

Inhaltsverzeichnis

Vorwort	17
Kapitel 1 Einleitung	19
1.1 Idee dieses Buches	20
1.2 Beispiele, Übungen und Rätsel	21
1.3 Begleitmaterial zu diesem Buch	22
1.4 MyLab Grundlagen der Informatik	23
1.5 Danksagung	23
1.6 Hinweis in eigener Sache	24
Teil I Einführung in die Informatik	25
Kapitel 2 Die Historie und die Teilgebiete der Informatik	27
2.1 Rätsel: Streichholzprobleme	28
2.2 Der Begriff Informatik	28
2.3 Historische Entwicklung der Informatik	28
2.3.1 Der Abakus	28
2.3.2 Der Begriff Algorithmus und Ibn Musa Al-Chwarismi	31
2.3.3 Wichtige Stationen von 1500 bis 1930	32
2.3.4 Konrad Zuse und der erste funktionstüchtige Computer	34
2.3.5 Howard H. Aiken und die Mark I	36
2.3.6 John von Neumann	36
2.3.7 Generationen der elektronischen Datenverarbeitung	37
2.4 Einordnung und Einteilung der Informatik	41
2.4.1 Verschiedene Einsatzgebiete von Computern (Informatik)	41
2.4.2 Die Teilgebiete der Informatik	42
2.4.3 Die Informatik und unsere Abhängigkeit von ihr	45
Kapitel 3 Speicherung und Interpretation von Information	47
3.1 Rätsel: Umfüllprobleme	48
3.2 Unterschiedliche Zahlensysteme	48
3.2.1 Das römische Zahlensystem	48
3.2.2 Positionssysteme	49
3.2.3 Positionssysteme bei natürlichen Zahlen	50
3.2.4 Positionssysteme bei gebrochenen Zahlen	55
3.3 Dual-, Oktal- und Hexadezimalsystem	56
3.3.1 Das Dualsystem und das Bit im Rechner	56
3.3.2 Konvertieren zwischen Dual- und Oktalsystem	57
3.3.3 Konvertieren zwischen Dual- und Hexadezimalsystem	57

3.4	Konvertierungsalgorithmen	59
3.4.1	Konvertieren von anderen Systemen in das Dezimalsystem	59
3.4.2	Konvertieren vom Dezimalsystem in andere Positionssysteme ...	59
3.4.3	Konvertieren echt gebrochener Zahlen	60
3.4.4	Konvertieren unecht gebrochener Zahlen	62
3.5	Rechenoperationen im Dualsystem	62
3.5.1	Addition	62
3.5.2	Subtraktion und Darstellung negativer Zahlen	63
3.5.3	Multiplikation und Division	67
3.5.4	Konvertieren durch sukzessive Multiplikation und Addition ...	67
3.6	Reelle Zahlen	68
3.6.1	Festpunktzahlen	68
3.6.2	Gleitpunktzahlen und das IEEE-Format	68
3.7	Codes zur Darstellung von Zeichen	71
3.7.1	ASCII-Code	71
3.7.2	Unicode	74
3.8	Weitere Codes für Zahlen und Zeichen	75
3.8.1	BCD-Code für Zahlen	75
3.8.2	Gray-Code	76
3.8.3	Barcode	77
3.9	Duale Größenangaben	77
3.10	Die Grunddatentypen in der Programmiersprache C/C++	78

Kapitel 4 Boolesche Algebra **81**

4.1	Rätsel: Analytische Rätsel (1)	82
4.2	George Boole und seine Algebra mit nur zwei Werten	82
4.3	Operatoren	83
4.4	Boolesche Schaltungen	85
4.5	Boolesche Rechenregeln	85
4.6	Funktionen	87

Kapitel 5 Hardwarekomponenten eines Computers **91**

5.1	Rätsel: Analytische Rätsel (2)	92
5.2	Aufbau von Computersystemen	92
5.2.1	Zentraleinheit und Peripheriegeräte	92
5.2.2	EVA und das von-Neumann'sche-Rechnermodell	94
5.3	Die heutigen Personal Computer (PCs)	95
5.4	Die Zentraleinheit	96
5.4.1	Der Prozessor	97
5.4.2	Der Arbeitsspeicher	108
5.4.3	ROMs zur Speicherung von Programmen und konstanten Daten .	110
5.4.4	Das BIOS	112
5.4.5	Busse und Schnittstellen (Anschlüsse)	113
5.5	Die Peripherie	118

