

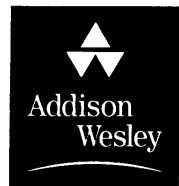


DATA STRUCTURES AND PROBLEM SOLVING USING JAVA

Second Edition

MARK ALLEN WEISS

Florida International University



Boston San Francisco New York

London Toronto Sydney Tokyo Singapore Madrid

Mexico City Munich Paris Cape Town Hong Kong Montreal

<i>Executive Editor</i>	Susan Hartman Sullivan
<i>Project Editor</i>	Katherine Harutunian
<i>Executive Marketing Manager</i>	Michael Hirsch
<i>Production and Editorial Services</i>	Argosy
<i>Technical Art</i>	Argosy
<i>Cover Design</i>	Gina Hagen
<i>Cover Photo</i>	©PhotoDisc
<i>Design Manager</i>	Gina Hagen
<i>Prepress and Manufacturing</i>	Caroline Fell

Access the latest information about Addison-Wesley titles from our World Wide Web site: <http://www.aw.com/cs>

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial caps or all caps.

The programs and applications presented in this book have been included for their instructional value. They have been tested with care, but are not guaranteed for any particular purpose. The publisher does not offer any warranties or representations, nor does it accept any liabilities with respect to the programs or applications.

Library of Congress Cataloging-in-Publication Data

Weiss, Mark Allen.

Data structures and problem solving using Java / Mark Weiss.--2nd ed.
p. cm.

ISBN 0-201-74835-5 (alk. Paper)

1. Java (Computer program language) 2. Data structures (Computer science)
3. Problem solving--Data processing. I. Title.

QA76.73.J38 W45 2001

005.7'3--dc21

2001027939

CIP

Copyright © 2002 by Addison Wesley, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America.

12345678910- DOC-04030201



CONTENTS

PART I: 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	8
	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
	Objects of the Game	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 of 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 45
- 2.5 Exception Handling 47**
 - 2.5.1 Processing Exceptions 47
 - 2.5.2 The finally Clause 47
 - 2.5.3 Common Exceptions 49
 - 2.5.4 The throw and throws Clauses 50
- 2.6 Input and Output 50**
 - 2.6.1 Basic Stream Operations 51
 - 2.6.2 The StringTokenizer Type 52
 - 2.6.3 Sequential Files 53

Summary 55**Objects of the Game 57****Common Errors 58****On the Internet 59****Exercises 59****References 60****CHAPTER 3 Objects and Classes 61**

- 3.1 What Is Object-Oriented Programming? 61**
- 3.2 A Simple Example 63**
- 3.3 Javadoc 65**
- 3.4 Basic Methods 68**
 - 3.4.1 Constructors 68
 - 3.4.2 Mutators and Accessors 68
 - 3.4.3 Output and toString 70
 - 3.4.4 equals 70
 - 3.4.5 main 70
 - 3.4.6 Static Fields and Methods 71
- 3.5 Additional Constructs 71**
 - 3.5.1 The this Reference 71
 - 3.5.2 The this Shorthand for Constructors 72
 - 3.5.3 The instanceof Operator 72

- 3.5.4 Instance Members vs. Static Members 73
- 3.5.5 Static Fields and Methods 73
- 3.5.6 Static Initializers 74
- 3.6 Packages 76**
 - 3.6.1 The import Directive 76
 - 3.6.2 The package Statement 78
 - 3.6.3 The CLASSPATH Environment Variable 78
 - 3.6.4 Package Visibility Rules 79

3.7 A Design Pattern: Composite (Pair) 80**Summary 81****Objects of the Game 83****Common Errors 84****On the Internet 85****Exercises 85****References 88****CHAPTER 4 Inheritance 89****4.1 What Is Inheritance? 90**

- 4.1.1 Creating New Classes 90
- 4.1.2 Type Compatibility 95
- 4.1.3 Dynamic Binding and Polymorphism 96
- 4.1.4 Inheritance Hierarchies 97
- 4.1.5 Visibility Rules 97
- 4.1.6 The Constructor and super 98
- 4.1.7 final Methods and Classes 99
- 4.1.8 Overriding a Method 100
- 4.1.9 Type Compatibility Revisited 101

