

eXamen.press

eXamen.press ist eine Reihe, die Theorie und Praxis aus allen Bereichen der Informatik für die Hochschulausbildung vermittelt.

Wolfgang Kuchlin · Andreas Weber

Einführung in die Informatik

Objektorientiert mit Java

3., überarbeitete Auflage
mit 48 Abbildungen und 4 Tabellen

 Springer

Prof. Dr. Wolfgang Kuchlin
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen
Sand 14, 72076 Tübingen
kuechlin@informatik.uni-tuebingen.de
<http://www-sr.informatik.uni-tuebingen.de>

Prof. Dr. Andreas Weber
Institut für Informatik II
Universität Bonn
Römerstr. 164
53117 Bonn
weber@cs.uni-bonn.de

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen
Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über
<http://dnb.ddb.de> abrufbar.

ACM Computing Classification (1998): A.1, D.1-3, F.2-3

ISSN 1614-5216
ISBN 3-540-20958-1 Springer Berlin Heidelberg New York
ISBN 3-540-43608-1 2. Auflage Springer Berlin Heidelberg

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Springer ist ein Unternehmen von Springer Science+Business Media
springer.de

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutzgesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. Text und Abbildungen wurden mit größter Sorgfalt erarbeitet. Verlag und Autor können jedoch für eventuell verbliebene fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Satz: Druckfertige Daten der Autoren
Herstellung: LE-TeX Jelonek, Schmidt & Vöckler GbR, Leipzig
Umschlaggestaltung: KünkelLopka Werbeagentur, Heidelberg
Gedruckt auf säurefreiem Papier 33/3142/YL - 5 4 3 2 1 0

Unseren Familien gewidmet
WK, AW

Vorwort

The Analytical Engine is therefore a machine of the most general nature.

Charles Babbage (1864)

Dies ist ein einführendes Lehrbuch der Informatik. Es umfaßt den Stoff, der typischerweise im ersten Studienjahr an Universitäten in 1–2 Vorlesungen wie „Informatik I“ und „Informatik II“ gelehrt wird. Es geht zentral um Grundkonzepte von (objektorientierten) Programmiersprachen und von Algorithmen, und zur ihrer Umsetzung wird durchgehend Modellierung mit der Sprache UML und Programmierung mit der Sprache Java eingesetzt. Dieses Buch ist also weder ein Java-Handbuch noch ein Schnellkurs im Programmieren. Der zentrale Stoff wird ergänzt durch eine Übersicht über die Architektur eines Computers am Beispiel eines modernen PC-Systems und eine Einführung in die mathematisch-logischen Grundlagen der Informatik. Unter der URL

www-sr.informatik.uni-tuebingen.de/InfoBuch

ist eine Web-Seite zu diesem Buch eingerichtet. Dort sind u. a. Foliensätze für Dozenten zu finden.

Heute ist der Einsatz von Computern nicht mehr auf das klassische Gebiet des technisch-wissenschaftlichen Rechnens konzentriert, sondern dringt auf breiter Front in alle Bereiche von Wissenschaft, Wirtschaft und Gesellschaft vor. Die neuen Einsatzgebiete, wie z. B. moderne Client-Server-Informationssysteme, verlangen in der Praxis neue Werkzeuge und Methoden, auch wenn die alten rein theoretisch noch genügen würden.

Objektorientierte Software-Entwicklung mit der Modellierungssprache UML, der Programmiersprache Java und neuen integrierten Programmierwerkzeugen wie Eclipse sind die wichtigsten neuen Hilfsmittel, mit denen man der Herausforderung immer vielseitigerer, vielschichtigerer und vernetzter Software begegnet. Objektorientierte Methoden haben in der Praxis wesentlich dazu beigetragen, den Komplexitätsschub in Entwurf, Programmierung und Wartung moderner Systeme in den Griff zu bekommen. Java, als Programmiersprache des Internet bekannt geworden, ist eine moderne objektorientierte Sprache, die sowohl durch klare theoretische Konzepte besticht als auch in der breiten Praxis – von Mobiltelefonen bis zu Großrechnern – Anwendung findet.

