



Springer-Verlag Berlin Heidelberg GmbH

Wolfgang Kuchlin • Andreas Weber

# Einführung in die Informatik

Objektorientiert mit Java

Mit 32 Abbildungen und 2 Tabellen



Springer

Professor Dr. Wolfgang Küchlin  
Wilhelm-Schickard-Institut für Informatik  
Universität Tübingen  
Sand 13, 72076 Tübingen  
kuechlin@informatik.uni-tuebingen.de  
<http://www-sr.informatik.uni-tuebingen.de>

Dr. Andreas Weber  
Fraunhofer-Institut für Graphische Datenverarbeitung  
Rundeturmstr. 6, 64283 Darmstadt  
aweber@igd.fhg.de  
<http://www.igd.fhg.de/~aweber>

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

ISBN 978-3-540-67384-2

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Küchlin, Wolfgang:  
Einführung in die Informatik: objektorientiert mit Java/Wolfgang  
Küchlin; Andreas Weber.

(Springer-Lehrbuch)

ISBN 978-3-540-67384-2

ISBN 978-3-662-06854-0 (eBook)

DOI 10.1007/978-3-662-06854-0

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-Verlag Berlin Heidelberg 2000

Ursprünglich erschienen bei Springer-Verlag Berlin Heidelberg New York 2000

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 Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Umschlaggestaltung: design & production GmbH, Heidelberg

SPIN 10717065

45/3142SR – 5 4 3 2 1 0 – Gedruckt auf säurefreiem Papier

*Unseren Familien gewidmet*  
**WK, AW**

# Vorwort

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

*Charles Babbage (1864)*

Der Computer ist eine Universalmaschine zur Informationsverarbeitung, und die Informatik (*Computer Science*) als Wissenschaft der Theorie, Technik und Anwendung von Computern ist eine universelle Querschnittswissenschaft wie kaum eine andere. Bereits die Pioniere Babbage (um 1840) und Turing (um 1940) hatten erkannt, daß der Computer vom Konzept her Teile der Mathematik revolutionieren bzw. darüber hinaus als universeller Symbolverarbeiter dienen würde. In jüngster Zeit rückt mit dem Internet insbesondere der Computer als universelles Kommunikationswerkzeug in den Vordergrund. Heute, runde 150 Jahre nach Babbage, 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 aber neue Werkzeuge und Methoden, auch wenn die alten rein theoretisch noch genügen würden. Kaum eine andere Wissenschaft befindet sich daher noch so im Umbruch wie die Informatik.

Objektorientierte Software-Entwicklung und die Programmiersprache Java sind vielleicht 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, insbesondere durch vielerlei standardisierte Schnittstellen von grafischen Oberflächen bis Datenbanken und Netzen, für den Einsatz in der breiten Praxis geeignet ist.

Der zentrale Beweggrund für dieses neue Lehrbuch der Informatik ist das Erreichen einer Balance zwischen Theorie und Praxis, also zwischen theoretisch-konzeptuellen Grundlagen und unmittelbar praxisrelevantem Wissen. Dieses Lehrbuch soll die traditionellen Konzepte, die in der Einführungsvorlesung Informatik an Universitäten gelehrt werden, um den Gesichtspunkt der Objektorientierung ergänzen und aus dieser Sicht neu präsentieren sowie anhand von Java einüben. Der Leser soll insbesondere

- grundlegende Konzepte der Software-Entwicklung und des Programmierens (objektorientiertes Modellieren, Datenstrukturen, Programmierkonstrukte) verstehen,
- eine moderne objektorientierte Sprache erlernen, die diese Konzepte umsetzt und 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 logischen und mathematischen Grundlagen der praktischen Informatik erwerben, um eine Basis für lebenslanges Lernen zu erhalten.

