

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	8
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	45
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	105
4	Literaturhinweise zum Teil I	111

II Algorithmen **113**

5	Ausgewählte Algorithmen	115
5.1	Suchen in sortierten Folgen	115
5.1.1	Sequenzielle Suche	116
5.1.2	Binäre Suche	118
5.2	Sortieren	122
5.2.1	Sortieren: Grundbegriffe	122
5.2.2	Sortieren durch Einfügen	123
5.2.3	Sortieren durch Selektion	125
5.2.4	Sortieren durch Vertauschen: BubbleSort	127
5.2.5	Sortieren durch Mischen: MergeSort	129
5.2.6	QuickSort	133
5.2.7	Sortierverfahren im Vergleich	137

6	Formale Algorithmenmodelle	141
6.1	Registermaschinen	141
6.2	Abstrakte Maschinen	150
6.3	Markov-Algorithmen	154
6.4	Church'sche These	160
6.5	Interpreter für formale Algorithmenmodelle in Java	162
6.5.1	Java: Markov-Interpreter	162
6.5.2	Registermaschine in Java	164
7	Eigenschaften von Algorithmen	171
7.1	Berechenbarkeit und Entscheidbarkeit	171
7.1.1	Existenz nichtberechenbarer Funktionen	172
7.1.2	Konkrete nichtberechenbare Funktionen	174
7.1.3	Das Halteproblem	176
7.1.4	Nichtentscheidbare Probleme	178
7.1.5	Post'sches Korrespondenzproblem	179
7.2	Korrektheit von Algorithmen	181
7.2.1	Relative Korrektheit	181
7.2.2	Korrektheit von imperativen Algorithmen	182
7.2.3	Korrektheitsbeweise für Anweisungstypen	185
7.2.4	Korrektheit imperativer Algorithmen an Beispielen	187
7.2.5	Korrektheit applikativer Algorithmen	192
7.3	Komplexität	194
7.3.1	Motivierendes Beispiel	194
7.3.2	Asymptotische Analyse	195
7.3.3	Komplexitätsklassen	199
7.3.4	Analyse von Algorithmen	202
8	Entwurf von Algorithmen	205
8.1	Entwurfsprinzipien	205
8.1.1	Schrittweise Verfeinerung	205
8.1.2	Einsatz von Algorithmenmustern	206
8.1.3	Problemreduzierung durch Rekursion	206
8.2	Algorithmenmuster: Greedy	207
8.2.1	Greedy-Algorithmen am Beispiel	207
8.2.2	Greedy: Optimales Kommunikationsnetz	209
8.2.3	Verfeinerung der Suche nach billigster Kante	210
8.3	Rekursion: Divide-and-conquer	212
8.3.1	Das Prinzip »Teile und herrsche«	212
8.3.2	Beispiel: Spielpläne für Turniere	214
8.4	Rekursion: Backtracking	216
8.4.1	Prinzip des Backtracking	217
8.4.2	Beispiel: Das Acht-Damen-Problem	218

8.5	Dynamische Programmierung	221
8.5.1	Das Rucksackproblem	222
8.5.2	Rekursive Lösung des Rucksackproblems	223
8.5.3	Prinzip der dynamischen Programmierung	224
9	Verteilte Berechnungen	227
9.1	Kommunizierende Prozesse	227
9.2	Modell der Petri-Netze	228
9.2.1	Definition von Petri-Netzen	228
9.2.2	Formalisierung von Petri-Netzen	232
9.2.3	Das Beispiel der fünf Philosophen	234
9.3	Programmieren nebenläufiger Abläufe	236
9.3.1	Koordinierte Prozesse	237
9.3.2	Programmieren mit Semaphoren	238
9.3.3	Philosophenproblem mit Semaphoren	240
9.3.4	Verklebungsfreie Philosophen	242
9.4	Beispielrealisierung in Java	244
10	Literaturhinweise zum Teil II	251

