

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	17
2.1	Intuitiver Algorithmusbegriff	17
2.1.1	Beispiele für Algorithmen	17
2.1.2	Bausteine für Algorithmen	22
2.1.3	Pseudocode-Notation für Algorithmen	23
2.1.4	Struktogramme	28
2.1.5	Rekursion	28
2.2	Sprachen und Grammatiken	31
2.2.1	Begriffsbildung	33
2.2.2	Reguläre Ausdrücke	33
2.2.3	Backus-Naur-Form (BNF)	34
2.3	Elementare Datentypen	36
2.3.1	Datentypen als Algebren	36
2.3.2	Signaturen von Datentypen	37
2.3.3	Der Datentyp <code>bool</code>	38
2.3.4	Der Datentyp <code>integer</code>	39
2.3.5	Felder und Zeichenketten	41
2.4	Terme	42
2.4.1	Bildung von Termen	42
2.4.2	Algorithmus zur Termauswertung	45
2.5	Datentypen in Java	45
2.5.1	Primitive Datentypen	46
2.5.2	Referenzdatentypen	48
2.5.3	Operatoren	52

3	Algorithmenparadigmen	55
3.1	Überblick über Algorithmenparadigmen	55
3.2	Applikative Algorithmen	56
3.2.1	Terme mit Unbestimmten	56
3.2.2	Funktionsdefinitionen	57
3.2.3	Auswertung von Funktionen	58
3.2.4	Erweiterung der Funktionsdefinition	59
3.2.5	Applikative Algorithmen	60
3.2.6	Beispiele für applikative Algorithmen	61
3.3	Imperative Algorithmen	68
3.3.1	Grundlagen imperativer Algorithmen	69
3.3.2	Komplexe Anweisungen	71
3.3.3	Beispiele für imperative Algorithmen	74
3.4	Das logische Paradigma	80
3.4.1	Logik der Fakten und Regeln	80
3.4.2	Deduktive Algorithmen	82
3.5	Umsetzung in Java	86
3.5.1	Ausdrücke und Anweisungen	87
3.5.2	Methoden	94
3.5.3	Applikative Algorithmen und Rekursion	98
4	Literaturhinweise zum Teil I	105
II	Algorithmen	107
5	Ausgewählte Algorithmen	109
5.1	Suchen in sortierten Folgen	109
5.1.1	Sequenzielle Suche	110
5.1.2	Binäre Suche	112
5.2	Sortieren	116
5.2.1	Sortieren: Grundbegriffe	116
5.2.2	Sortieren durch Einfügen	117
5.2.3	Sortieren durch Selektion	119
5.2.4	Sortieren durch Vertauschen: BubbleSort	121
5.2.5	Sortieren durch Mischen: MergeSort	123
5.2.6	QuickSort	127
5.2.7	Sortierverfahren im Vergleich	131
6	Formale Algorithmenmodelle	135
6.1	Registermaschinen	136
6.2	Abstrakte Maschinen	143
6.3	Markov-Algorithmen	148
6.4	Church'sche These	153
6.5	Interpreter für formale Algorithmenmodelle in Java	155

6.5.1	Java: Markov-Interpreter	156
6.5.2	Registermaschine in Java	158
7	Eigenschaften von Algorithmen	165
7.1	Berechenbarkeit und Entscheidbarkeit	165
7.1.1	Existenz nichtberechenbarer Funktionen	166
7.1.2	Konkrete nichtberechenbare Funktionen	168
7.1.3	Das Halteproblem	170
7.1.4	Nichtentscheidbare Probleme	172
7.1.5	Post'sches Korrespondenzproblem	173
7.2	Korrektheit von Algorithmen	175
7.2.1	Relative Korrektheit	176
7.2.2	Korrektheit von imperativen Algorithmen	176
7.2.3	Korrektheitsbeweise für Anweisungstypen	179
7.2.4	Korrektheit imperativer Algorithmen an Beispielen	182
7.2.5	Korrektheit applikativer Algorithmen	187
7.3	Komplexität	188
7.3.1	Motivierendes Beispiel	189
7.3.2	Asymptotische Analyse	190
7.3.3	Komplexitätsklassen	194
7.3.4	Analyse von Algorithmen	197
8	Entwurf von Algorithmen	201
8.1	Entwurfsprinzipien	201
8.1.1	Schrittweise Verfeinerung	201
8.1.2	Einsatz von Algorithmenmustern	202
8.1.3	Problemreduzierung durch Rekursion	202
8.2	Algorithmenmuster: Greedy	203
8.2.1	Greedy-Algorithmen am Beispiel	204
8.2.2	Greedy: Optimales Kommunikationsnetz	205
8.2.3	Verfeinerung der Suche nach billigster Kante	206
8.3	Rekursion: Divide-and-conquer	209
8.3.1	Das Prinzip »Teile und herrsche«	209
8.3.2	Beispiel: Spielpläne für Turniere	210
8.4	Rekursion: Backtracking	212
8.4.1	Prinzip des Backtracking	213
8.4.2	Beispiel: Das Acht-Damen-Problem	215
8.5	Dynamische Programmierung	218
8.5.1	Das Rucksackproblem	218
8.5.2	Rekursive Lösung des Rucksackproblems	220
8.5.3	Prinzip der dynamischen Programmierung	220