*Danksagung.* Dieses Buch fußt auf den Vorlesungen Informatik I und II an der Universität Tübingen in den Jahren 1994/95 (mit C++) und 1997/98 und 1998/99 (mit Java). Wir schulden allen Dank, die am Zustandekommen dieser Vorlesungen in irgendeiner Form mitgewirkt haben, insbesondere natürlich den Mitarbeitern am Arbeitsbereich Symbolisches Rechnen.

Wolfgang Blochinger hat drei Semester lang verantwortlich die Übungen organisiert, die ein wichtiger Bestandteil der Vorlesungen waren; ein Teil seiner Aufgaben ist auch in dieses Buch eingeflossen. Beiträge für die Übungen kamen auch von Dr. Jens Hahn und Stefan Müller sowie von studentischen Tutoren, insbesondere von Christoph Ludwig. Ideen für einige der Übungen gehen noch auf die C++-basierte Vorlesung des Jahres 1994/95 zurück, deren Übungen zu größeren Teilen von Frau Dr. Beatrice Amrhein erstellt wurden. Sie überarbeitete auch ein Skriptum für die C++-Vorlesung, das für die Erstellung eines ersten Vorlesungsskriptums für die jetzt mit Java gehaltenen Vorlesungen sehr nützlich war.

Kapitel 9 beruht zu wesentlichen Teilen auf Vorlagen von Dr. Jens Hahn, die er uns freundlicherweise zur Verfügung gestellt hat und die für uns sehr wertvoll waren.

Große Teile des Manuskripts wurden von Herrn O. Bausinger, Frau M. Quenzer, Frau M. Ludwig und Herrn C. Pomm erfaßt. Herr Bausinger überarbeitete und testete darüber hinaus einige der Programme. Für sein Engagement bei all diesen Arbeiten, die uns sehr geholfen haben, sei ihm herzlich gedankt.

Für viele Korrekturen und nützliche Anregungen möchten wir Herrn Prof. Dr. U. Güntzer und Herrn Prof. Dr. M. Kaufmann herzlich danken, die mit erheblichem Zeitaufwand eine Vorversion des Manuskripts durchgesehen haben. Teile des Manuskripts wurden ferner von Dieter Bühler, Dr. Georg Hagel, Andreas Kaiser, Dr. Thomas Lumpp, Patrick Maier, Gerd Nusser, Ralf-Dieter Schimkat, Carsten Sinz und Dr. Andreas Speck korrekturgelesen. Viele Fehler konnten durch ihre Hilfe gefunden werden. Für alle verbliebenen Fehler und Ungereimtheiten tragen wir aber natürlich allein die Verantwortung.

Nicht zuletzt möchten wir Herrn Dr. Hans Wössner vom Springer-Verlag für die gute und hilfreiche Zusammenarbeit danken.

# Inhaltsverzeichnis

<b>1. Einführung und Überblick</b> .....	1
1.1 Bedeutung und Grundprobleme der Informatik .....	1
1.1.1 Die Bedeutung des Berechnens von Funktionen .....	2
1.1.2 Das Problem der Komplexität .....	4
1.2 Konzeption des Buches .....	5
1.2.1 Objektorientierung und Java .....	5
1.2.2 Aufbau des Buches .....	7
1.2.3 Verwendung in der Grundvorlesung Informatik .....	9
1.2.4 Englische Begriffe .....	9

---

## Teil I. Konzepte der Software-Entwicklung

---

<b>2. Datenstrukturen</b> .....	13
2.1 Einleitung .....	13
2.2 Reihungen ( <i>arrays</i> ) .....	14
2.3 Verbunde ( <i>records, structs</i> ) .....	14
2.4 Typ-Kombinationen von Reihung und Verbund .....	16
2.5 Modellierung des Enthaltenseins – Referenzen .....	16
2.6 Abstrakte Datentypen und Objekte .....	18
<b>3. Objektorientierte Software-Entwicklung</b> .....	21
3.1 Einleitung .....	21
3.2 Objekte, Klassen, abstrakte Datentypen .....	22
3.3 Objektbeziehungen .....	25
3.3.1 Informationsfluß- und Client/Server-Beziehungen .....	26
3.3.2 Einschlußbeziehungen ( <i>has-a</i> ) .....	29
3.3.3 Subtyp- bzw. Vererbungsbeziehungen ( <i>is-a</i> ) .....	31
3.4 Objektorientierte Analyse und Entwurf .....	32
3.4.1 Analyse einer Werkstück-Vereinzelungseinheit .....	33
3.5 Entwurfsmuster .....	38

