.3	i/o: the decorator pattern	120
4.6	implementing generic components using inheritance	124
4.6.1	using Object for genericity	124
4.6.2	wrappers for primitive types	125
4.6.3	autoboxing/unboxing	127
4.6.4	adapters: changing an interface	128
4.6.5	using interface types for genericity	129
4.7	implementing generic components using java 1.5 generics	131
4.7.1	simple generic classes and interfaces	131
4.7.2	wildcards with bounds	132
4.7.3	generic static methods	133
4.7.4	type bounds	134
4.7.5	type erasure	135
4.7.6	restrictions on generics	136
4.8	the functor (function objects)	137
4.8.1	nested classes	142
4.8.2	local classes	143
4.8.3	anonymous classes	144
4.8.4	nested classes and generics	145
4.9	dynamic dispatch details	146
	summary	149
	key concepts	150
	common errors	152
	on the internet	153
	exercises	154
	references	160

part two **Algorithms and Building Blocks**

chapter 5 algorithm analysis 163

5.1	what is algorithm analysis?	164
5.2	examples of algorithm running times	168

5.3	the maximum contiguous subsequence sum problem	169
5.3.1	the obvious $O(N^3)$ algorithm	170
5.3.2	an improved $O(N^2)$ algorithm	173
5.3.3	a linear algorithm	173
5.4	general big-oh rules	177
5.5	the logarithm	181
5.6	static searching problem	183
5.6.1	sequential search	183
5.6.2	binary search	184
5.6.3	interpolation search	187
5.7	checking an algorithm analysis	188
5.8	limitations of big-oh analysis	189
	summary	190
	key concepts	190
	common errors	191
	on the Internet	192
	exercises	192
	references	199

chapter 6 the collections api 201

6.1	introduction	202
6.2	the iterator pattern	203
6.2.1	basic iterator design	204
6.2.2	inheritance-based iterators and factories	206
6.3	collections api: containers and iterators	208
6.3.1	the Collection interface	209
6.3.2	Iterator interface	212
6.4	generic algorithms	214
6.4.1	Comparator function objects	215
6.4.2	the Collections class	215
6.4.3	binary search	218
6.4.4	sorting	218
6.5	the List interface	220
6.5.1	the ListIterator interface	221
6.5.2	LinkedList class	223

6.6	stacks and queues	225
6.6.1	stacks	225
6.6.2	stacks and computer languages	226
6.6.3	queues	227
6.6.4	stacks and queues in the collections api	228
6.7	sets	228
6.7.1	the TreeSet class	229
6.7.2	the HashSet class	231
6.8	maps	236
6.9	priority queues	239
	summary	243
	key concepts	244
	common errors	245
	on the internet	245
	exercises	246
	references	249

chapter 7 recursion

251

7.1	what is recursion?	252
7.2	background: proofs by mathematical induction	253
7.3	basic recursion	255
7.3.1	printing numbers in any base	257
7.3.2	why it works	259
7.3.3	how it works	261
7.3.4	too much recursion can be dangerous	262
7.3.5	preview of trees	263
7.3.6	additional examples	264
7.4	numerical applications	269
7.4.1	modular arithmetic	269
7.4.2	modular exponentiation	270
7.4.3	greatest common divisor and multiplicative inverses	272
7.4.4	the rsa cryptosystem	275
7.5	divide-and-conquer algorithms	277
7.5.1	the maximum contiguous subsequence sum problem	278
7.5.2	analysis of a basic divide-and-conquer recurrence	281
7.5.3	a general upper bound for divide-and-conquer running times	285

7.6 dynamic programming 287**7.7 backtracking 291****summary 294****key concepts 296****common errors 297****on the internet 297****exercises 298****references 302****chapter 8 sorting algorithms****303****8.1 why is sorting important? 304****8.2 preliminaries 305****8.3 analysis of the insertion sort and other simple sorts 305****8.4 shellsort 309**

8.4.1 performance of shellsort 310

8.5 mergesort 313

8.5.1 linear-time merging of sorted arrays 313

8.5.2 the mergesort algorithm 315

8.6 quicksort 316

8.6.1 the quicksort algorithm 317

8.6.2 analysis of quicksort 320

8.6.3 picking the pivot 323

8.6.4 a partitioning strategy 325

8.6.5 keys equal to the pivot 327

8.6.6 median-of-three partitioning 327

8.6.7 small arrays 328

8.6.8 java quicksort routine 329

8.7 quickselect 331**8.8 a lower bound for sorting 332****summary 334****key concepts 335****common errors 336****on the internet 336****exercises 336****references 341**

chapter 9 randomization 343

- 9.1 why do we need random numbers?** 343
- 9.2 random number generators** 344
- 9.3 nonuniform random numbers** 349
- 9.4 generating a random permutation** 353
- 9.5 randomized algorithms** 354
- 9.6 randomized primality testing** 357
- summary** 360
- key concepts** 360
- common errors** 361
- on the internet** 362
- exercises** 362
- references** 364