9	Verteilte Berechnungen	223
9.1	Kommunizierende Prozesse	223
9.2	Modell der Petri-Netze	224
9.2.1	Definition von Petri-Netzen	224
9.2.2	Formalisierung von Petri-Netzen	228
9.2.3	Das Beispiel der fünf Philosophen	230
9.3	Programmieren nebenläufiger Abläufe	233
9.3.1	Koordinierte Prozesse	233
9.3.2	Programmieren mit Semaphoren	234
9.3.3	Philosophenproblem mit Semaphoren	237
9.3.4	Verklemmungsfreie Philosophen	238
9.4	Beispielrealisierung in Java	240
10	Literaturhinweise zum Teil II	247
III	Datenstrukturen	249
11	Abstrakte Datentypen	251
11.1	Signaturen und Algebren	252
11.2	Algebraische Spezifikation	254
11.2.1	Spezifikationen und Modelle	255
11.2.2	Termalgebra und Quotiententermalgebra	256
11.2.3	Probleme mit initialer Semantik	259
11.3	Beispiele für abstrakte Datentypen	260
11.3.1	Der Kellerspeicher (Stack)	261
11.3.2	Beispiel für Kellernutzung	263
11.3.3	Die Warteschlange (Queue)	268
11.4	Entwurf von Datentypen	269
12	Klassen, Schnittstellen und Objekte in Java	273
12.1	Grundzüge der Objektorientierung	273
12.2	Klassen und Objekte in Java	276
12.3	Vererbung	280
12.4	Abstrakte Klassen und Schnittstellen	288
12.5	Ausnahmen	291
12.6	Umsetzung abstrakter Datentypen	293
13	Grundlegende Datenstrukturen	297
13.1	Stack und Queue als Datentypen	297
13.1.1	Implementierung des Stacks	301
13.1.2	Implementierung der Queue	303
13.1.3	Bewertung der Implementierungen	305
13.2	Verkettete Listen	305
13.3	Doppelt verkettete Listen	313

13.4	Das Iterator-Konzept	317
13.5	Java Collection Framework	320
14	Bäume	325
14.1	Bäume: Begriffe und Konzepte	325
14.2	Binärer Baum: Datentyp und Basisalgorithmen	328
14.2.1	Der Datentyp »Binärer Baum«	328
14.2.2	Algorithmen zur Traversierung	333
14.3	Suchbäume	338
14.3.1	Suchen in Suchbäumen	339
14.3.2	Einfügen und Löschen	342
14.3.3	Komplexität der Operationen	349
14.4	Ausgeglichene Bäume	350
14.4.1	AVL-Bäume	351
14.4.2	B-Bäume	358
14.4.3	Prinzip des B-Baumes	358
14.4.4	Suchen in B-Bäumen	360
14.4.5	Einfügen und Löschen	361
14.4.6	Komplexität der Operationen	365
14.5	Digitale Bäume	366
14.5.1	Tries	366
14.5.2	Patricia-Bäume	368
14.6	Praktische Nutzung von Bäumen	369
14.6.1	Sortieren mit Bäumen: HeapSort	369
14.6.2	Sets mit binären Suchbäumen	376
15	Hashverfahren	381
15.1	Grundprinzip des Hashens	381
15.2	Grundlagen und Verfahren	382
15.2.1	Hashfunktionen	383
15.2.2	Behandlung von Kollisionen	384
15.2.3	Aufwand beim Hashen	388
15.2.4	Hashen in Java	389
15.3	Dynamische Hashverfahren	393
15.3.1	Grundideen für dynamische Hashverfahren	394
15.3.2	Erweiterbares Hashen	397
16	Graphen	401
16.1	Arten von Graphen	401
16.1.1	Ungerichtete Graphen	402
16.1.2	Gerichtete Graphen	403
16.1.3	Gewichtete Graphen	404
16.2	Realisierung von Graphen	405
16.2.1	Knoten- und Kantenlisten	405

16.2.2	Adjazenzmatrix	406
16.2.3	Graphen als dynamische Datenstrukturen	406
16.2.4	Transformationen zwischen Darstellungen	407
16.2.5	Vergleich der Komplexität	408
16.2.6	Eine Java-Klasse für Graphen	408
16.3	Ausgewählte Graphenalgorithmen	411
16.3.1	Breitendurchlauf	411
16.3.2	Tiefendurchlauf	415
16.3.3	Zyklenfreiheit und topologisches Sortieren	419
16.4	Algorithmen auf gewichteten Graphen	422
16.4.1	Kürzeste Wege	423
16.4.2	Dijkstras Algorithmus	423
16.4.3	Kürzeste Wege mit negativen Kantengewichten	427
16.4.4	Maximaler Durchfluss	431
16.4.5	Der Ford-Fulkerson-Algorithmus	433
16.5	Weitere Fragestellungen für Graphen	437
17	Suchen in Texten	441
17.1	Probleme der Worterkennung	441
17.2	Knuth-Morris-Pratt	443
17.3	Boyer-Moore	447
17.4	Pattern Matching	453
17.4.1	Reguläre Ausdrücke	454
17.4.2	Endliche Automaten	455
18	Literaturhinweise zum Teil III	463
A	Quelltext der Klasse IOUtils	465
	Abbildungsverzeichnis	469
	Tabellenverzeichnis	475
	Algorithmenverzeichnis	477
	Beispielverzeichnis	479
	Programmverzeichnis	481
	Literaturverzeichnis	483
	Index	487