- 3.5.1 Beispiel: Architekturmuster einer Gerätefernsteuerung . . . . . 38
- 3.6 Übungen . . . . . 42
- 4. Algorithmen und algorithmische Sprachkonzepte . . . . . 43**
  - 4.1 Einleitung und Begriffsdefinition . . . . . 43
  - 4.2 Beispiel: Berechnung der modulus-Funktion . . . . . 46
    - 4.2.1 Der rekursive Ansatz . . . . . 46
    - 4.2.2 Ein rekursives Verfahren in mathematischer Notation . . . . . 46
    - 4.2.3 Ein rekursives Verfahren in Java . . . . . 48
    - 4.2.4 Der strukturiert-iterative Ansatz . . . . . 49
    - 4.2.5 Ein strukturiert-iteratives Verfahren in Java . . . . . 49
  - 4.3 Algorithmische Notation und abstrakte Sprachkonzepte . . . . . 51
    - 4.3.1 Abstrakte Beschreibung von Algorithmen . . . . . 52
    - 4.3.2 Programmiersprachliche Grundkonzepte zur Beschreibung von Algorithmen . . . . . 54
  - 4.4 Übungen . . . . . 60

---

**Teil II. Sprachkonzepte und ihre Verwirklichung in Java**

---

- 5. Grundlagen der Rechnerarchitektur und elementare Datentypen . . . 63**
  - 5.1 Speicheraufbau . . . . . 64
  - 5.2 Binäre Repräsentation ganzer Zahlen . . . . . 65
    - 5.2.1 Repräsentation von Bitmustern . . . . . 68
  - 5.3 Binäre Repräsentation von Zeichen . . . . . 68
    - 5.3.1 Der ASCII-Zeichensatz . . . . . 68
  - 5.4 Gleitkommazahlen . . . . . 70
  - 5.5 Arithmetik in Java . . . . . 71
    - 5.5.1 Ganzzahl-Arithmetik . . . . . 71
    - 5.5.2 Gleitkomma-Arithmetik . . . . . 72
  - 5.6 Übungen . . . . . 73
- 6. Grundkonzepte von Programmiersprachen . . . . . 77**
  - 6.1 Einleitung und Übersicht . . . . . 77
  - 6.2 Programmentwicklung in Java . . . . . 79
    - 6.2.1 Die Struktur von Java . . . . . 79
    - 6.2.2 Ein Rahmenprogramm für Java-Anweisungen . . . . . 80
  - 6.3 Elementare Datentypen . . . . . 82
  - 6.4 Variablen, Referenzen, Zuweisungen . . . . . 83
    - 6.4.1 Reihungsvariablen . . . . . 85
  - 6.5 Operatoren und Ausdrücke . . . . . 86
    - 6.5.1 Zuweisungsoperatoren . . . . . 86