4.2 Designing Hierarchies 103

- 4.2.1 Abstract Methods and Classes 107

4.3 Multiple Inheritance 109**4.4 The Interface 110**

- 4.4.1 Specifying an Interface 110
- 4.4.2 Implementing an Interface 111
- 4.4.3 Multiple Interfaces 112
- 4.4.4 Interfaces are Abstract Classes 112

- 4.5 Fundamental Inheritance in Java 113**
 - 4.5.1 The Object Class 113
 - 4.5.2 The Hierarchy of Exceptions 113
 - 4.5.3 I/O: The Decorator Pattern 113
- 4.6 Implementing Generic Components 118**
 - 4.6.1 Using Object for Genericity 119
 - 4.6.2 Wrappers for Primitive Types 120
 - 4.6.3 Adapters: Changing an Interface 120
 - 4.6.4 Using Interface Types for Genericity 123
- 4.7 The Functor (Function Objects) 125**
 - 4.7.1 Nested Classes 128
 - 4.7.2 Local Classes 129
 - 4.7.3 Anonymous Classes 130
- 4.8 Dynamic Binding Details 131**
 - Summary 135
 - Objects of the Game 136
 - Common Errors 138
 - On the Internet 138
 - Exercises 140
 - References 144

PART II: ALGORITHMS AND BUILDING BLOCKS

CHAPTER 5 Algorithm Analysis 147

- 5.1 What Is Algorithm Analysis? 148**
- 5.2 Examples of Algorithm Running Times 151**
- 5.3 The Maximum Contiguous Subsequence Sum Problem 153**
 - 5.3.1 The Obvious $O(N^3)$ Algorithm 154
 - 5.3.2 An Improved $O(N^2)$ Algorithm 157
 - 5.3.3 A Linear Algorithm 158
- 5.4 General Big-Oh Rules 160**
- 5.5 The Logarithm 164**
- 5.6 Static Searching Problem 167**
 - 5.6.1 Sequential Search 167
 - 5.6.2 Binary Search 168

- 5.6.3 Interpolation Search 170
- 5.7 Checking an Algorithm Analysis 171**
- 5.8 Limitations of Big-Oh Analysis 173**
 - Summary 174
 - Objects of the Game 174
 - Common Errors 175
 - On the Internet 175
 - Exercises 176
 - References 181

CHAPTER 6 The Collections API 183

- 6.1 Introduction 184**
- 6.2 The Iterator Pattern 185**
 - 6.2.1 Basic Iterator Design 186
 - 6.2.2 Inheritance-based Iterators and Factories 188
- 6.3 Collections API: Containers and Iterators 190**
 - 6.3.1 The Collection Interface 190
 - 6.3.2 Iterator Interface 193
- 6.4 Generic Algorithms 195**
 - 6.4.1 Comparator Function Objects 195
 - 6.4.2 The Collections Class 195
 - 6.4.3 Binary Search 198
 - 6.4.4 Sorting 199
- 6.5 The List Interface 199**
 - 6.5.1 The ListIterator Interface 201
 - 6.5.2 LinkedList Class 202
- 6.6 Stacks and Queues 205**
 - 6.6.1 Stacks 205
 - 6.6.2 Stacks and Computer Languages 206
 - 6.6.3 Queues 208
 - 6.6.4 Stacks and Queues in the Collections API 208
- 6.7 Sets 209**
 - 6.7.1 The TreeSet Class 210
 - 6.7.2 The HashSet Class 211

	6.8 Maps 216
	6.9 Priority Queues 220
	Summary 224
	Objects of the Game 225
	Common Errors 226
	On the Internet 226
	Exercises 227
	References 230
CHAPTER 7	Recursion 231
	7.1 What Is Recursion? 232
	7.2 Background: Proofs by Mathematical Induction 233
	7.3 Basic Recursion 235
	7.3.1 Printing Numbers in Any Base 237
	7.3.2 Why It Works 239
	7.3.3 How It Works 241
	7.3.4 Too Much Recursion Can Be Dangerous 242
	7.3.5 Preview of Trees 243
	7.3.6 Additional Examples 244
	7.4 Numerical Applications 248
	7.4.1 Modular Arithmetic 250
	7.4.2 Modular Exponentiation 250
	7.4.3 Greatest Common Divisor and Multiplicative Inverses 252
	7.4.4 The RSA Cryptosystem 254
	7.5 Divide-and-Conquer Algorithms 257
	7.5.1 The Maximum Contiguous Subsequence Sum Problem 25
	7.5.2 Analysis of a Basic Divide-and-Conquer Recurrence 260
	7.5.3 A General Upper Bound for Divide-and-Conquer Running Times 265
	7.6 Dynamic Programming 267
	7.7 Backtracking 272
	Summary 276
	Objects of the Game 276
	Common Errors 277