5.5.1	Massenspeicher	118
5.5.2	Eingabegeräte	123
5.5.3	Ausgabegeräte	125
5.6	Modell eines einfachen Prozessorsystems	129
5.7	Alternative Rechnerarchitekturen (Neuronale Netze)	134

Kapitel 6 Vom Programm zum Maschinenprogramm 135

6.1	Rätsel: Analytische Rätsel (3)	136
6.2	Entwicklung eines Programms	136
6.3	Programmierwerkzeuge	137
6.3.1	Unterschiedliche Arten der Übersetzung	137
6.3.2	Der Compiler	138
6.3.3	Der Linker	139
6.3.4	Der Lader (und Locator)	141
6.3.5	Der Debugger	142

Teil II Praktische Informatik 145

Kapitel 7 Programmiersprachen 147

7.1	Rätsel: Analytische Rätsel (4)	148
7.2	Höhere Programmiersprachen	148
7.3	Grundlagen der Programmierung	151
7.3.1	Spezifikation einer Aufgabenstellung	151
7.3.2	Der Begriff Algorithmus	152
7.3.3	Formulierung und Darstellung eines Algorithmus	152
7.3.4	Programm = Daten + Algorithmus	154
7.4	Datentypen und Operatoren in C/C++ und Java	160
7.4.1	Datentypen und Konstanten	160
7.4.2	Bezeichner	162
7.4.3	Grundlegende Operatoren	163
7.4.4	Die logischen Operatoren &&, und !	163
7.4.5	Die Shift-Operatoren << und >>	164
7.4.6	Die Bit-Operatoren &, , ^ und ~	164
7.4.7	Die Postfix- und Präfixoperatoren ++ und --	166
7.4.8	Prioritäten und Assoziativitäten der Operatoren	167
7.5	Formulierung von Algorithmen in C/C++ und Java	169
7.5.1	Sequenz	169
7.5.2	Verzweigungen mit if	169
7.5.3	Verzweigungen mit switch	175
7.5.4	for-Schleife (Schleife mit der Abfrage am Anfang)	176
7.5.5	while-Schleife (Schleife mit der Abfrage am Anfang)	183
7.5.6	do...while-Schleife (Schleife mit der Abfrage am Ende)	186
7.5.7	Abbruch von Schleifen mit break	187
7.5.8	Abbruch eines einzelnen Schleifendurchlaufs mit continue	189

7.5.9	Abbruch mehrerer geschachtelter Schleifen mit goto	189
7.5.10	Programmabbruch mit exit	190
7.5.11	Allgemeines zu Funktionen bzw. Methoden	190
7.5.12	Rekursive Funktionen bzw. rekursive Methoden	200
7.5.13	Arrays	209
7.5.14	Strings	214
7.5.15	Zufallszahlen	217
7.5.16	Argumente auf der Kommandozeile	219
7.5.17	Ausnahmen (Exceptions) in Java	220
7.5.18	Dateien	221
7.5.19	Strukturen in C/C++	230
7.6	Objektorientierte Programmierung mit Java	232
7.6.1	Meilensteine in der Softwareentwicklung	232
7.6.2	Einführung in die Objektorientierung	240
7.6.3	Klassen und Objekte	247
7.6.4	Konstruktoren	253
7.6.5	Vererbung und Polymorphismus	254
7.6.6	GUI-Programmierung in Java	263
7.7	Programmierung mit Python	275
7.7.1	Grundlagen	275
7.7.2	Einfache Datentypen	279
7.7.3	Strukturierte Datentypen (Datenstrukturen)	285
7.7.4	Operatoren	292
7.7.5	Formulierung von Algorithmen in Python	295
7.7.6	Dateien	300
7.7.7	Funktionen und Methoden	305
7.7.8	Klassen und Objekte	306
7.8	Portable GUI-Programmierung mit Qt	310
7.8.1	Allgemeines zu Qt	310
7.8.2	Grundlegende Konzepte und Konstrukte von Qt	312
7.8.3	Das Signal-Slot-Konzept von Qt	315
7.9	Programmierung paralleler Abläufe (Parallel-Programmierung)	323
7.9.1	Konzepte und HW-Architekturen für parallele Abläufe	324
7.9.2	SW-Konzepte und Erstellung paralleler Programme	326
7.9.3	Parallele Programmierung mit Threads	329
7.9.4	Parallele Programmierung mit openMP	335
7.9.5	Besondere Probleme bei paralleler Bearbeitung	348
7.9.6	Ausblick	359
7.10	Funktionale Programmierung (Scala, F#)	363
Kapitel 8 Datenstrukturen und Algorithmen		367
8.1	Rätsel: Analytische Rätsel (5)	368
8.2	Grundlegende Datenstrukturen	369
8.2.1	Allgemeine Eigenschaften von Daten	369
8.2.2	Basis-Datentypen	369
8.2.3	Datenstruktur = Daten + Operationen	369