6.5.2	Arithmetische Operatoren .....	87
6.5.3	Boolesche Operatoren .....	87
6.5.4	Bitmuster .....	89
6.5.5	Ausdrücke .....	90
6.5.6	Syntax von Ausdrücken .....	91
6.5.7	Präzedenz von Operatoren .....	91
6.5.8	Semantik von Ausdrücken .....	92
6.5.9	Bedingte Ausdrücke .....	94
6.5.10	Typkonversionen .....	95
6.6	Anweisungen .....	98
6.6.1	Blöcke, Gültigkeitsbereich und Lebensdauer .....	99
6.6.2	Bedingte Anweisungen (if und switch) .....	102
6.6.3	Schleifenkonstrukte (while, do-while, for) .....	105
6.6.4	Marken, break und continue .....	108
6.7	Unterprogramme – Prozeduren und Funktionen .....	110
6.7.1	Konzepte und Terminologie .....	110
6.7.2	Unterprogramme in Java .....	114
6.7.3	Parameterübergabe und Laufzeitstapel .....	116
6.7.4	Spezifikation von Unterprogrammen .....	124
6.7.5	Rekursion .....	128
6.7.6	Allgemeine Rekursion und Speicherverwaltung .....	131
6.8	Übungen .....	134
<b>7.</b>	<b>Klassen und höhere Datentypen .....</b>	<b>137</b>
7.1	Einführung und Überblick .....	137
7.2	Objekte, Felder und Methoden .....	138
7.2.1	Überladen von Methoden .....	141
7.2.2	Klassenvariablen und Klassenmethoden .....	142
7.2.3	Pakete ( <i>packages</i> ) .....	143
7.2.4	Kapselung und Zugriffskontrolle .....	144
7.2.5	Kontrakt und Aufrufchnittstelle .....	145
7.2.6	Verwaltung von Objekten im Speicher .....	146
7.2.7	Initialisierung und Konstruktoren .....	149
7.2.8	Selektoren .....	151
7.2.9	Beispiel eines Datentyps: komplexe Zahlen .....	154
7.3	Objekte für Ausnahmen ( <i>exceptions</i> ) .....	156
7.3.1	Einleitung und Überblick .....	156
7.3.2	Ausnahmeklassen .....	158
7.3.3	Die throw-Anweisung .....	160
7.3.4	Der Rahmen try-catch-finally .....	161
7.3.5	Deklaration von Ausnahmen mit throws .....	162

7.4	Reihungen ( <i>arrays</i> ) .....	164
7.4.1	Konzepte und Terminologie .....	164
7.4.2	Eindimensionale Reihungen .....	164
7.4.3	Skalar- und Vektor-Operationen .....	166
7.4.4	Vektoren .....	168
7.4.5	Mehrdimensionale Reihungen .....	169
7.5	Zeichenketten ( <i>strings</i> ) .....	172
7.5.1	Veränderliche Zeichenketten .....	173
7.6	Listen .....	174
7.6.1	Konzepte, Terminologie und Entwurf .....	174
7.6.2	Die Implementierung von Listen .....	176
7.6.3	Einfügen eines Elementes .....	178
7.6.4	Sortiertes Einfügen eines Elements .....	179
7.6.5	Invertieren einer Liste .....	181
7.6.6	Doppelt verkettete Listen .....	183
7.7	Stapel ( <i>stacks</i> ) .....	187
7.7.1	Konzept und Terminologie .....	187
7.7.2	Implementierung von Stacks .....	188
7.8	Warteschlangen ( <i>queues</i> ) .....	189
7.8.1	Konzept und Terminologie .....	189
7.8.2	Implementierung von Queues .....	189
7.9	Übungen .....	190
<b>8.</b>	<b>Höhere objektorientierte Konzepte .....</b>	<b>191</b>
8.1	Einleitung .....	191
8.2	Vererbung und abgeleitete Klassen .....	193
8.2.1	Der Zugriffsschutz <code>protected</code> in Klassenhierarchien .....	195
8.2.2	Konstruktoren in Klassen-Hierarchien .....	196
8.3	Virtuelle Funktionen und dynamisches Binden .....	198
8.3.1	Konzepte und Terminologie .....	198
8.3.2	Realisierung des dynamischen Bindens .....	201
8.3.3	Klassenkontrakte und virtuelle Funktionen .....	202
8.3.4	Typanpassungen in Klassenhierarchien .....	203
8.3.5	Zugriffsregeln und Auswahlregeln in Klassenhierarchien – Überschreiben und Verdecken .....	204
8.4	Abstrakte Klassen und Interfaces .....	208
8.4.1	Abstrakte Klassen .....	208
8.4.2	Schnittstellen ( <i>interfaces</i> ) .....	209
8.5	Mehrfachvererbung .....	209
8.6	Generisches Programmieren .....	211
8.6.1	Generische Datentypen .....	211