Der zentrale Beweggrund für dieses Lehrbuch der Informatik war das Erreichen einer Balance zwischen Theorie und Praxis, also zwischen theoretisch-konzeptuellen Grundlagen und unmittelbar praxisrelevantem Wissen. Dieses Lehr-

buch soll die traditionellen Konzepte, die in der Einführungsvorlesung Informatik im ersten Studienjahr gelehrt werden, um den Gesichtspunkt der Objektorientierung ergänzen und aus dieser Sicht neu präsentieren sowie anhand von ausgewählten Teilen von UML und Java einüben.

Der Leser soll insbesondere

- grundlegende Konzepte der objektorientierten Software-Entwicklung und des Programmierens verstehen,
- mit Java eine moderne objektorientierte Sprache erlernen, die diese Konzepte umsetzt und die auch in der breiten Praxis von Wissenschaft und Wirtschaft in allen Anwendungsgebieten und bei großen komplexen Aufgaben verwendet wird,
- höhere Datenstrukturen, Algorithmen und deren zugrundeliegende Entwurfsmuster anhand des Standardrepertoires der Informatik kennenlernen und
- ein zukunftsfestes Wissen der theoretischen Grundlagen der praktischen Informatik erwerben, um eine Basis für lebenslanges Lernen zu erhalten.

Diese Neuauflage schließt sich an die Vorlesungen Informatik I in Bonn im WS 2002/2003 und Informatik I und II in Tübingen im akademischen Jahr 2003/2004 an. In Tübingen haben wir zum ersten Mal von Anfang an die neue integrierte Entwicklungsumgebung Eclipse eingesetzt.¹ Eine für Anfänger nützliche Einführung in Eclipse hätte den Umfang dieses Buchs gesprengt, aber einige Hinweise konnten in den Text aufgenommen werden.

Neben zahlreichen weiteren Detailverbesserungen verdienen die folgenden Punkte besondere Erwähnung:

- Die Reihenfolge der Kapitel in Teil I wurde verändert und ist jetzt: 2. Rechnerarchitektur – 3. Algorithmen – 4. Datenstrukturen – 5. Objektorientierung. Die neue Anordnung folgt einem *bottom-up* Prinzip und läßt sich gerade zu Beginn schlüssiger lehren.
- Kapitel 2 (Rechnerarchitektur) wurde überarbeitet. Die Behandlung von Zahldarstellungen und Konversionsmethoden wurde ergänzt, insbesondere auch im Teil zu Gleitkommazahlen und dient jetzt auch als natürliche Motivation für Algorithmen im nachfolgenden Kapitel 3.
- Die Flußdiagramme in Kapitel 3 (Algorithmen) wurden auf die Notation von UML Aktivitätsdiagrammen umgestellt. Da UML auch von Eclipse unterstützt wird, lassen sich jeweils das Diagramm und der Java-Code nebeneinander in Eclipse betrachten.
- Viele der UML Klassendiagramme wurden neu gezeichnet und dabei einige Abweichungen vom UML Standard bereinigt.
- Es wurden zahlreiche weitere in den oben genannten Veranstaltungen erprobte Übungsaufgaben aufgenommen.

¹ Dieses Konzept mit dem Namen FOOD (*foundations of object oriented development*) wurde 2003 mit einem *IBM Eclipse Innovation Award* ausgezeichnet (siehe www-sr.informatik.uni-tuebingen.de/food).

Danksagung. Wir danken wiederum allen unseren Mitarbeitern, insbesondere denen, die unsere Vorlesungen betreut haben. In Tübingen sind dies Dr. W. Blochinger, W. Westje und Dr. V. Simonis; Frau E.-M. Dieringer hat wiederum zahlreiche neue Abbildungen angefertigt. In Bonn möchten wir insbesondere G. Sobottka, M. Guthe, D. Goldbach und Frau M. Gnasa für Korrektur- und Ergänzungsvorschläge danken.

