

<b>preface</b>	<b>vii</b>
<b>chapter 1 software engineering</b>	<b>1</b>
<b>1.1 software development</b>	<b>2</b>
professionalism and ethics	3
<b>1.2 software quality</b>	<b>3</b>
correctness	4
reliability	4
robustness	5
usability	6
maintainability	6
reusability	6
portability	7
efficiency	8
quality issues	8
<b>1.3 development life cycle models</b>	<b>8</b>
the waterfall model	10
the spiral model	11
the evolutionary development model	12
<b>1.4 the unified modeling language (UML)</b>	<b>13</b>
UML relationships	15
<b>1.5 error handling</b>	<b>18</b>
<b>1.6 analysis of algorithms</b>	<b>19</b>
growth functions and big O() notation	20
comparing growth functions	22
analyzing loop execution	22
nested loops	24
<b>1.7 software engineering and data structures</b>	<b>25</b>

# contents

<b>chapter 2</b>	<b>collections</b>	<b>31</b>
<b>2.1</b>	<b>introduction to collections</b>	<b>32</b>
	abstraction	33
	the Java Collections API	35
<b>2.2</b>	<b>a bag collection</b>	<b>35</b>
	interfaces	37
	iterators	39
	exceptions	40
<b>2.3</b>	<b>using a bag: bingo</b>	<b>40</b>
<b>2.4</b>	<b>implementing a bag: with arrays</b>	<b>44</b>
	managing capacity	45
	the ArrayBag class	46
	the size and isEmpty operations	48
	the add operation	49
	the addAll operation	51
	the removeRandom operation	52
	the remove operation	53
	the union operation	54
	the contains operation	55
	the equals operation	56
	the iterator operation	57
	the toString operation	57
	UML description	59
<b>2.5</b>	<b>analysis of the array implementation of a bag</b>	<b>61</b>
	analysis of add	61
	analysis of remove	61
	analysis of removeRandom	62
	analysis of addAll	62
	analysis of find and contains	62
	analysis of union	62
	analysis of equals	63

<b>chapter 3</b>	<b>linked structures</b>	<b>69</b>
<b>3.1</b>	<b>references as links</b>	<b>70</b>
<b>3.2</b>	<b>managing linked lists</b>	<b>72</b>
	inserting nodes	72
	deleting nodes	74
	dummy nodes	75
<b>3.3</b>	<b>elements without links</b>	<b>75</b>
	doubly linked lists	76
<b>3.4</b>	<b>implementing a bag: with links</b>	<b>77</b>
	the <code>LinkedBag</code> class	77
	the <code>add</code> operation	80
	the <code>removeRandom</code> operation	81
	the <code>remove</code> operation	82
	the <code>iterator</code> operation	83
<b>3.5</b>	<b>analysis of the linked implementation of a bag</b>	<b>85</b>
	analysis of <code>add</code>	86
	analysis of <code>remove</code>	87
	analysis of <code>removeRandom</code>	87
<b>chapter 4</b>	<b>recursion</b>	<b>93</b>
<b>4.1</b>	<b>recursive thinking</b>	<b>94</b>
	infinite recursion	95
	recursion in math	95
<b>4.2</b>	<b>recursive programming</b>	<b>96</b>
	recursion vs. iteration	99
	direct vs. indirect recursion	99
<b>4.3</b>	<b>using recursion</b>	<b>100</b>
	traversing a maze	100
	the Towers of Hanoi	106
<b>4.4</b>	<b>analyzing recursive algorithms</b>	<b>111</b>

<b>chapter 5</b>	<b>searching and sorting</b>	<b>119</b>
<b>5.1</b>	<b>searching</b>	<b>120</b>
	linear search	120
	binary search	122
	comparing search algorithms	124
<b>5.2</b>	<b>sorting</b>	<b>125</b>
	selection sort	129
	insertion sort	131
	bubble sort	132
	quick sort	135
	merge sort	138
<b>chapter 6</b>	<b>stacks</b>	<b>145</b>
<b>6.1</b>	<b>a stack ADT</b>	<b>146</b>
<b>6.2</b>	<b>using stacks: evaluating postfix expressions</b>	<b>149</b>
<b>6.3</b>	<b>using stacks: simulating recursion</b>	<b>157</b>
<b>6.4</b>	<b>implementing stacks: with links</b>	<b>159</b>
	the push operation	160
	the pop operation	162
	other operations	162
<b>6.5</b>	<b>implementing stacks: with arrays</b>	<b>163</b>
	the push operation	164
	the pop operation	165
	other operations	166