	On the Internet 278
	Exercises 278
	References 282

CHAPTER 8	Sorting Algorithms 283
	8.1 Why Is Sorting Important? 284
	8.2 Preliminaries 285
	8.3 Analysis of the Insertion Sort and Other Simple Sorts 285
	8.4 Shellsort 289
	8.4.1 Performance of Shellsort 290
	8.5 Mergesort 293
	8.5.1 Linear-Time Merging of Sorted Arrays 293
	8.5.2 The Mergesort Algorithm 295
	8.6 Quicksort 295
	8.6.1 The Quicksort Algorithm 296
	8.6.2 Analysis of Quicksort 299
	8.6.3 Picking the Pivot 303
	8.6.4 A Partitioning Strategy 304
	8.6.5 Keys Equal to the Pivot 307
	8.6.6 Median-of-Three Partitioning 307
	8.6.7 Small Arrays 308
	8.6.8 Java Quicksort Routine 309
	8.7 Quickselect 311
	8.8 A Lower Bound for Sorting 312
	Summary 314
	Objects of the Game 315
	Common Errors 315
	On the Internet 316
	Exercises 316
	References 320
CHAPTER 9	Randomization 323
	9.1 Why Do We Need Random Numbers? 323
	9.2 Random Number Generators 324

- 9.3 Nonuniform Random Numbers 330**
- 9.4 Generating a Random Permutation 333**
- 9.5 Randomized Algorithms 334**
- 9.6 Randomized Primality Testing 337**
- Summary 340**
- Objects of the Game 340**
- Common Errors 341**
- On the Internet 342**
- Exercises 342**
- References 344**

PART III: APPLICATIONS

CHAPTER 10 Fun and Games 347

- 10.1 Word Search Puzzles 347**
 - 10.1.1 Theory 348
 - 10.1.2 Java Implementation 350
- 10.2 The Game of Tic-Tac-Toe 350**
 - 10.2.1 Alpha–Beta Pruning 353
 - 10.2.2 Transposition Tables 359
 - 10.2.3 Computer Chess 362
- Summary 364**
- Objects of the Game 364**
- Common Errors 364**
- On the Internet 365**
- Exercises 365**
- References 367**

CHAPTER 11 Stacks and Compilers 369

- 11.1 Balanced-Symbol Checker 369**
 - 11.1.1 Basic Algorithm 370
 - 11.1.2 Implementation 371
- 11.2 A Simple Calculator 380**
 - 11.2.1 Postfix Machines 382

- 11.2.2 Infix to Postfix Conversion 383
- 11.2.3 Implementation 386
- 11.2.4 Expression Trees 391

- Summary 395**
- Objects of the Game 396**
- Common Errors 396**
- On the Internet 397**
- Exercises 397**
- References 398**

CHAPTER 12 Utilities 399

- 12.1 File Compression 399**
 - 12.1.1 Prefix Codes 401
 - 12.1.2 Huffman’s Algorithm 403
 - 12.1.3 Implementation 405
- 12.2 A Cross-Reference Generator 420**
 - 12.2.1 Basic Ideas 420
 - 12.2.2 Java Implementation 420
- Summary 424**
- Objects of the Game 425**
- Common Errors 425**
- On the Internet 425**
- Exercises 425**
- References 428**

CHAPTER 13 Simulation 429

- 13.1 The Josephus Problem 429**
 - 13.1.1 The Simple Solution 431
 - 13.1.2 A More Efficient Algorithm 431
- 13.2 Event-Driven Simulation 435**
 - 13.2.1 Basic Ideas 435
 - 13.2.2 Example: A Modem Bank Simulation 436
- Summary 444**
- Objects of the Game 444**
- Common Errors 444**

On the Internet 444**Exercises 445****CHAPTER 14 Graphs and Paths 447****14.1 Definitions 448**