part three Applications

chapter 10 fun and games 367

- 10.1 word search puzzles** 367
 - 10.1.1 theory 368
 - 10.1.2 java implementation 369
- 10.2 the game of tic-tac-toe** 373
 - 10.2.1 alpha-beta pruning 374
 - 10.2.2 transposition tables 377
 - 10.2.3 computer chess 381
- summary** 384
- key concepts** 384
- common errors** 384
- on the internet** 384
- exercises** 385
- references** 387

chapter 11 stacks and compilers**389****11.1 balanced-symbol checker 389**

11.1.1 basic algorithm 390

11.1.2 implementation 391

11.2 a simple calculator 400

11.2.1 postfix machines 402

11.2.2 infix to postfix conversion 403

11.2.3 implementation 405

11.2.4 expression trees 414

summary 415

key concepts 416

common errors 416

on the internet 417

exercises 417

references 418

chapter 12 utilities**419****12.1 file compression 420**

12.1.1 prefix codes 421

12.1.2 huffman's algorithm 423

12.1.3 implementation 425

12.2 a cross-reference generator 441

12.2.1 basic ideas 441

12.2.2 java implementation 441

summary 445

key concepts 446

common errors 446

on the internet 446

exercises 446

references 450

chapter 13 simulation**451****13.1 the josephus problem 451**

13.1.1 the simple solution 453

13.1.2 a more efficient algorithm 453

13.2 event-driven simulation 457

13.2.1 basic ideas 457

13.2.2 example: a modern bank simulation 458

summary 466

key concepts 466

common errors 467

on the internet 467

exercises 467

chapter 14 graphs and paths

471

14.1 definitions 472

14.1.1 representation 474

14.2 unweighted shortest-path problem 483

14.2.1 theory 483

14.2.2 java implementation 489

14.3 positive-weighted, shortest-path problem 489

14.3.1 theory: dijkstra's algorithm 490

14.3.2 java implementation 494

14.4 negative-weighted, shortest-path problem 496

14.4.1 theory 496

14.4.2 java implementation 497

14.5 path problems in acyclic graphs 499

14.5.1 topological sorting 499

14.5.2 theory of the acyclic shortest-path algorithm 501

14.5.3 java implementation 501

14.5.4 an application: critical-path analysis 504

summary 506

key concepts 507

common errors 508

on the internet 509

exercises 509

references 513

part four **Implementations**

chapter 15 inner classes and implementation of ArrayList

- 15.1 iterators and nested classes** 518
- 15.2 iterators and inner classes** 520
- 15.3 the AbstractCollection class** 524
- 15.4 StringBuilder** 528
- 15.5 implementation of ArrayList with an iterator** 529
- summary** 534
- key concepts** 535
- common errors** 535
- on the internet** 535
- exercises** 535

chapter 16 stacks and queues **539**

- 16.1 dynamic array implementations** 539
 - 16.1.1 stacks 540
 - 16.1.2 queues 544
- 16.2 linked list implementations** 549
 - 16.2.1 stacks 550
 - 16.2.2 queues 553
- 16.3 comparison of the two methods** 557
- 16.4 the java.util.Stack class** 557
- 16.5 double-ended queues** 559
- summary** 559
- key concepts** 559
- common errors** 559
- on the internet** 559
- exercises** 560

chapter 17 linked lists 563

- 17.1 basic ideas 563**
 - 17.1.1 header nodes 565
 - 17.1.2 iterator classes 566
- 17.2 java implementation 568**
- 17.3 doubly linked lists and circularly linked lists 574**
- 17.4 sorted linked lists 577**
- 17.5 implementing the collections api LinkedList class 579**
 - summary 590
 - key concepts 590
 - common errors 591
 - on the internet 591
 - exercises 591

chapter 18 trees 595

- 18.1 general trees 595**
 - 18.1.1 definitions 596
 - 18.1.2 implementation 597
 - 18.1.3 an application: file systems 598
- 18.2 binary trees 602**
- 18.3 recursion and trees 609**
- 18.4 tree traversal: iterator classes 611**
 - 18.4.1 postorder traversal 615
 - 18.4.2 inorder traversal 619
 - 18.4.3 preorder traversal 619
 - 18.4.4 level-order traversals 622
- summary 623
- key concepts 624
- common errors 625
- on the internet 626
- exercises 626