8.2.4	Verkettete Listen	370
8.2.5	Binäre Suche in einfach verketteten Listen (Skiplisten)	383
8.2.6	Stack (Stapel)	386
8.2.7	Queue (Warteschlange)	394
8.3	Bäume	399
8.3.1	Grundlegendes zu Bäumen	399
8.3.2	Binäre Bäume	401
8.3.3	Sich selbst balancierende Binärbäume	416
8.3.4	Splay-Bäume	418
8.3.5	B-Bäume	418
8.3.6	Baumrekursion bei Bäumen mit mehr als zwei Zweigen	420
8.4	Komplexität von Algorithmen und O-Notation	431
8.4.1	Zeitaufwand	431
8.4.2	Speicherplatzbedarf	434
8.4.3	Klassifikation von Algorithmen	435
8.4.4	Die O-Notation	437
8.4.5	Wahl eines Algorithmus	442
8.4.6	Einfache Optimierungen bei der Implementierung	444
8.5	Elementare Sortieralgorithmen	447
8.5.1	Grundsätzliches zu Sortieralgorithmen	447
8.5.2	Bubble-Sort	448
8.5.3	Insert-Sort	450
8.5.4	Select-Sort	451
8.5.5	Zeitmessungen für Bubble-, Insert- und Select-Sort	452
8.5.6	Distribution Count-Sort (Bucket-Sort)	453
8.6	Shell-Sort	456
8.7	Quicksort	458
8.8	Mergesort	460
8.8.1	Rekursiver Mergesort für Arrays	460
8.8.2	Nicht-rekursiver Mergesort für Arrays	462
8.8.3	Analyse des Mergesort	463
8.8.4	Mischen von zwei sortierten Arrays	463
8.9	Backtracking	464
8.9.1	Finden in einem Labyrinth	464
8.9.2	Das Achtdamen-Problem	466
8.9.3	Rekursives Füllen von Figuren	468
8.9.4	Sudoku	468
8.9.5	Branch-and-Bound-Verfahren	469
Kapitel 9 Betriebssysteme		471
9.1	Rätsel: Überquerung einer Hängebrücke	472
9.2	Der Begriff Betriebssystem	472
9.3	Die Geschichte von Betriebssystemen	472
9.4	Grundaufgaben von Betriebssystemen	475
9.5	Aufbau und Dienste von Betriebssystemen	476
9.5.1	Schichtenaufbau	477

9.5.2	Prozesse, Threads, Scheduling	478
9.5.3	Synchronisationsmechanismen	481
9.5.4	Zeitdienste (Timer)	484
9.5.5	Speicherverwaltung	486
9.5.6	Dateiverwaltung und Dateisysteme	487
9.5.7	Geräteverwaltung und Treiber	490
9.5.8	Benutzerschnittstelle (Kommandozeile bzw. GUI)	492
9.5.9	Programmierschnittstelle (API)	494
9.6	Besonderheiten bei Embedded Systems	497