14.1.1 Representation 449

14.2 Unweighted Shortest-Path Problem 459

14.2.1 Theory 462

14.2.2 Java Implementation 466

14.3 Positive-Weighted, Shortest-Path Problem 467

14.3.1 Theory: Dijkstra's Algorithm 467

14.3.2 Java Implementation 469

14.4 Negative-Weighted, Shortest-Path Problem 471

14.4.1 Theory 473

14.4.2 Java Implementation 474

14.5 Path Problems in Acyclic Graphs 474

14.5.1 Topological Sorting 474

14.5.2 Theory of the Acyclic Shortest-Path Algorithm 476

14.5.3 Java Implementation 478

14.5.4 An Application: Critical-Path Analysis 480

Summary 483**Objects of the Game 483****Common Errors 485****On the Internet 485****Exercises 485****References 489****PART IV: IMPLEMENTATIONS****CHAPTER 15 Inner Classes and Implementation of ArrayList 493****15.1 Iterators and Nested Classes 493****15.2 Iterators and Inner Classes 495****15.3 The AbstractCollection Class 500****15.4 Implementation of ArrayList with an Iterator 500****Summary 508****Objects of the Game 508****Common Error 508****On the Internet 509****Exercises 509****CHAPTER 16 Stacks and Queues 513****16.1 Dynamic Array Implementations 513**

16.1.1 Stacks 514

16.1.2 Queues 518

16.2 Linked List Implementations 524

16.2.1 Stacks 524

16.2.2 Queues 525

16.3 Comparison of the Two Methods 530**16.4 The java.util.Stack Class 531****16.5 Double-Ended Queues 533****Summary 534****Objects of the Game 534****Common Errors 534****On the Internet 534****Exercises 535****CHAPTER 17 Linked Lists 537****17.1 Basic Ideas 537**

17.1.1 Header Nodes 539

17.1.2 Iterator Classes 541

17.2 Java Implementation 542**17.3 Doubly Linked Lists and Circularly Linked Lists 548****17.4 Sorted Linked Lists 551****17.5 Implementing the Collections API LinkedList Class 551****Summary 563****Objects of the Game 563****Common Errors 563****On the Internet 563****Exercises 564**

CHAPTER 18 Trees 569**18.1 General Trees 569**

18.1.1 Definitions 570

18.1.2 Implementation 571

18.1.3 An Application: File Systems 572

18.2 Binary Trees 576**18.3 Recursion and Trees 582****18.4 Tree Traversal: Iterator Classes 585**

18.4.1 Postorder Traversal 589

18.4.2 Inorder Traversal 592

18.4.3 Preorder Traversal 592

18.4.4 Level-Order Traversals 596

Summary 597

Objects of the Game 598

Common Errors 599

On the Internet 599

Exercises 600

CHAPTER 19 Binary Search Trees 603**19.1 Basic Ideas 603**

19.1.1 The Operations 604

19.1.2 Java Implementation 606

19.2 Order Statistics 613

19.2.1 Java Implementation 614

19.3 Analysis of Binary Search Tree Operations 618**19.4 AVL Trees 621**

19.4.1 Properties 622

19.4.2 Single Rotation 624

19.4.3 Double Rotation 627

19.4.4 Summary of AVL Insertion 630

19.5 Red–Black Trees 630

19.5.1 Bottom–Up Insertion 631

19.5.2 Top–Down Red–Black Trees 634

19.5.3 Java Implementation 636

19.5.4 Top–Down Deletion 641

19.6 AA-Trees 644

19.6.1 Insertion 646

19.6.2 Deletion 648

19.6.3 Java Implementation 649

19.7 Implementing the Collections API TreeSet and TreeMap Classes 654**19.8 B-Trees 663**

Summary 675

Objects of the Game 676

Common Errors 677

On the Internet 677

Exercises 678

References 680

CHAPTER 20 Hash Tables 683**20.1 Basic Ideas 684****20.2 Hash Function 685****20.3 Linear Probing 687**

20.3.1 Naive Analysis of Linear Probing 689

20.3.2 What Really Happens: Primary Clustering 690

20.3.3 Analysis of the find Operation 691

20.4 Quadratic Probing 693

20.4.1 Java Implementation 697

20.4.2 Analysis of Quadratic Probing 703

20.5 Separate Chaining Hashing 706**20.6 Hash Tables Versus Binary Search Trees 707****20.7 Hashing Applications 707**

Summary 708

Objects of the Game 708

Common Errors 709

On the Internet 710

Exercises 710

References 712

CHAPTER 21 A Priority Queue: The Binary Heap 715**21.1 Basic Ideas 716**

- 21.1.1 Structure Property 716
- 21.1.2 Heap-Order Property 718
- 21.1.3 Allowed Operations 718