<b>6.6</b>	<b>implementing stacks: the <code>java.util.Stack</code> class</b>	<b>166</b>
	unique operations	167
	inheritance and implementation	167
<b>6.7</b>	<b>analysis of stack implementations</b>	<b>168</b>
	analysis of <code>push</code>	169
	analysis of <code>pop</code>	170
<b>chapter 7</b>	<b>queues</b>	<b>177</b>
<b>7.1</b>	<b>a queue ADT</b>	<b>178</b>
<b>7.2</b>	<b>using queues: code keys</b>	<b>181</b>
<b>7.3</b>	<b>using queues: ticket counter simulation</b>	<b>185</b>
<b>7.4</b>	<b>using queues: radix sort</b>	<b>192</b>
<b>7.5</b>	<b>implementing queues: with links</b>	<b>197</b>
	the <code>enqueue</code> operation	198
	the <code>dequeue</code> operation	198
	other operations	200
<b>7.6</b>	<b>implementing queues: with arrays</b>	<b>201</b>
	the <code>enqueue</code> operation	202
	the <code>dequeue</code> operation	203
	other operations	204
<b>7.7</b>	<b>implementing queues: with circular arrays</b>	<b>204</b>
<b>7.8</b>	<b>analysis of queue implementations</b>	<b>209</b>
	<code>enqueue</code>	209
	<code>dequeue</code>	210

<b>chapter 8</b>	<b>lists</b>	<b>217</b>
<b>8.1</b>	<b>a list ADT</b>	<b>218</b>
<b>8.2</b>	<b>using ordered lists: tournament maker</b>	<b>226</b>
<b>8.3</b>	<b>using indexed lists: the Josephus problem</b>	<b>233</b>
<b>8.4</b>	<b>implementing lists: with arrays</b>	<b>237</b>
	the remove operation	238
	the contains operation	240
	the add operation for an ordered list	240
	operations particular to unordered lists	242
	operations particular to indexed lists	242
<b>8.5</b>	<b>implementing lists: with links</b>	<b>243</b>
	the remove operation	243
	doubly linked lists	245
<b>8.6</b>	<b>analysis of list implementations</b>	<b>248</b>
	analysis of ordered list implementations	249
	analysis of unordered list implementations	249
	analysis of indexed list implementations	254
<b>chapter 9</b>	<b>trees</b>	<b>261</b>
<b>9.1</b>	<b>trees</b>	<b>262</b>
	tree classifications	263
<b>9.2</b>	<b>strategies for implementing trees</b>	<b>265</b>
	computational strategy for array implementation of trees	265
	simulated link strategy for array implementation of trees	265
	analysis of trees	267

<b>9.3</b>	<b>tree traversals</b>	<b>267</b>
	preorder traversal	268
	inorder traversal	268
	postorder traversal	269
	level-order traversal	270
<b>9.4</b>	<b>implementing binary trees</b>	<b>271</b>
	the <code>removeLeftSubtree</code> method	277
	the <code>find</code> method	277
	the <code>iteratorInOrder</code> method	279
<b>9.5</b>	<b>using binary trees: expression trees</b>	<b>280</b>
<b>chapter 10</b>	<b>binary search trees</b>	<b>297</b>
<b>10.1</b>	<b>a binary search tree</b>	<b>298</b>
<b>10.2</b>	<b>implementing binary search trees: with links</b>	<b>300</b>
	the <code>addElement</code> operation	301
	the <code>removeElement</code> operation	302
	the <code>removeAllOccurrences</code> operation	306
	the <code>removeMin</code> operation	307
<b>10.3</b>	<b>using binary search trees: implementing ordered lists</b>	<b>309</b>
	analysis of the <code>BinarySearchTreeOrderedList</code> implementation	310
<b>10.4</b>	<b>balanced binary search trees</b>	<b>313</b>
	right rotation	314
	left rotation	315
	rightleft rotation	316
	leftright rotation	316

<b>10.5</b>	<b>implementing binary search trees: AVL trees</b>	<b>318</b>
	right rotation in an AVL tree	318
	left rotation in an AVL tree	319
	rightleft rotation in an AVL tree	319
	leftright rotation in an AVL tree	321
<b>10.6</b>	<b>implementing binary search trees: red/black trees</b>	<b>321</b>
	insertion into a red/black tree	322
	element removal from a red/black tree	326
<b>10.7</b>	<b>implementing binary search trees: the Java Collections API</b>	<b>328</b>
<b>chapter 11</b>	<b>heaps</b>	<b>339</b>
<b>11.1</b>	<b>a heap</b>	<b>340</b>
	the <code>addElement</code> operation	340
	the <code>removeMin</code> operation	343
	the <code>findMin</code> operation	344
<b>11.2</b>	<b>using heaps: heap sort</b>	<b>344</b>
<b>11.3</b>	<b>using heaps: priority queues</b>	<b>346</b>
<b>11.4</b>	<b>implementing heaps: with links</b>	<b>350</b>
	the <code>addElement</code> operation	351
	the <code>removeMin</code> operation	353
	the <code>findMin</code> operation	355
<b>11.5</b>	<b>implementing heaps: with arrays</b>	<b>356</b>
	the <code>addElement</code> operation	356
	the <code>removeMin</code> operation	357
	the <code>findMin</code> operation	359
<b>11.6</b>	<b>analysis of heap implementations</b>	<b>359</b>
	the <code>addElement</code> operation	359
	the <code>removeMin</code> operation	360
	the <code>findMin</code> operation	360
	heap sort	360

