

Inhaltsverzeichnis

I	Grundlegende Konzepte	1
1	Vorbemerkungen und Überblick	3
1.1	Informatik, Algorithmen und Datenstrukturen	3
1.2	Historischer Überblick: Algorithmen	5
1.3	Historie von Programmiersprachen und Java	6
1.4	Grundkonzepte der Programmierung in Java	9
2	Algorithmische Grundkonzepte	15
2.1	Intuitiver Algorithmusbegriff	15
2.1.1	Beispiele für Algorithmen	15
2.1.2	Bausteine für Algorithmen	19
2.1.3	Pseudocode-Notation für Algorithmen	21
2.1.4	Struktogramme	26
2.1.5	Rekursion	27
2.2	Sprachen und Grammatiken	30
2.2.1	Begriffsbildung	31
2.2.2	Reguläre Ausdrücke	32
2.2.3	Backus-Naur-Form (BNF)	33
2.3	Elementare Datentypen	34
2.3.1	Datentypen als Algebren	35
2.3.2	Signaturen von Datentypen	35
2.3.3	Der Datentyp <code>bool</code>	37
2.3.4	Der Datentyp <code>integer</code>	38
2.3.5	Felder und Zeichenketten	39
2.4	Terme	41
2.4.1	Bildung von Termen	41
2.4.2	Algorithmus zur Termauswertung	43
2.5	Datentypen in Java	44
2.5.1	Primitive Datentypen	44
2.5.2	Referenzdatentypen	46
2.5.3	Operatoren	50

3	Algorithmenparadigmen	53
3.1	Überblick über Algorithmenparadigmen	53
3.2	Applikative Algorithmen	54
3.2.1	Terme mit Unbestimmten	54
3.2.2	Funktionsdefinitionen	55
3.2.3	Auswertung von Funktionen	55
3.2.4	Erweiterung der Funktionsdefinition	57
3.2.5	Applikative Algorithmen	58
3.2.6	Beispiele für applikative Algorithmen	59
3.3	Imperative Algorithmen	67
3.3.1	Grundlagen imperativer Algorithmen	67
3.3.2	Komplexe Anweisungen	70
3.3.3	Beispiele für imperative Algorithmen	73
3.4	Das logische Paradigma	79
3.4.1	Logik der Fakten und Regeln	79
3.4.2	Deduktive Algorithmen	81
3.5	Weitere Paradigmen	85
3.5.1	Genetische Algorithmen	86
3.5.2	Neuronale Netze	89
3.6	Umsetzung in Java	92
3.6.1	Ausdrücke und Anweisungen	93
3.6.2	Methoden	101
3.6.3	Applikative Algorithmen und Rekursion	107
4	Literaturhinweise zum Teil I	113
II Algorithmen		115
5	Ausgewählte Algorithmen	117
5.1	Suchen in sortierten Folgen	117
5.1.1	Sequenzielle Suche	118
5.1.2	Binäre Suche	120
5.2	Sortieren	124
5.2.1	Sortieren: Grundbegriffe	124
5.2.2	Sortieren durch Einfügen	125
5.2.3	Sortieren durch Selektion	127
5.2.4	Sortieren durch Vertauschen: BubbleSort	129
5.2.5	Sortieren durch Mischen: MergeSort	131
5.2.6	QuickSort	135
5.2.7	Sortierverfahren im Vergleich	139