21.2 Implementation of the Basic Operations 721

- 21.2.1 The insert Operation 722
- 21.2.2 The deleteMin Operation 723

21.3 The buildHeap Operation: Linear-Time Heap Construction 726**21.4 Advanced Operations: decreaseKey and merge 731****21.5 Internal Sorting: Heapsort 731****21.6 External Sorting 734**

- 21.6.1 Why We Need New Algorithms 734
- 21.6.2 Model for External Sorting 735
- 21.6.3 The Simple Algorithm 735
- 21.6.4 Multiway Merge 737
- 21.6.5 Polyphase Merge 738
- 21.6.6 Replacement Selection 740

Summary 741**Objects of the Game 741****Common Errors 742****On the Internet 742****Exercises 743****References 747****PART V: ADVANCED DATA STRUCTURES****CHAPTER 22 Splay Trees 751****22.1 Self-Adjustment and Amortized Analysis 752**

- 22.1.1 Amortized Time Bounds 753
- 22.1.2 A Simple Self-Adjusting Strategy (That Does Not Work) 753

22.2 The Basic Bottom-Up Splay Tree 755**22.3 Basic Splay Tree Operations 758****22.4 Analysis of Bottom-Up Splaying 759**

- 22.4.1 Proof of the Splaying Bound 762

22.5 Top-Down Splay Trees 765**22.6 Implementation of Top-Down Splay Trees 768****22.7 Comparison of the Splay Tree with Other Search Trees 771****Summary 773****Objects of the Game 773****Common Errors 776****On the Internet 776****Exercises 776****References 777****CHAPTER 23 Merging Priority Queues 779****23.1 The Skew Heap 779**

- 23.1.1 Merging Is Fundamental 780
- 23.1.2 Simplistic Merging of Heap-Ordered Trees 780
- 23.1.3 The Skew Heap: A Simple Modification 781
- 23.1.4 Analysis of the Skew Heap 782

23.2 The Pairing Heap 784

- 23.2.1 Pairing Heap Operations 785
- 23.2.2 Implementation of the Pairing Heap 786
- 23.2.3 Application: Dijkstra's Shortest Weighted Path Algorithm 791

Summary 796**Objects of the Game 796****Common Errors 796****On the Internet 797****Exercises 797****References 798****CHAPTER 24 The Disjoint Set Class 801****24.1 Equivalence Relations 802****24.2 Dynamic Equivalence and Applications 802**

- 24.2.1 Application: Generating Mazes 803
- 24.2.2 Application: Minimum Spanning Trees 806

24.2.3 Application: The Nearest Common Ancestor Problem	809
24.3 The Quick-Find Algorithm	813
24.4 The Quick-Union Algorithm	814
24.4.1 Smart Union Algorithms	815
24.4.2 Path Compression	817
24.5 Java Implementation	818
24.6 Worst Case for Union-by-Rank and Path Compression	821
24.6.1 Analysis of the Union/Find Algorithm	822
Summary	828
Objects of the Game	829
Common Errors	830
On the Internet	830
Exercises	830
References	833

APPENDICES

APPENDIX A Operators 835

APPENDIX B Graphical User Interfaces 837

B.1 The Abstract Window Toolkit and Swing	838
B.2 Basic Objects in Swing	839
B.2.1 Component	839
B.2.2 Container	841
B.2.3 Top-level Containers	841
B.2.4 JPanel	842
B.2.5 Important I/O Components	843
B.3 Basic Principles	848
B.3.1 Layout Managers	848
B.3.2 Graphics	852
B.3.3 Events	853
B.3.4 Event Handling: Adapters and Anonymous Inner Classes	856
B.3.5 Summary: Putting the Pieces Together	859
B.3.6 Is This Everything I Need To Know About Swing?	860

Summary	860
Objects of the Game	861
Common Errors	863
On the Internet	863
Exercises	863
Reference	865

APPENDIX C Bitwise Operators 867

Index	871
--------------	------------