Besonders zu Dank verpflichtet sind wir unseren Lesern und denjenigen Kollegen an zahlreichen Hochschulen, die die bisherigen Auflagen für Ihre Vorlesungen verwendet haben; sie haben die neue Auflage erst möglich gemacht. Wertvolle Hinweise verdanken wir Herrn Prof. Dr. B. Eberhardt, Herrn Prof. Dr. R. Klein und insbesondere Herrn Prof. Dr. D. Saupe, der uns auch einige Foliensätze überlassen hat. Wir freuen uns über jede weitere Anregung und sind stets offen für Verbesserungen.

Tübingen, Bonn, im August 2004

W. Küchlin, A. Weber

Aus dem Vorwort zur zweiten Auflage

Bei der ersten Auflage lag unser Hauptaugenmerk auf der Weiterentwicklung des klassischen Stoffs in die Tiefe, zur Objekttechnik hin. Nun haben wir das Buch in die Breite ergänzt, insbesondere in Hinblick auf einen Leserkreis, der an einem Einblick in die Struktur eines Rechnersystems und einer Gesamtsicht auf die Software-Entwicklung, auch in ihren klassischen Teilen, interessiert ist.

Danksagung. Wir danken allen unseren Mitarbeitern, die am Zustandekommen dieser Neuauflage beteiligt waren oder unter ihr zu leiden hatten: In Tübingen sind dies Dr. W. Blochinger, C. Sinz, M. Friedrich, R. Schimkat, G. Nusser und A. Kaiser; Frau E.-M. Dieringer hat einige neue Abbildungen angefertigt. In Bonn möchten wir Herrn G. Sobottka und Frau S. Schäfer für Korrekturvorschläge danken.

Besonders zu Dank verpflichtet sind wir unseren Lesern und denjenigen Kollegen, die die erste Auflage für Ihre Vorlesungen verwendet haben; wertvolle Hinweise verdanken wir Herrn Prof. Dr. B. Eberhardt und Herrn Prof. Dr. L. Voelkel.

Tübingen, Bonn, im Juli 2002

W. Küchlin, A. Weber

Aus dem Vorwort zur ersten Auflage

Dieses Buch fußt auf unseren Vorlesungen Informatik I und II an der Universität Tübingen. Wir schulden allen Dank, die am Zustandekommen dieser Vorlesungen in irgendeiner Form mitgewirkt haben, insbesondere natürlich den Mitarbeitern am Arbeitsbereich Symbolisches Rechnen.

Dr. W. Blochinger hat drei Semester lang verantwortlich die Übungen zu den Vorlesungen organisiert; ein Teil seiner Aufgaben ist auch in dieses Buch eingeflossen. Beiträge für die Übungen kamen auch von Dr. J. Hahn, Dr. B. Amrhein und

S. Müller, sowie von studentischen Tutoren, insbesondere von Ch. Ludwig. Kapitel 9 beruht auf Vorlagen von Dr. J. Hahn.

Für viele Korrekturen und nützliche Anregungen möchten wir Herrn Prof. Dr. U. Güntzer und Herrn Prof. Dr. M. Kaufmann herzlich danken, die eine Vorversion des Manuskripts durchgesehen haben. Teile des Manuskripts wurden ferner von Dr. D. Bühler, Dr. G. Hagel, A. Kaiser, Dr. Th. Lumpp, P. Maier, G. Nusser, R. Schimkat, C. Sinz und Dr. A. Speck korrekturgelesen.

Tübingen, Darmstadt, August 2000

W. Küchlin, A. Weber

These discussions were of great value to me in several ways. I was thus obliged to put into language the various views I had taken, and I observed the effect of my explanations on different minds. My own ideas became clearer, and I profited by many of the remarks made by my highly-gifted friends.