Kapitel 10 Rechnernetze und das Internet 501

10.1	Rätsel: Synthetische Rätsel (1)	502
10.2	Grundlagen der Vernetzung von Rechnern	502
10.3	Das ISO/OSI-Modell und Internet-Protokolle	503
10.4	Internet-Protokolle in Rechnernetzen	505
	10.4.1 Grundbegriffe zu TCP/IP-Netzen	505
	10.4.2 TCP/IP-Protokolle	508
10.5	Hubs, Switches, Router und Gateways	513
10.6	Grundlagen der Socket-Programmierung	513
10.7	Verteilte Anwendungen	513
10.8	Das World Wide Web (WWW)	515
	10.8.1 Komponenten und Konzepte des WWW	515
	10.8.2 Einführung in HTML	518
	10.8.3 Einführung in CSS	528
	10.8.4 Datenstrukturen zum Informationsaustausch	545
	10.8.5 Clientseitige Web-Programmierung mit JavaScript	551
	10.8.6 Serverseitige Webentwicklung	562
10.9	Gefahren durch Software	564
	10.9.1 Arten von Schadsoftware	564
	10.9.2 Pufferüberläufe (Buffer Overflows)	567

Kapitel 11 Datenbanksysteme 577

11.1	Rätsel: Synthetische Rätsel (2)	578
11.2	Grundlegendes zu Datenbanksystemen	578
	11.2.1 Aufgaben einer Datenbank	578
	11.2.2 Vorteile von Datenbanken	579
	11.2.3 Datenunabhängigkeit	580
11.3	Datenmodelle	581
	11.3.1 Das Entity-Relationship-Modell	581
	11.3.2 Das relationale Datenmodell	582
	11.3.3 Die relationale Algebra	584
11.4	Die Datenbanksprache SQL	585
	11.4.1 Datendefinition	586
	11.4.2 Einfügen, Ändern und Löschen von Datensätzen	587
	11.4.3 Anfragen mit select	588

Kapitel 12	Software Engineering	591
12.1	Rätsel: Synthetische Rätsel (3)	592
12.2	Die Software-Krise	592
12.3	Eine geeignete Software-Architektur	594
12.4	UML-Diagramme für die Modellierung	594
	12.4.1 Statische Modellierung in UML	595
	12.4.2 Dynamische Modellierung in UML	597
12.5	Modellierungsmöglichkeiten für die Software	599
12.6	Notwendigkeit von Prozessen	599
12.7	Der wichtige Prozess „Requirement Engineering“	600
	12.7.1 Das UML-Anwendungsfalldiagramm (Use Case Diagram)	601
	12.7.2 Das UML-Aktivitätsdiagramm	602
	12.7.3 Genaue Klärung der Kundenanforderungen	604
12.8	Prozessmodelle	605
	12.8.1 Schwer- und leichtgewichtige Prozessmodelle	605
	12.8.2 Das Wasserfall-Modell	605
	12.8.3 Das V-Modell	607
	12.8.4 Inkrementelle und iterative Prozessmodelle	608
	12.8.5 Agiles Vorgehen mit eXtreme Programming (XP)	610
12.9	Qualität eines Software-Produktes aus Kundensicht	612

Teil III Technische Informatik 615

Kapitel 13	Transistoren, Chips und logische Bausteine	617
13.1	Rätsel: Synthetische Rätsel (4)	618
13.2	Transistoren	618
	13.2.1 Funktionsweise und Aufbau von Transistoren	618
	13.2.2 Realisierung boolescher Funktionen mit Transistoren	620
13.3	Chips	621
	13.3.1 Geschichtliche Entwicklung	621
	13.3.2 Herstellungsprozess	622
13.4	Logische Bausteine	623
	13.4.1 Gatter	623
	13.4.2 Decoder	624
	13.4.3 Encoder	625
	13.4.4 Multiplexer (Selektor)	625
	13.4.5 Demultiplexer	628

Kapitel 14	Schaltnetze	631
14.1	Rätsel: Ein dialektisches Rätsel	632
14.2	Normalformen von Schaltfunktionen	632
	14.2.1 Disjunktive Normalform (DNF)	632
	14.2.2 Konjunktive Normalform (KNF)	633
	14.2.3 Allgemeines Verfahren beim Erstellen einer Schaltung	634