<b>chapter 12</b>	<b>multi-way search trees</b>	<b>367</b>
<b>12.1</b>	<b>combining tree concepts</b>	<b>368</b>
<b>12.2</b>	<b>2-3 trees</b>	<b>368</b>
	inserting elements into a 2-3 tree	369
	removing elements from a 2-3 tree	372
<b>12.3</b>	<b>2-4 trees</b>	<b>376</b>
<b>12.4</b>	<b>B-trees</b>	<b>377</b>
	motivation for B-trees	378
	B*-trees	379
	B <sup>+</sup> -trees	379
	analysis of B-trees	380
<b>12.5</b>	<b>implementation strategies     for B-trees</b>	<b>380</b>
<b>chapter 13</b>	<b>hashing</b>	<b>387</b>
<b>13.1</b>	<b>a hashing</b>	<b>388</b>
<b>13.2</b>	<b>hashing functions</b>	<b>390</b>
	the division method	391
	the folding method	391
	the mid-square method	392
	the radix transformation method	392
	the digit analysis method	393
	the length-dependent method	393
	hashing functions in the Java language	393
<b>13.3</b>	<b>resolving collisions</b>	<b>394</b>
	chaining	394
	open addressing	397

<b>13.4 deleting elements from a hash table</b>	<b>401</b>
deleting from a chained implementation	402
deleting from open addressing implementation	403
<b>13.5 hash tables in the Java Collections API</b>	<b>404</b>
Hashtable	405
HashSet	406
the HashMap class	408
the IdentityHashMap class	408
the WeakHashMap class	408
LinkedHashSet and LinkedHashMap	411
<b>chapter 14 graphs</b>	<b>419</b>
<b>14.1 undirected graphs</b>	<b>420</b>
<b>14.2 directed graphs</b>	<b>422</b>
<b>14.3 networks</b>	<b>423</b>
<b>14.4 common graph algorithms</b>	<b>425</b>
traversals	425
testing for connectivity	429
minimum spanning trees	430
determining the shortest path	431
<b>14.5 strategies for implementing graphs</b>	<b>435</b>
adjacency lists	436
adjacency matrices	436

<b>appendix a</b>	<b>object-oriented concepts in Java</b>	<b>443</b>
<b>A.1</b>	<b>an overview of object-orientation</b>	<b>443</b>
<b>A.2</b>	<b>using objects</b>	<b>445</b>
	abstraction	445
	creating objects	446
<b>A.3</b>	<b>class libraries and packages</b>	<b>448</b>
	the import declaration	449
<b>A.4</b>	<b>object state and behavior</b>	<b>450</b>
<b>A.5</b>	<b>classes</b>	<b>451</b>
	instance data	454
<b>A.6</b>	<b>encapsulation</b>	<b>454</b>
	visibility modifiers	455
	local data	457
<b>A.7</b>	<b>constructors</b>	<b>457</b>
<b>A.8</b>	<b>method overloading</b>	<b>458</b>
<b>A.9</b>	<b>references revisited</b>	<b>459</b>
	null reference	460
	the this reference	461
	aliases	462
	garbage collection	464
	passing objects as parameters	465
<b>A.10</b>	<b>the static modifier</b>	<b>466</b>
	static variables	466
	static methods	466
<b>A.11</b>	<b>wrapper classes</b>	<b>467</b>