- 8.6.2 Generische Methoden ..... 212
- 8.6.3 Explizite Typkonversion ..... 214
- 8.6.4 Klassen-Muster (*template classes*) ..... 215
- 8.6.5 Generische Funktionsparameter ..... 216
- 8.7 Übungen ..... 220
- 9. Das „Abstract Window Toolkit“ (AWT) von Java ..... 221**
  - 9.1 Graphische Komponenten ..... 221
    - 9.1.1 Klassenhierarchie der graphischen Komponenten ..... 221
    - 9.1.2 Funktionalität von Component ..... 221
    - 9.1.3 Die Klasse Graphics ..... 223
    - 9.1.4 Container ..... 224
    - 9.1.5 Frames ..... 224
    - 9.1.6 Applets ..... 225
  - 9.2 Ereignisse (*events*) ..... 226
    - 9.2.1 AWT-Events ..... 226
    - 9.2.2 Ereignisquellen und Ereignisempfänger ..... 226
    - 9.2.3 Adapter-Klassen ..... 228
  - 9.3 Ein Beispiel: Ein Rahmen zum Zeichnen reeller Funktionen ..... 228
  - 9.4 Ein größeres Beispiel: Darstellung einer Winterlandschaft ..... 232
    - 9.4.1 Anforderungsanalyse ..... 232
    - 9.4.2 Objektorientierte Analyse und Design ..... 232
    - 9.4.3 Implementierung der Klassen ..... 235

---

**Teil III. Algorithmen und weiterführende Datenstrukturen**

---

- 10. Theorie der Algorithmenkonstruktion ..... 247**
  - 10.1 Einleitung und Überblick ..... 247
    - 10.1.1 Motivation und Begriffsdefinition ..... 247
    - 10.1.2 Notation ..... 249
  - 10.2 Problemspezifikation und Korrektheitsbeweise ..... 250
    - 10.2.1 Spezifikation ..... 250
    - 10.2.2 Partielle Korrektheit ..... 252
    - 10.2.3 Terminierung ..... 252
    - 10.2.4 Beispiel: Berechnung der Quadratwurzel ..... 253
  - 10.3 Schemata für den Algorithmenentwurf ..... 256
  - 10.4 Aufwand und asymptotische Komplexität ..... 259
    - 10.4.1 Exakte Bestimmung der Komplexität ..... 260
    - 10.4.2 Asymptotische Notation ..... 263

- 11. Such-Algorithmen** ..... 269
  - 11.1 Einleitung und Problemstellung ..... 269
  - 11.2 Lineare Suche ..... 269
    - 11.2.1 Suche mit Wächter ..... 270
    - 11.2.2 Komplexität der linearen Suche ..... 271
  - 11.3 Divide-and-Conquer-Suche ..... 272
    - 11.3.1 Komplexität der binären Suche ..... 273
  - 11.4 Kombinations-Suche ..... 274
  
- 12. Sortier-Algorithmen** ..... 277
  - 12.1 Einleitung und Problemstellung ..... 277
  - 12.2 Greedy-Sortieren ..... 278
    - 12.2.1 Sortieren durch Auswahl ..... 278
    - 12.2.2 Sortieren durch Einfügen ..... 280
    - 12.2.3 Sortieren durch Austauschen ..... 281
  - 12.3 Divide-and-Conquer-Sortieren ..... 282
    - 12.3.1 Quicksort ..... 283
    - 12.3.2 Sortieren durch Mischen ..... 286
  - 12.4 Übungen ..... 289
  