III Datenstrukturen 253

11	Abstrakte Datentypen	255
11.1	Signaturen und Algebren	256
11.2	Algebraische Spezifikation	258
11.2.1	Spezifikationen und Modelle	259
11.2.2	Termalgebra und Quotiententermalgebra	260
11.2.3	Probleme mit initialer Semantik	263
11.3	Beispiele für abstrakte Datentypen	264
11.3.1	Der Kellerspeicher (Stack)	265
11.3.2	Beispiel für Kellernutzung	267
11.3.3	Die Warteschlange (Queue)	271
11.4	Entwurf von Datentypen	272
12	Klassen, Schnittstellen und Objekte in Java	275
12.1	Grundzüge der Objektorientierung	275
12.2	Klassen und Objekte in Java	278
12.3	Vererbung	283
12.4	Abstrakte Klassen und Schnittstellen	290
12.5	Ausnahmen	293
12.6	Umsetzung abstrakter Datentypen	295

13	Grundlegende Datenstrukturen	299
13.1	Stack und Queue als Datentypen	299
13.1.1	Implementierung des Stacks	303
13.1.2	Implementierung der Queue	304
13.1.3	Bewertung der Implementierungen	306
13.2	Verkettete Listen	307
13.3	Doppelt verkettete Listen	314
13.4	Das Iterator-Konzept	319
13.5	Java Collection Framework	322
13.6	J2SE 5.0 und Generics	326
14	Bäume	329
14.1	Bäume: Begriffe und Konzepte	329
14.2	Binärer Baum: Datentyp und Basisalgorithmen	332
14.2.1	Der Datentyp »Binärer Baum«	332
14.2.2	Algorithmen zur Traversierung	337
14.3	Suchbäume	342
14.3.1	Suchen in Suchbäumen	343
14.3.2	Einfügen und Löschen	346
14.3.3	Komplexität der Operationen	351
14.4	Ausgeglichene Bäume	352
14.4.1	Rot-Schwarz-Bäume	353
14.4.2	AVL-Bäume	362
14.4.3	B-Bäume	370
14.5	Digitale Bäume	377
14.5.1	Tries	377
14.5.2	Patricia-Bäume	383
14.6	Praktische Nutzung von Bäumen	384
14.6.1	Sortieren mit Bäumen: HeapSort	385
14.6.2	Sets mit binären Suchbäumen	391
15	Hashverfahren	397
15.1	Grundprinzip des Hashens	397
15.2	Grundlagen und Verfahren	398
15.2.1	Hashfunktionen	398
15.2.2	Behandlung von Kollisionen	400
15.2.3	Aufwand beim Hashen	404
15.2.4	Hashen in Java	406
15.3	Dynamische Hashverfahren	410
15.3.1	Grundideen für dynamische Hashverfahren	411
15.3.2	Erweiterbares Hashen	414
15.3.3	Umsetzung des erweiterbaren Hashens	416

16	Graphen	421
16.1	Arten von Graphen	421
16.1.1	Ungerichtete Graphen	422
16.1.2	Gerichtete Graphen	423
16.1.3	Gewichtete Graphen	424
16.2	Realisierung von Graphen	425
16.2.1	Knoten- und Kantenlisten	425
16.2.2	Adjazenzmatrix	426
16.2.3	Graphen als dynamische Datenstrukturen	426
16.2.4	Transformationen zwischen Darstellungen	427
16.2.5	Vergleich der Komplexität	428
16.2.6	Eine Java-Klasse für Graphen	428
16.3	Ausgewählte Graphenalgorithmen	430
16.3.1	Breitendurchlauf	431
16.3.2	Tiefendurchlauf	435
16.3.3	Zyklenfreiheit und topologisches Sortieren	439
16.4	Algorithmen auf gewichteten Graphen	441
16.4.1	Kürzeste Wege	442
16.4.2	Dijkstras Algorithmus	443
16.4.3	Kürzeste Wege mit negativen Kantengewichten ...	447
16.4.4	Maximaler Durchfluss	450
16.4.5	Der Ford-Fulkerson-Algorithmus	452
16.5	Weitere Fragestellungen für Graphen	456
17	Suchen in Texten	459
17.1	Probleme der Worterkennung	459
17.2	Knuth-Morris-Pratt	461
17.3	Boyer-Moore	465
17.4	Pattern Matching	471
17.4.1	Reguläre Ausdrücke	471
17.4.2	Endliche Automaten	472
17.4.3	Java-Klassen für reguläre Ausdrücke	478
18	Literaturhinweise zum Teil III	481
A	Quelltext der Klasse IOUtils	483
	Abbildungsverzeichnis	487
	Tabellenverzeichnis	493
	Algorithmenverzeichnis	495
	Beispielverzeichnis	497

Programmverzeichnis	499
Literaturverzeichnis	501
Index	505