6	Formale Algorithmenmodelle	143
6.1	Registermaschinen	143
6.2	Abstrakte Maschinen	152
6.3	Markov-Algorithmen	156
6.4	Church'sche These	162
6.5	Interpreter für formale Algorithmenmodelle in Java ..	164
	6.5.1 Java: Markov-Interpreter	164
	6.5.2 Registermaschine in Java	166
7	Eigenschaften von Algorithmen	173
7.1	Berechenbarkeit und Entscheidbarkeit	173
	7.1.1 Existenz nichtberechenbarer Funktionen	174
	7.1.2 Konkrete nichtberechenbare Funktionen	176
	7.1.3 Das Halteproblem	178
	7.1.4 Nichtentscheidbare Probleme	180
	7.1.5 Post'sches Korrespondenzproblem	181
7.2	Korrektheit von Algorithmen	183
	7.2.1 Relative Korrektheit	183
	7.2.2 Korrektheit von imperativen Algorithmen	184
	7.2.3 Korrektheitsbeweise für Anweisungstypen	187
	7.2.4 Korrektheit imperativer Algorithmen an Bei- spielen	189
	7.2.5 Korrektheit applikativer Algorithmen	194
7.3	Komplexität	196
	7.3.1 Motivierendes Beispiel	196
	7.3.2 Asymptotische Analyse	197
	7.3.3 Komplexitätsklassen	202
	7.3.4 Analyse von Algorithmen	204
8	Entwurf von Algorithmen	207
8.1	Entwurfsprinzipien	207
	8.1.1 Schrittweise Verfeinerung	207
	8.1.2 Einsatz von Algorithmenmustern	212
	8.1.3 Problemreduzierung durch Rekursion	213
8.2	Algorithmenmuster: Greedy	213
	8.2.1 Greedy-Algorithmen am Beispiel	214
	8.2.2 Greedy: Optimales Kommunikationsnetz	215
	8.2.3 Verfeinerung der Suche nach billigster Kante ..	217
8.3	Rekursion: Divide-and-conquer	218
	8.3.1 Das Prinzip »Teile und herrsche«	219
	8.3.2 Beispiel: Spielpläne für Turniere	220
8.4	Rekursion: Backtracking	222
	8.4.1 Prinzip des Backtracking	223

8.4.2	Beispiel: Das Acht-Damen-Problem	225
8.4.3	Beispiel: Tic Tac Toe mit Backtracking	227
8.5	Dynamische Programmierung	230
8.5.1	Das Rucksackproblem	231
8.5.2	Rekursive Lösung des Rucksackproblems	233
8.5.3	Prinzip der dynamischen Programmierung	233
9	Verteilte Berechnungen	237
9.1	Kommunizierende Prozesse	237
9.2	Modell der Petri-Netze	238
9.2.1	Definition von Petri-Netzen	238
9.2.2	Formalisierung von Petri-Netzen	242
9.2.3	Das Beispiel der fünf Philosophen	244
9.3	Programmieren nebenläufiger Abläufe	246
9.3.1	Koordinierte Prozesse	247
9.3.2	Programmieren mit Semaphoren	248
9.3.3	Philosophenproblem mit Semaphoren	250
9.3.4	Verklemmungsfreie Philosophen	252
9.4	Beispielrealisierung in Java	254
10	Literaturhinweise zum Teil II	261
III Datenstrukturen		263
11	Abstrakte Datentypen	265
11.1	Signaturen und Algebren	266
11.2	Algebraische Spezifikation	268
11.2.1	Spezifikationen und Modelle	269
11.2.2	Termalgebra und Quotiententermalgebra	270
11.2.3	Probleme mit initialer Semantik	273
11.3	Beispiele für abstrakte Datentypen	274
11.3.1	Der Kellerspeicher (Stack)	275
11.3.2	Beispiel für Kellernutzung	277
11.3.3	Die Warteschlange (Queue)	281
11.4	Entwurf von Datentypen	282
12	Klassen, Schnittstellen und Objekte in Java	285
12.1	Grundzüge der Objektorientierung	285
12.2	Klassen und Objekte in Java	288
12.3	Vererbung	293
12.4	Abstrakte Klassen und Schnittstellen	300
12.5	Ausnahmen	303
12.6	Umsetzung abstrakter Datentypen	305