chapter 19 binary search trees 629

- 19.1 basic ideas 629**
 - 19.1.1 the operations 630
 - 19.1.2 java implementation 632
- 19.2 order statistics 639**
 - 19.2.1 java implementation 640
- 19.3 analysis of binary search tree operations 644**
- 19.4 avl trees 648**
 - 19.4.1 properties 649
 - 19.4.2 single rotation 651
 - 19.4.3 double rotation 654
 - 19.4.4 summary of avl insertion 656
- 19.5 red-black trees 657**
 - 19.5.1 bottom-up insertion 658
 - 19.5.2 top-down red-black trees 660
 - 19.5.3 java implementation 661
 - 19.5.4 top-down deletion 668
- 19.6 aa-trees 670**
 - 19.6.1 insertion 672
 - 19.6.2 deletion 674
 - 19.6.3 java implementation 675
- 19.7 implementing the collections api TreeSet and
 TreeMap classes 680**
- 19.8 b-trees 698**
 - summary 704
 - key concepts 705
 - common errors 706
 - on the internet 706
 - exercises 707
 - references 709

chapter 20 hash tables 713

- 20.1 basic ideas 714**
- 20.2 hash function 715**

20.3	linear probing	717
20.3.1	naive analysis of linear probing	719
20.3.2	what really happens: primary clustering	720
20.3.3	analysis of the find operation	721
20.4	quadratic probing	723
20.4.1	java implementation	727
20.4.2	analysis of quadratic probing	733
20.5	separate chaining hashing	736
20.6	hash tables versus binary search trees	737
20.7	hashing applications	737
	summary	738
	key concepts	738
	common errors	739
	on the internet	740
	exercises	740
	references	742

chapter 21 a priority queue: the binary heap

745

21.1	basic ideas	746
21.1.1	structure property	746
21.1.2	heap-order property	748
21.1.3	allowed operations	749
21.2	implementation of the basic operations	752
21.2.1	insertion	752
21.2.2	the deleteMin operation	754
21.3	the buildHeap operation: linear-time heap construction	756
21.4	advanced operations: decreaseKey and merge	761
21.5	internal sorting: heapsort	761
21.6	external sorting	764
21.6.1	why we need new algorithms	764
21.6.2	model for external sorting	765
21.6.3	the simple algorithm	765
21.6.4	multiway merge	767
21.6.5	polyphase merge	768
21.6.6	replacement selection	770

summary 771
key concepts 772
common errors 772
on the internet 773
exercises 773
references 777

part five **Advanced Data Structures**

chapter 22 splay trees 781

22.1 self-adjustment and amortized analysis 782
22.1.1 amortized time bounds 783
22.1.2 a simple self-adjusting strategy (that does not work) 783
22.2 the basic bottom-up splay tree 785
22.3 basic splay tree operations 788
22.4 analysis of bottom-up splaying 789
22.4.1 proof of the splaying bound 792
22.5 top-down splay trees 795
22.6 implementation of top-down splay trees 798
22.7 comparison of the splay tree with other search trees 803
summary 804
key concepts 804
common errors 805
on the internet 805
exercises 805
references 806

chapter 23 merging priority queues 809

23.1 the skew heap 809
23.1.1 merging is fundamental 810
23.1.2 simplistic merging of heap-ordered trees 810

23.1.3	the skew heap: a simple modification	811
23.1.4	analysis of the skew heap	812
23.2	the pairing heap	814
23.2.1	pairing heap operations	815
23.2.2	implementation of the pairing heap	816
23.2.3	application: dijkstra's shortest weighted path algorithm	822
	summary	826
	key concepts	826
	common errors	826
	on the internet	827
	exercises	827
	references	828

chapter 24 the disjoint set class

831

24.1	equivalence relations	832
24.2	dynamic equivalence and applications	832
24.2.1	application: generating mazes	833
24.2.2	application: minimum spanning trees	836
24.2.3	application: the nearest common ancestor problem	839
24.3	the quick-find algorithm	842
24.4	the quick-union algorithm	843
24.4.1	smart union algorithms	845
24.4.2	path compression	847
24.5	java implementation	848
24.6	worst case for union-by-rank and path compression	851
24.6.1	analysis of the union/find algorithm	852
	summary	859
	key concepts	859
	common errors	860
	on the internet	860
	exercises	861
	references	863

appendix A operators 865

appendix B graphical user interfaces 867

- B.1 the abstract window toolkit and swing 868**
- B.2 basic objects in swing 869**
 - B.2.1 Component 870
 - B.2.2 Container 871
 - B.2.3 top-level containers 871
 - B.2.4 JPanel 872
 - B.2.5 important i/o components 874
- B.3 basic principles 878**
 - B.3.1 layout managers 879
 - B.3.2 graphics 883
 - B.3.3 events 885
 - B.3.4 event handling: adapters and anonymous inner classes 887
 - B.3.5 summary: putting the pieces together 889
 - B.3.6 is this everything i need to know about swing? 890
- summary 891**
- key concepts 891**
- common errors 893**
- on the internet 894**
- exercises 894**
- references 895**

appendix C bitwise operators 897

index 901