

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	26
2.2	Sprachen und Grammatiken	29
2.2.1	Begriffsbildung	30
2.2.2	Reguläre Ausdrücke	31
2.2.3	Backus-Naur-Form (BNF)	32
2.3	Elementare Datentypen	34
2.3.1	Datentypen als Algebren	34
2.3.2	Signaturen von Datentypen	35
2.3.3	Der Datentyp <code>bool</code>	36
2.3.4	Der Datentyp <code>integer</code>	37
2.3.5	Felder und Zeichenketten	38
2.4	Terme	40
2.4.1	Bildung von Termen	40
2.4.2	Algorithmus zur Termauswertung	42
2.5	Datentypen in Java	43
2.5.1	Primitive Datentypen	43
2.5.2	Referenzdatentypen	46
2.5.3	Operatoren	49

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	100
3.6.3	Applikative Algorithmen und Rekursion	106
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 Beispielen .	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	201
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	210
8.1.3	Problemreduzierung durch Rekursion	210
8.2	Algorithmenmuster: Greedy	211
8.2.1	Greedy-Algorithmen am Beispiel	211
8.2.2	Greedy: Optimales Kommunikationsnetz	213
8.2.3	Verfeinerung der Suche nach billigster Kante	214
8.3	Rekursion: Divide-and-conquer	216
8.3.1	Das Prinzip »Teile und herrsche«	217
8.3.2	Beispiel: Spielpläne für Turniere	218
8.4	Rekursion: Backtracking	220
8.4.1	Prinzip des Backtracking	221
8.4.2	Beispiel: Das Acht-Damen-Problem	223

8.5	Dynamische Programmierung	225
8.5.1	Das Rucksackproblem	226
8.5.2	Rekursive Lösung des Rucksackproblems	227
8.5.3	Prinzip der dynamischen Programmierung	228
9	Verteilte Berechnungen	231
9.1	Kommunizierende Prozesse	231
9.2	Modell der Petri-Netze	232
9.2.1	Definition von Petri-Netzen	232
9.2.2	Formalisierung von Petri-Netzen	236
9.2.3	Das Beispiel der fünf Philosophen	238
9.3	Programmieren nebenläufiger Abläufe	240
9.3.1	Koordinierte Prozesse	241
9.3.2	Programmieren mit Semaphoren	242
9.3.3	Philosophenproblem mit Semaphoren	244
9.3.4	Verklebungsfreie Philosophen	246
9.4	Beispielrealisierung in Java	248
10	Literaturhinweise zum Teil II	255
III	Datenstrukturen	257
11	Abstrakte Datentypen	259
11.1	Signaturen und Algebren	260
11.2	Algebraische Spezifikation	262
11.2.1	Spezifikationen und Modelle	263
11.2.2	Termalgebra und Quotiententermalgebra	264
11.2.3	Probleme mit initialer Semantik	267
11.3	Beispiele für abstrakte Datentypen	268
11.3.1	Der Kellerspeicher (Stack)	269
11.3.2	Beispiel für Kellernutzung	271
11.3.3	Die Warteschlange (Queue)	275
11.4	Entwurf von Datentypen	276
12	Klassen, Schnittstellen und Objekte in Java	279
12.1	Grundzüge der Objektorientierung	279
12.2	Klassen und Objekte in Java	282
12.3	Vererbung	287
12.4	Abstrakte Klassen und Schnittstellen	294
12.5	Ausnahmen	297
12.6	Umsetzung abstrakter Datentypen	299

13	Grundlegende Datenstrukturen	305
13.1	Stack und Queue als Datentypen	305
13.1.1	Implementierung des Stacks	309
13.1.2	Implementierung der Queue	310
13.1.3	Bewertung der Implementierungen	312
13.2	Verkettete Listen	313
13.3	Doppelt verkettete Listen	320
13.4	Das Iterator-Konzept	325
13.5	Java Collection Framework	328
13.6	J2SE 5.0 und Generics	332
14	Bäume	335
14.1	Bäume: Begriffe und Konzepte	335
14.2	Binärer Baum: Datentyp und Basisalgorithmen	338
14.2.1	Der Datentyp »Binärer Baum«	338
14.2.2	Algorithmen zur Traversierung	343
14.3	Suchbäume	348
14.3.1	Suchen in Suchbäumen	349
14.3.2	Einfügen und Löschen	352
14.3.3	Komplexität der Operationen	357
14.4	Ausgeglichene Bäume	358
14.4.1	Rot-Schwarz-Bäume	359
14.4.2	AVL-Bäume	368
14.4.3	B-Bäume	376
14.5	Digitale Bäume	389
14.5.1	Tries	390
14.5.2	Patricia-Bäume	396
14.6	Praktische Nutzung von Bäumen	397
14.6.1	Sortieren mit Bäumen: HeapSort	397
14.6.2	Sets mit binären Suchbäumen	403
15	Hashverfahren	409
15.1	Grundprinzip des Hashens	409
15.2	Grundlagen und Verfahren	410
15.2.1	Hashfunktionen	410
15.2.2	Behandlung von Kollisionen	412
15.2.3	Aufwand beim Hashen	416
15.2.4	Hashen in Java	418
15.3	Dynamische Hashverfahren	422
15.3.1	Grundideen für dynamische Hashverfahren	423
15.3.2	Erweiterbares Hashen	426
15.3.3	Umsetzung des erweiterbaren Hashens	428

16	Graphen	433
16.1	Arten von Graphen	433
16.1.1	Ungerichtete Graphen	434
16.1.2	Gerichtete Graphen	435
16.1.3	Gewichtete Graphen	436
16.2	Realisierung von Graphen	437
16.2.1	Knoten- und Kantenlisten	437
16.2.2	Adjazenzmatrix	438
16.2.3	Graphen als dynamische Datenstrukturen	438
16.2.4	Transformationen zwischen Darstellungen	439
16.2.5	Vergleich der Komplexität	440
16.2.6	Eine Java-Klasse für Graphen	440
16.3	Ausgewählte Graphenalgorithmen	443
16.3.1	Breitendurchlauf	443
16.3.2	Tiefendurchlauf	447
16.3.3	Zyklenfreiheit und topologisches Sortieren	451
16.4	Algorithmen auf gewichteten Graphen	454
16.4.1	Kürzeste Wege	455
16.4.2	Dijkstras Algorithmus	456
16.4.3	A*-Algorithmus	459
16.4.4	Kürzeste Wege mit negativen Kantengewichten ...	466
16.4.5	Maximaler Durchfluss	469
16.4.6	Der Ford-Fulkerson-Algorithmus	471
16.5	Weitere Fragestellungen für Graphen	475
17	Suchen in Texten	479
17.1	Probleme der Worterkennung	479
17.2	Knuth-Morris-Pratt	481
17.3	Boyer-Moore	485
17.4	Pattern Matching	491
17.4.1	Reguläre Ausdrücke	491
17.4.2	Endliche Automaten	492
17.4.3	Java-Klassen für reguläre Ausdrücke	498
17.4.4	Ähnlichkeit von Zeichenketten: Die Levenshtein-Distanz	499
18	Literaturhinweise zum Teil III	503
A	Quelltext der Klasse IOUtils	505
	Abbildungsverzeichnis	509
	Tabellenverzeichnis	515

Algorithmenverzeichnis	517
Beispielverzeichnis	519
Programmverzeichnis	521
Literaturverzeichnis	523
Index	527