12.6.1	Lambda-Ausdrücke in Java 8	309
13	Grundlegende Datenstrukturen	315
13.1	Stack und Queue als Datentypen	315
13.1.1	Implementierung des Stacks	319
13.1.2	Implementierung der Queue	320
13.1.3	Bewertung der Implementierungen	322
13.2	Verkettete Listen	323
13.3	Doppelt verkettete Listen	330
13.4	Das Iterator-Konzept	335
13.5	Java Collection Framework	338
13.6	J2SE 5.0 und Generics	342
14	Bäume	345
14.1	Bäume: Begriffe und Konzepte	345
14.2	Binärer Baum: Datentyp und Basisalgorithmen	348
14.2.1	Der Datentyp »Binärer Baum«	348
14.2.2	Algorithmen zur Traversierung	353
14.3	Suchbäume	358
14.3.1	Suchen in Suchbäumen	359
14.3.2	Einfügen und Löschen	362
14.3.3	Komplexität der Operationen	367
14.4	Ausgeglichene Bäume	368
14.4.1	Rot-Schwarz-Bäume	369
14.4.2	AVL-Bäume	378
14.4.3	B-Bäume	386
14.5	Digitale Bäume	399
14.5.1	Tries	400
14.5.2	Patricia-Bäume	406
14.6	Praktische Nutzung von Bäumen	407
14.6.1	Sortieren mit Bäumen: HeapSort	407
14.6.2	Sets mit binären Suchbäumen	413
15	Hashverfahren	419
15.1	Grundprinzip des Hashens	419
15.2	Grundlagen und Verfahren	420
15.2.1	Hashfunktionen	420
15.2.2	Behandlung von Kollisionen	422
15.2.3	Aufwand beim Hashen	426
15.2.4	Hashen in Java	428
15.3	Dynamische Hashverfahren	432
15.3.1	Grundideen für dynamische Hashverfahren	433
15.3.2	Erweiterbares Hashen	436

	15.3.3 Umsetzung des erweiterbaren Hashens	439
16	Graphen	445
16.1	Arten von Graphen	445
	16.1.1 Ungerichtete Graphen	446
	16.1.2 Gerichtete Graphen	447
	16.1.3 Gewichtete Graphen	448
16.2	Realisierung von Graphen	449
	16.2.1 Knoten- und Kantenlisten	449
	16.2.2 Adjazenzmatrix	450
	16.2.3 Graphen als dynamische Datenstrukturen	450
	16.2.4 Transformationen zwischen Darstellungen	451
	16.2.5 Vergleich der Komplexität	452
	16.2.6 Eine Java-Klasse für Graphen	452
16.3	Ausgewählte Graphenalgorithmen	455
	16.3.1 Breitendurchlauf	455
	16.3.2 Tiefendurchlauf	459
	16.3.3 Zyklenfreiheit und topologisches Sortieren	463
16.4	Algorithmen auf gewichteten Graphen	466
	16.4.1 Kürzeste Wege	467
	16.4.2 Dijkstras Algorithmus	468
	16.4.3 A*-Algorithmus	471
	16.4.4 Kürzeste Wege mit negativen Kantengewichten	478
	16.4.5 Maximaler Durchfluss	481
	16.4.6 Der Ford-Fulkerson-Algorithmus	483
16.5	Weitere Fragestellungen für Graphen	487
17	Algorithmen auf Texten	491
17.1	Probleme der Worterkennung	491
17.2	Knuth-Morris-Pratt	493
17.3	Boyer-Moore	497
17.4	Pattern Matching	503
	17.4.1 Reguläre Ausdrücke	503
	17.4.2 Endliche Automaten	504
	17.4.3 Java-Klassen für reguläre Ausdrücke	510
17.5	Ähnlichkeit von Zeichenketten	512
	17.5.1 Levenshtein-Distanz	512
	17.5.2 n-Gramme	515
	17.5.3 Zusammenfassung	517
18	Literaturhinweise zum Teil III	519

IV	Anhang	521
A	Quelltext der Klasse <code>IOUtils</code>	523
	Abbildungsverzeichnis	527
	Tabellenverzeichnis	533
	Algorithmenverzeichnis	535
	Beispielverzeichnis	537
	Programmverzeichnis	539
	Literaturverzeichnis	543
	Index	547