Charles Babbage (1864)

Inhaltsverzeichnis

1. Einführung und Überblick	1
1.1 Bedeutung und Grundprobleme der Informatik	1
1.1.1 Die Bedeutung des Berechnens von Funktionen	5
1.1.2 Das Problem der Komplexität	7
1.2 Konzeption des Buches	8
1.2.1 Aufbau des Buches	9
1.2.2 Hinweise für Dozenten	11

Teil I. Grundkonzepte von Hardware und Software

2. Aufbau und Funktionsweise eines Computers	15
2.1 Einleitung und Überblick	15
2.2 Der Kern des Rechners: von Neumann-Architektur	17
2.2.1 Speicher	18
2.2.2 Prozessor und Programmausführung	20
2.3 System-Architektur der Hardware	22
2.4 System-Architektur der Software	26
2.4.1 Schichtenaufbau	26
2.4.2 Das Betriebssystem	30
2.4.3 Java und die Virtuelle Java-Maschine JVM	31
2.5 Binärcodierung elementarer Datentypen	32
2.5.1 Ganze Zahlen (Dualzahlen)	33
2.5.2 Hexadezimalzahlen und Oktalzahlen	39
2.5.3 Zeichen (ASCII und Unicode)	39
2.5.4 Gleitkommazahlen (IEEE 754)	41
2.6 Übungen	45
3. Abstrakte Algorithmen und Sprachkonzepte	47
3.1 Einleitung und Begriffsdefinition	47
3.2 Aufbau und Beschreibung von Algorithmen	51
3.2.1 Textuelle Beschreibung in Schritten	51

3.2.2	Graphische Beschreibung mit UML (Flußdiagramme)	53
3.2.3	Grundschema des Algorithmenaufbaus	54
3.2.4	Strukturiert-iterative Beschreibungen	57
3.2.5	Rekursive Beschreibung in mathematischer Notation	59
3.2.6	Beschreibung mit Pseudo-Code	59
3.3	Programmiersprachliche Grundkonzepte	59
3.3.1	Das Sprung-Konzept <code>goto</code>	61
3.3.2	Verzweigung mit der bedingten Anweisung <code>if-then-else</code>	61
3.3.3	Rekursion	63
3.3.4	Die <code>while</code> -Schleife	64
3.3.5	Die <code>repeat-until</code> -Schleife	65
3.3.6	Die <code>for</code> -Schleife	66
3.4	Konstruktion und Verifikation rekursiver Algorithmen	66
3.4.1	Der rekursive Ansatz zur Problemlösung	66
3.4.2	Ein rekursives Verfahren in mathematischer Notation	67
3.4.3	Ein rekursives Verfahren in Java	68
3.5	Konstruktion und Verifikation iterativer Algorithmen	70
3.5.1	Der iterative Ansatz zur Problemlösung	70
3.5.2	Die Verifikation nach Floyd	73
3.5.3	Ein strukturiert-iteratives Verfahren in Java	76
3.6	Übungen	77
4.	Konzepte benutzerdefinierter Datenstrukturen	79
4.1	Einleitung	79
4.2	Reihungen (<i>arrays</i>)	80
4.3	Verbunde (<i>records, structs</i>)	81
4.4	Typ-Kombinationen von Reihung und Verbund	83
4.5	Modellierung des Enthaltenseins – Referenzen	83
4.6	Klassen, Objekte, abstrakte Datentypen	86
5.	Objektorientierte Software-Konzepte und UML	89
5.1	Objektorientierte Software-Entwicklung	89
5.2	Objekte, Klassen, abstrakte Datentypen	93
5.3	Objektbeziehungen	97
5.3.1	Informationsfluß- und Client/Server-Beziehungen	99
5.3.2	Einschlußbeziehungen (<i>has-a</i>)	102
5.3.3	Subtyp- bzw. Vererbungsbeziehungen (<i>is-a</i>)	104
5.4	Objektorientierte Analyse und Entwurf	106
5.4.1	Analyse einer Werkstück-Vereinzelungseinheit	106
5.5	Entwurfsmuster	110
5.5.1	Beispiel: Architekturmuster einer Gerätefernsteuerung	111
5.6	Übungen	115