<b>A.12</b>	<b>interfaces</b>	<b>468</b>
	the Comparable interface	470
	the iterator interface	471
<b>A.13</b>	<b>inheritance</b>	<b>471</b>
	derived classes	472
	the protected modifier	474
	the super reference	475
	overriding methods	476
	shadowing variables	476
<b>A.14</b>	<b>class hierarchies</b>	<b>477</b>
	the Object class	478
	abstract classes	479
	interface hierarchies	481
<b>A.15</b>	<b>polymorphism</b>	<b>481</b>
	references and class hierarchies	482
	polymorphism via inheritance	483
	polymorphism via interfaces	485
<b>A.16</b>	<b>exceptions</b>	<b>487</b>
	exception messages	487
	the try statement	488
	exception propagation	489
	the exception class hierarchy	490
<b>appendix b</b>	<b>the Java class library</b>	<b>493</b>
	AbstractButton (javax.swing)	493
	ActionEvent (java.awt.event)	495
	AdjustmentEvent (java.awt.event)	496
	Applet (java.applet)	497
	ArrayList (java.util)	499
	AWTEvent (java.awt)	500
	BigDecimal (java.math)	501
	BigInteger (java.math)	504
	BitSet (java.util)	507
	Boolean (java.lang)	508
	BorderFactory (javax.swing)	509
	BorderLayout (java.awt)	511

Box (javax.swing)	512
BoxLayout (javax.swing)	513
BufferedReader (java.io)	513
BufferedWriter (java.io)	515
ButtonGroup (javax.swing)	515
Byte (java.lang)	516
Calendar (java.util)	517
CardLayout (java.awt)	521
Character (java.lang)	523
Class (java.lang)	526
Color (java.awt)	528
Component (java.awt)	530
ComponentAdapter (java.awt.event)	535
ComponentEvent (java.awt.event)	536
Container (java.awt)	537
ContainerAdapter (java.awt.event)	539
ContainerEvent (java.awt.event)	539
Cursor (java.awt)	540
Date (java.util)	541
DateFormat (java.text)	542
DateFormatSymbols (java.text)	545
DecimalFormat (java.text)	546
DecimalFormatSymbols (java.text)	548
Dimension (java.awt)	549
Double (java.lang)	550
Error (java.lang)	552
Event (java.awt)	552
Exception (java.lang)	556
Float (java.lang)	556
FlowLayout (java.awt)	558
FocusAdapter (java.awt.event)	559
FocusEvent (java.awt.event)	559
Font (java.awt)	560
FontMetrics (java.awt)	561
Format (java.text)	563
Graphics (java.awt)	563
GregorianCalendar (java.util)	567
GridBagConstraints (java.awt)	569
GridBagLayout (java.awt)	570
GridLayout (java.awt)	573
Hashtable (java.util)	574

## xxviii contents

Image (java.awt)	575
ImageIcon (javax.swing)	576
InputEvent (java.awt.event)	577
InputStream (java.io)	578
InputStreamReader (java.io)	579
Insets (java.awt)	580
Integer (java.lang)	581
ItemEvent (java.awt.event)	583
JApplet (javax.swing)	584
JButton (javax.swing)	584
JCheckBox (javax.swing)	585
JCheckBoxMenuItem (javax.swing)	585
JColorChooser (javax.swing)	586
JComboBox (javax.swing)	586
JComponent (javax.swing)	587
JFileChooser (javax.swing)	589
JFrame (javax.swing)	590
JLabel (javax.swing)	591
JList (javax.swing)	592
JOptionPane (javax.swing)	594
JPanel (javax.swing)	595
JPasswordField (javax.swing)	596
JRadioButton (javax.swing)	596
JScrollPane (javax.swing)	597
JSlider (javax.swing)	597
JTabbedPane (javax.swing)	599
JTextArea (javax.swing)	600
TextField (javax.swing)	601
JToggleButton (javax.swing)	601
JToolTip (javax.swing)	602
KeyAdapter (java.awt.event)	602
KeyEvent (java.awt.event)	603
Locale (java.util)	607
Long (java.lang)	609
Math (java.lang)	611
MessageFormat (java.text)	613
MouseAdapter (java.awt.event)	614
MouseEvent (java.awt.event)	614
MouseMotionAdapter (java.awt.event)	615
Number (java.lang)	616
NumberFormat (java.text)	616

Object (java.lang)	619
ParsePosition (java.text)	620
Point (java.awt)	620
Polygon (java.awt)	621
PrintStream (java.io)	622
Random (java.util)	623
Rectangle (java.awt)	624
Short (java.lang)	626
SimpleDateFormat (java.text)	627
SimpleTimeZone (java.util)	628
Stack (java.util)	630
String (java.lang)	630
StringBuffer (java.lang)	634
StringTokenizer (java.util)	636
System (java.lang)	637
SystemColor (java.awt)	638
Thread (java.lang)	641
Throwable (java.lang)	644
Timer (javax.swing)	644
TimeZone (java.util)	645
URL (java.net)	646
Vector (java.util)	647
Void (java.lang)	649
Window (java.awt)	650
WindowAdapter (java.awt.event)	651
WindowEvent (java.awt.event)	652

**index****653**