14.2.4	Schaltkreisrealisierung durch PLAs	635
14.3	Entwurf von Schaltnetzen	638
14.4	Minimierung logischer Ausdrücke	639
14.4.1	Karnaugh-Veitch-Diagramme (KV-Diagramme)	639
14.4.2	Don't Care Argumente	643
14.4.3	Quine-McCluskey-Verfahren	646
14.5	Addiernetze	652
14.5.1	Paralleladdierer	652
14.5.2	Paralleladdierer und -subtrahierer	654
14.5.3	Carry-Select-Addiernetze	655
14.5.4	Carry-Save-Addiernetze	657
14.5.5	Multiplizierer	658
14.6	Prinzipieller Aufbau einer ALU	660

Kapitel 15 Schaltwerke 663

15.1	Rätsel: Waldlauf, Schnapsgläser und mehr	664
15.2	Synchrone und asynchrone Schaltwerke	665
15.3	Schaltungen mit Delays	666
15.3.1	4-Bit-Ringzähler als synchrones Schaltwerk	666
15.3.2	Delays	667
15.3.3	Realisierung von Delays mit Flipflops	669
15.4	Zähler und Frequenzteiler	677
15.4.1	Synchroner 4-Bit-Ringzähler mit JK-Flipflops	677
15.4.2	Asynchroner 4-Bit-Ringzähler mit T-Flipflops	679
15.4.3	Synchroner BCD-Zähler (Mod-10) mit T-Flipflops	680
15.4.4	Asynchroner BCD-Zähler (Mod-10) mit JK-Flipflops	680
15.5	Schieberegister	681
15.6	Entwurf synchroner Schaltwerke mittels Automaten	683
15.6.1	Kurze Einführung in die Automatentheorie	683
15.6.2	Entwurf von Schaltwerken mit Moore- und Mealy-Automaten ...	686

Kapitel 16 Prozessorarchitekturen, Speicher und Caches 697

16.1	Rätsel: Schachbrett-Quadrate, Flickmuster, Kreuzformfirma	698
16.2	CISC und RISC	699
16.3	Pipelining (Fließbandverarbeitung)	701
16.3.1	Unterschiedliche Phasen beim Pipelining	701
16.3.2	Geschwindigkeitsgewinn beim Pipelining	703
16.3.3	Hazards beim Pipelining	705
16.4	Speicher für Prozessoren	708
16.5	Caches	711
16.5.1	Das Lokalitätsprinzip und der Cache-Controller	712
16.5.2	Der Lesezugriff	713
16.5.3	Vollassoziative und direkt abgebildete Caches	715
16.5.4	Der Schreibzugriff	718
16.6	Virtueller Speicher	720

16.6.1	Paging	721
16.6.2	Segmentierung	723

Teil IV Theoretische Informatik 725

Kapitel 17 Automatentheorie und formale Sprachen 727

17.1	Rätsel: Weg durch ein Labyrinth und um die Ecke gedacht	728
17.2	Lexikalische und syntaktische Analyse	728
17.3	Reguläre Sprachen und endliche Automaten	730
17.3.1	Alphabet, Wort und Sprache	730
17.3.2	Reguläre Ausdrücke	731
17.3.3	Endliche Automaten und reguläre Sprachen	733
17.3.4	Realisierung endlicher Automaten	735
17.3.5	lex – Ein Werkzeug für die lexikalische Analyse	736
17.4	Kontextfreie Sprachen und Kellerautomaten	740
17.4.1	Kontextfreie Grammatiken	740
17.4.2	Kellerautomaten	743
17.4.3	yacc – Ein Werkzeug für die Syntaxanalyse	746
17.4.4	lex und yacc im Zusammenspiel	750
17.4.5	Rekursion bei der Syntaxanalyse	751
17.5	Die unterschiedlichen Phasen eines Compilers	751