- 13. Bäume** ..... 291
  - 13.1 Einleitung und Terminologie ..... 291
  - 13.2 Graphen und Bäume ..... 292
    - 13.2.1 Gerichtete Graphen ..... 292
    - 13.2.2 Ungerichtete Graphen ..... 293
    - 13.2.3 Bäume als ungerichtete Graphen ..... 294
  - 13.3 Eigenschaften von Bäumen ..... 294
  - 13.4 Implementierung von Bäumen ..... 296
  - 13.5 Baumdurchläufe ..... 297
    - 13.5.1 Aktionsobjekte für Baumdurchläufe ..... 297
    - 13.5.2 Präorder-Sequenz ..... 299
    - 13.5.3 Postorder-Sequenz ..... 301
    - 13.5.4 Inorder-Sequenz ..... 302
    - 13.5.5 Levelorder-Sequenz ..... 304
    - 13.5.6 Optimierung der Baumdurchläufe ..... 305
  - 13.6 Übungen ..... 307
  
- 14. Hashing** ..... 309
  - 14.1 Einleitung ..... 309
  - 14.2 Hash-Funktionen ..... 310
  - 14.3 Kollisionsbehandlung ..... 312
    - 14.3.1 Separates Ketten ..... 312

14.3.2 Offenes Adressieren ..... 313  
 14.4 Hash-Tabellen in Java ..... 313  
 14.5 Übungen ..... 316

**Teil IV. Theoretische Grundlagen**

**15. Mathematische Grundlagen** ..... 319  
 15.1 Einleitung ..... 319  
 15.2 Mengen ..... 320  
 15.3 Relationen ..... 321  
     15.3.1 Binäre Relationen ..... 322  
     15.3.2 Äquivalenzrelationen ..... 323  
 15.4 Funktionen ..... 324  
     15.4.1 Partielle Funktionen ..... 324  
     15.4.2 Totale Funktionen ..... 324  
     15.4.3 Definitions- und Bildbereich von Funktionen ..... 324  
     15.4.4 Eigenschaften von Funktionen ..... 325  
     15.4.5 Charakteristische Funktionen ..... 325  
 15.5 Ordnungen ..... 326  
     15.5.1 Partielle und totale Ordnungen ..... 326  
     15.5.2 Lexikographische Ordnung ..... 327  
     15.5.3 Multiset-Ordnungen ..... 328  
 15.6 Das Prinzip der vollständigen Induktion ..... 329  
 15.7 Übungen ..... 330

**16. Einführung in die Logik** ..... 333  
 16.1 Einleitung ..... 333  
 16.2 Die Algebra der Booleschen Wahrheitswerte ..... 334  
 16.3 Aussagenlogik (*PROP*) ..... 335  
     16.3.1 Die Syntax der Aussagenlogik ..... 335  
     16.3.2 Semantik der Aussagenlogik ..... 336  
 16.4 Prädikatenlogik erster Stufe (*FOPL*) ..... 339  
     16.4.1 Syntax von *FOPL* ..... 339  
     16.4.2 Semantik von *FOPL* ..... 340  
 16.5 Beweise ..... 342  
     16.5.1 Logische Äquivalenzen ..... 342  
     16.5.2 Ableitungen und Logik-Kalküle ..... 343  
     16.5.3 Beweisbäume ..... 345  
 16.6 Übungen ..... 346

<b>17. Korrektheit von Unterprogrammen</b> .....	347
17.1 Terminologie und Problemstellung .....	347
17.2 Der Hoare-Kalkül .....	349
17.2.1 Regeln des Hoare-Kalküls .....	350
17.2.2 Konsequenzregeln .....	352
17.2.3 Zuweisungsaxiom .....	352
17.2.4 Sequenzregel .....	355
17.2.5 Alternativregeln .....	357
17.2.6 Iterationsregel .....	359
17.3 Übungen .....	362
<b>Literaturverzeichnis</b> .....	363
<b>Index</b> .....	367