Teil II. Sprachkonzepte und ihre Verwirklichung in Java

6.	Elementare Konzepte von Programmiersprachen	119
6.1	Einleitung und Überblick	119
6.2	Programmentwicklung in Java	121
6.2.1	Entwicklungsumgebungen für Java	122
6.2.2	Ein Rahmenprogramm für Java-Anweisungen	123
6.2.3	Ein Rahmenprogramm für Java-Funktionen	124
6.2.4	Übersetzung und Ausführung von Java-Programmen	125
6.3	Schlüsselwörter, Literale und Namen	127
6.4	Elementare Datentypen	129
6.5	Variablen, Referenzen, Zuweisungen	131
6.5.1	Grundkonzepte	131
6.5.2	Referenzvariablen	134
6.5.3	Reihungsvariablen	135
6.6	Java-Arithmetik	137
6.6.1	Elementare Zahltypen	137
6.6.2	Ganzzahl-Arithmetik	138
6.6.3	Gleitkomma-Arithmetik	139
6.7	Operatoren und Ausdrücke	141
6.7.1	Zuweisungsoperatoren	141
6.7.2	Arithmetische Operatoren	141
6.7.3	Boolesche Operatoren	142
6.7.4	Bitmuster	143
6.7.5	Ausdrücke	144
6.7.6	Syntax von Ausdrücken	145
6.7.7	Präzedenz von Operatoren	145
6.7.8	Semantik von Ausdrücken	147
6.7.9	Bedingte Ausdrücke	150
6.7.10	Typkonversionen	151
6.8	Anweisungen	154
6.8.1	Blöcke, Gültigkeitsbereich und Lebensdauer	155
6.8.2	Bedingte Anweisungen (if und switch)	159
6.8.3	Schleifenkonstrukte (while, do-while, for)	161
6.8.4	Marken, break und continue	165
6.9	Unterprogramme – Prozeduren und Funktionen	168
6.9.1	Konzepte und Terminologie	168
6.9.2	Unterprogramme in Java	173
6.9.3	Parameterübergabe und Laufzeitstapel	174
6.9.4	Spezifikation von Unterprogrammen	183

6.9.5	Rekursion	187
6.9.6	Allgemeine Rekursion und Speicherverwaltung	190
6.10	Übungen	193
7.	Klassen und höhere Datentypen	199
7.1	Einleitung und Überblick	199
7.2	Objekte, Felder und Methoden	201
7.2.1	Überladen von Methoden	204
7.2.2	Klassenvariablen und Klassenmethoden	204
7.2.3	Pakete (<i>packages</i>)	205
7.2.4	Kapselung und Zugriffskontrolle	206
7.2.5	Kontrakt und Aufrufchnittstelle	207
7.2.6	Verwaltung von Objekten im Speicher	208
7.2.7	Initialisierung und Konstruktoren	212
7.2.8	Selektoren	215
7.2.9	Beispiel eines Datentyps: komplexe Zahlen	216
7.3	Objekte für Ausnahmen (<i>exceptions</i>)	218
7.3.1	Einleitung und Überblick	218
7.3.2	Ausnahmeklassen	220
7.3.3	Die <i>throw</i> -Anweisung	222
7.3.4	Der Rahmen <i>try-catch-finally</i>	223
7.3.5	Deklaration von Ausnahmen mit <i>throws</i>	224
7.4	Wahrheitsbehauptungen und Zusicherungen (<i>assertions</i>)	226
7.5	Reihungen (<i>arrays</i>)	230
7.5.1	Allgemeine Konzepte, Terminologie und Realisierung	230
7.5.2	Eindimensionale Reihungen	232
7.5.3	Skalar- und Vektor-Operationen	234
7.5.4	Mehrdimensionale Reihungen und Matrizen	238
7.6	Zeichenketten (<i>strings</i>)	241
7.6.1	Veränderliche Zeichenketten	243
7.7	Listen (<i>linked lists</i>)	243
7.7.1	Konzepte, Terminologie und Entwurf	244
7.7.2	Die Implementierung von Listen	246
7.7.3	Einfügen eines Elementes	248
7.7.4	Sortiertes Einfügen eines Elements	249
7.7.5	Invertieren einer Liste	252
7.7.6	Doppelt verkettete Listen (<i>doubly linked lists</i>)	253
7.8	Stapel (<i>stacks</i>)	257
7.8.1	Konzept und Terminologie	257
7.8.2	Implementierung von Stacks	258
7.9	Warteschlangen (<i>queues</i>)	259