Kapitel 18 Berechenbarkeitstheorie 755

18.1	Rätsel: Kneipen, Ei, stehen gebliebene Uhr und Alter	756
18.2	Berechenbare Funktionen	757
18.3	Nicht berechenbare Funktionen	758
18.3.1	Das Diagonalverfahren von Cantor	758
18.3.2	Nicht durch einen Algorithmus berechenbare Funktionen	759
18.3.3	Die Church'sche Algorithmus-Definition	759
18.4	Berechenbarkeitskonzepte	760
18.4.1	Turingmaschinen	760
18.4.2	Turing-berechenbare Funktionen	763
18.4.3	Registermaschinen	763
18.4.4	GOTO- und WHILE-Programme	764
18.4.5	LOOP-Programme (FOR-Programme)	766
18.4.6	Primitive Rekursion	767
18.4.7	μ -Rekursion	770
18.4.8	Die Ackermann-Funktion	771
18.4.9	Die Church'sche These und die Chomsky-Hierarchie	773
18.5	Prinzipiell unlösbare Probleme	774
18.5.1	Entscheidbare Mengen	774
18.5.2	semi-entscheidbare Mengen (Game of Life und Halteproblem) ...	775
18.5.3	Unberechenbarkeit (Fleißiger Biber)	779

Kapitel 19	Komplexitätstheorie	783
19.1	Rätsel: Falsche Uhrzeit, Kalenderrechnen und mehr	784
19.2	Die Klasse P für praktisch lösbare Probleme	784
19.3	Nichtdeterminismus und die Klasse NP	785
19.3.1	Das SAT-Problem als erstes NP-Problem	785
19.3.2	Reduzierung auf ja/nein-Probleme mit zugehörigen Sprachen ...	786
19.3.3	Nichtdeterminismus	786
19.3.4	Die Klasse NP	787
19.4	Der Satz von Cook und NP-Vollständigkeit	789
19.4.1	Das Dreifarbenproblem als Spezialfall des SAT-Problems	789
19.4.2	NP-Vollständigkeit	790
19.4.3	P = NP?	791
19.4.4	Das 3SAT-Problem	791
19.4.5	Das Cliquesproblem	792
19.4.6	Das Rucksack- und Teilsommen-Problem	794
19.4.7	Das Hamilton-Problem	799
19.4.8	Das Problem des Handlungsreisenden	799
19.4.9	Hierarchie der NP-vollständigen Probleme	802
19.5	Approximationsalgorithmen	802

Teil V **Codes, Kompression, Kryptografie** **807**

Kapitel 20	Fehlertolerante Codes	809
20.1	Rätsel: Auf der Demo mit Bruder und Schwester	810
20.2	Motivation für fehlertolerante Codes	810
20.3	„k aus n“-Codes	810
20.4	Der Hammingabstand eines Codes	811
20.5	Eindimensionale Parity-Prüfung	813
20.6	Zweidimensionale Parity-Prüfung	814
20.7	Hamming-Codes	819
20.8	CRC-Kodierung	821

Kapitel 21	Datenkompression	825
21.1	Rätsel: Tierseuche	826
21.2	Verlustbehaftete und verlustlose Kompression	826
21.3	Codes mit variabel langen Codewörtern	826
21.4	Fano-Bedingung für Dekodierbarkeit eines Codes	827
21.5	Laufängenkodierung („run-length encoding“)	828
21.6	Shannon-Fano-Kodierung	829
21.7	Huffman-Kodierung	829
21.8	Arithmetische Kodierung	833
21.9	Lempel-Ziv-Kodierungen	836

21.9.1	Der LZ77-Algorithmus	838
21.9.2	Der LZSS-Algorithmus	839
21.9.3	Der LZ78-Algorithmus	840
21.9.4	Der LZW-Algorithmus	841
21.9.5	Varianten der Lempel-Ziv-Kodierung	845
Kapitel 22 Kryptografie		847
22.1	Rätsel: Weinflasche und Erben von Weinfässern	848
22.2	Allgemeines zu Kryptosystemen	848
22.3	Einfache Verschlüsselungsmethoden	848
22.3.1	Cäsar-Chiffre	848
22.3.2	Chiffre mit eigener Zuordnungstabelle	849
22.4	Vigenère-Verschlüsselungsmethoden	849
22.5	Verschlüsselung mittels Zufallsfolgen	850
22.6	Kryptosysteme mit öffentlichen Schlüsseln	852
22.6.1	Eigenschaften von Public-Key-Systemen	852
22.6.2	Der Satz von Euler	853
22.6.3	Schlüsselerzeugung beim RSA-Algorithmus	854
22.6.4	Ver- und Entschlüsselung mit dem RSA-Algorithmus	856
Weiterführende Literatur		859
Sachregister		865