7.9.1	Konzept und Terminologie	259
7.9.2	Implementierung von Queues	260
7.10	Übungen	261
8.	Höhere objektorientierte Konzepte	263
8.1	Einleitung	263
8.2	Vererbung und abgeleitete Klassen	265
8.2.1	Der Zugriffsschutz <code>protected</code> in Klassenhierarchien	267
8.2.2	Konstruktoren in Klassen-Hierarchien	268
8.3	Virtuelle Funktionen und dynamisches Binden	270
8.3.1	Konzepte und Terminologie	270
8.3.2	Realisierung des dynamischen Bindens	273
8.3.3	Klassenkontrakte und virtuelle Funktionen	274
8.3.4	Typanpassungen in Klassenhierarchien	274
8.3.5	Zugriffsregeln und Auswahlregeln in Klassenhierarchien – Überschreiben und Verdecken	275
8.4	Abstrakte Klassen und Interfaces	279
8.4.1	Abstrakte Klassen	279
8.4.2	Schnittstellen (<i>interfaces</i>)	280
8.5	Mehrfachvererbung	281
8.6	Generisches Programmieren	282
8.6.1	Generische Datentypen	282
8.6.2	Generische Methoden	284
8.6.3	Explizite Typkonversion	285
8.6.4	Klassen-Muster (<i>template classes</i>) und generisches Java	286
8.6.5	Generische Funktionsparameter	289
8.7	Übungen	294
9.	Das „Abstract Window Toolkit“ (AWT)	299
9.1	Graphische Komponenten	300
9.1.1	Klassenhierarchie der graphischen Komponenten	300
9.1.2	Funktionalität von <code>Component</code>	300
9.1.3	Die Klasse <code>Graphics</code>	301
9.1.4	Frames	302
9.1.5	Applets	304
9.1.6	Container	305
9.2	Ereignisse (<i>events</i>)	308
9.2.1	AWT-Events	308
9.2.2	Ereignisquellen und Ereignisempfänger	308
9.2.3	Adapter-Klassen	310
9.3	Ein Beispiel: Ein Rahmen zum Zeichnen reeller Funktionen	310
9.4	Ein größeres Beispiel: Darstellung einer Winterlandschaft	314

9.4.1	Anforderungsanalyse	315
9.4.2	Objektorientierte Analyse und Design	315
9.4.3	Implementierung der Klassen	316

Teil III. Algorithmen und weiterführende Datenstrukturen

10.	Theorie der Algorithmenkonstruktion	327
10.1	Einleitung und Überblick	327
10.1.1	Motivation und Begriffsdefinition	327
10.1.2	Notation	329
10.2	Problemspezifikation und Korrektheitsbeweise	330
10.2.1	Spezifikation	330
10.2.2	Partielle Korrektheit	332
10.2.3	Terminierung	332
10.2.4	Beispiel: Berechnung der Quadratwurzel	334
10.3	Schemata für den Algorithmenentwurf	336
10.4	Aufwand und asymptotische Komplexität	339
10.4.1	Exakte Bestimmung der Komplexität	341
10.4.2	Asymptotische Notation	343
11.	Such-Algorithmen	349
11.1	Einleitung und Problemstellung	349
11.2	Lineare Suche	349
11.2.1	Suche mit Wächter	350
11.2.2	Komplexität der linearen Suche	351
11.3	Divide-and-Conquer-Suche	352
11.3.1	Komplexität der binären Suche	353
11.4	Kombinationsverfahren	354
11.4.1	Analyse und Design von Kombinationsverfahren	355
12.	Sortier-Algorithmen	357
12.1	Einleitung und Problemstellung	357
12.2	Greedy-Sortieren	358
12.2.1	Sortieren durch Auswahl	358
12.2.2	Sortieren durch Einfügen	360
12.2.3	Sortieren durch Austauschen	361
12.3	Divide-and-Conquer-Sortieren	362
12.3.1	Quicksort	363
12.3.2	Sortieren durch Mischen	366
12.4	Übungen	369

13. Bäume	371
13.1 Einleitung und Terminologie	371
13.2 Graphen und Bäume	372
13.2.1 Gerichtete Graphen	372
13.2.2 Ungerichtete Graphen	373
13.2.3 Bäume als ungerichtete Graphen	374
13.3 Eigenschaften von Bäumen	374
13.4 Implementierung von Bäumen	376
13.5 Baumdurchläufe	377
13.5.1 Aktionsobjekte für generische Baumdurchläufe	377
13.5.2 Präorder-Sequenz	379
13.5.3 Postorder-Sequenz	381
13.5.4 Inorder-Sequenz	382
13.5.5 Levelorder-Sequenz	384
13.5.6 Optimierung der Baumdurchläufe	385
13.6 Übungen	387
14. Hashing	389
14.1 Einleitung	389
14.2 Hash-Funktionen	390
14.3 Kollisionsbehandlung	392
14.3.1 Separates Ketten	392
14.3.2 Offenes Adressieren	393
14.4 Hash-Tabellen in Java	393
14.5 Übungen	398

Teil IV. Theoretische Grundlagen

15. Mathematische Grundlagen	401
15.1 Einleitung	401
15.2 Mengen	402
15.3 Relationen	404
15.3.1 Binäre Relationen	404
15.3.2 Äquivalenzrelationen	405
15.4 Funktionen	406
15.4.1 Partielle Funktionen	406
15.4.2 Totale Funktionen	406
15.4.3 Definitions- und Bildbereich von Funktionen	406
15.4.4 Eigenschaften von Funktionen	407
15.4.5 Charakteristische Funktionen	408
15.5 Ordnungen	409

15.5.1	Partielle und totale Ordnungen	409
15.5.2	Lexikographische Ordnung	410
15.5.3	Multiset-Ordnungen	410
15.6	Das Prinzip der vollständigen Induktion	411
15.7	Übungen	412
16.	Einführung in die Logik	415
16.1	Einleitung	415
16.2	Die Algebra der Booleschen Wahrheitswerte	416
16.3	Aussagenlogik (<i>PROP</i>)	417
16.3.1	Die Syntax der Aussagenlogik	417
16.3.2	Semantik der Aussagenlogik	418
16.4	Prädikatenlogik erster Stufe (<i>FOPL</i>)	421
16.4.1	Syntax von <i>FOPL</i>	421
16.4.2	Semantik von <i>FOPL</i>	422
16.5	Beweise	424
16.5.1	Logische Äquivalenzen	424
16.5.2	Ableitungen und Logik-Kalküle	425
16.5.3	Beweisbäume	426
16.6	Übungen	427
17.	Korrektheit von Unterprogrammen	429
17.1	Terminologie und Problemstellung	429
17.2	Der Hoare-Kalkül	431
17.2.1	Regeln des Hoare-Kalküls	432
17.2.2	Konsequenzregeln	434
17.2.3	Zuweisungsaxiom	434
17.2.4	Sequenzregel	437
17.2.5	Alternativregeln	439
17.2.6	Iterationsregel	441
17.3	Übungen	444
	Literaturverzeichnis	445
	Index	449