



Sascha Kersken

42 69 74 74 65 20  
6D 61 69 6C 65 6E 20 53 69  
65 20 75 6E 74 65 72 20  
69 63  
47  
6F  
20 2D 20 54 68 65 20 48  
65 58 2D 46 69 6C 65 73  
22 20 61 6E 20 73 74 65 70  
68 61 6E 2E 6D 61 74 74 65  
73 63 68 65 63 6B 40 67  
61 6C 69 6C 65 6F 2D 70 72

# IT-Handbuch für Fachinformatiker

Der Ausbildungsbegleiter

IT-Handbuch für  
Fachinformatiker

- EDV-Grundlagen, Netzwerktechnik, Programmierung
- Praxisorientiertes Lehr- und Nachschlagewerk
- Für Fachinformatiker der Bereiche Anwendungs-entwicklung und Systemintegration

3., aktualisierte und erweiterte Auflage

Galileo Computing 

## Auf einen Blick

	<b>Geleitwort des Fachgutachters</b> .....	15
	<b>Vorwort</b> .....	17
<b>1</b>	<b>Einführung</b> .....	25
<b>2</b>	<b>Mathematische und technische Grundlagen</b> .....	57
<b>3</b>	<b>Hardware</b> .....	107
<b>4</b>	<b>Netzwerkgrundlagen</b> .....	165
<b>5</b>	<b>Betriebssystemgrundlagen</b> .....	273
<b>6</b>	<b>Windows</b> .....	305
<b>7</b>	<b>Linux und UNIX</b> .....	351
<b>8</b>	<b>Grundlagen der Programmierung</b> .....	429
<b>9</b>	<b>Konzepte der Programmierung</b> .....	515
<b>10</b>	<b>Software-Engineering</b> .....	617
<b>11</b>	<b>Datenbanken</b> .....	653
<b>12</b>	<b>Server für Webanwendungen</b> .....	703
<b>13</b>	<b>Weitere Internet-Serverdienste</b> .....	747
<b>14</b>	<b>XML</b> .....	769
<b>15</b>	<b>Weitere Datei- und Datenformate</b> .....	811
<b>16</b>	<b>Webseitenerstellung mit (X)HTML und CSS</b> .....	839
<b>17</b>	<b>Webserveranwendungen</b> .....	893
<b>18</b>	<b>JavaScript und Ajax</b> .....	939
<b>19</b>	<b>Computer- und Netzwerksicherheit</b> .....	1001
<b>A</b>	<b>Glossar</b> .....	1031
<b>B</b>	<b>Zweisprachige Wortliste</b> .....	1047
<b>C</b>	<b>Kommentiertes Literatur- und Linkverzeichnis</b> .....	1053

# Inhalt

Geleitwort des Fachgutachters .....	15
Vorwort .....	17
<b>1 Einführung .....</b>	<b>25</b>
1.1 Informationstechnik, Informatik und EDV .....	25
1.1.1 Fachrichtungen der Informatik .....	26
1.1.2 Überblick über die IT-Ausbildung .....	27
1.2 Die Geschichte der Rechenmaschinen und Computer .....	33
1.2.1 Die Vorgeschichte .....	35
1.2.2 Die Entwicklung der elektronischen Rechner .....	36
1.2.3 Entwicklung der Programmiersprachen .....	45
1.3 Digitale Speicherung und Verarbeitung von Informationen .....	51
1.3.1 Digitale Bilddaten .....	53
1.3.2 Digitale Audiodaten .....	54
1.3.3 Digitale Speicherung von Text .....	55
1.4 Zusammenfassung .....	56
<b>2 Mathematische und technische Grundlagen .....</b>	<b>57</b>
2.1 Einführung in die Logik .....	57
2.1.1 Aussagen .....	58
2.1.2 Aussageformen .....	59
2.1.3 Logische Verknüpfungen .....	60
2.1.4 Mengenoperationen .....	67
2.1.5 Weitere wichtige Berechnungsverfahren .....	70
2.2 Informationsspeicherung im Computer .....	72
2.2.1 Zahlensysteme .....	72
2.2.2 Bits und Bytes .....	78
2.3 Elektronische Grundlagen .....	82
2.3.1 Einfache Schaltungen .....	82
2.3.2 Zusammengesetzte Schaltungen .....	86
2.4 Automatentheorien und -simulationen .....	90
2.4.1 Algorithmen .....	90
2.4.2 Die Turing-Maschine .....	95
2.4.3 Der virtuelle Prozessor .....	99
2.5 Zusammenfassung .....	105

<b>3</b>	<b>Hardware .....</b>	<b>107</b>
3.1	Grundlagen .....	107
3.2	Die Zentraleinheit .....	111
3.2.1	Aufbau und Aufgaben des Prozessors .....	112
3.2.2	Der Arbeitsspeicher .....	122
3.2.3	Das BIOS .....	124
3.2.4	Bus- und Anschlusssysteme .....	130
3.3	Die Peripherie .....	139
3.3.1	Massenspeicher .....	140
3.3.2	Eingabegeräte .....	152
3.3.3	Ausgabegeräte .....	156
3.3.4	Sound-Hardware .....	162
3.4	Zusammenfassung .....	163
<b>4</b>	<b>Netzwerkgrundlagen .....</b>	<b>165</b>
4.1	Grundlagen .....	165
4.1.1	Was ist ein Netzwerk? .....	165
4.1.2	Entstehung der Netzwerke .....	167
4.2	Funktionsebenen von Netzwerken .....	172
4.2.1	Das OSI-Referenzmodell .....	172
4.2.2	Das Schichtenmodell der Internetprotokolle .....	175
4.2.3	Netzwerkcommunication über die Schichten eines Schichtenmodells .....	178
4.3	Klassifizierung von Netzwerken .....	181
4.3.1	Die Reichweite des Netzwerks .....	182
4.3.2	Die Netzwerktopologie .....	183
4.3.3	Der Zentralisierungsgrad des Netzwerks .....	184
4.4	Netzwerkarten, Netzwerkkabel und Netzzugangsverfahren .....	191
4.4.1	Die verschiedenen Ethernet-Standards .....	192
4.4.2	Token Ring .....	197
4.4.3	Drahtlose Netze .....	198
4.4.4	Sonstige Zugangsarten .....	203
4.5	Datenfernübertragung .....	204
4.5.1	Netzwerkzugang per Modem (analoge Telefonleitung) .....	205
4.5.2	ISDN .....	207
4.5.3	DSL-Dienste .....	209
4.6	Die TCP/IP-Protokollfamilie .....	211
4.6.1	Netzzugang in TCP/IP-Netzwerken .....	212
4.6.2	IP-Adressen, Datagramme und Routing .....	213
4.6.3	Transportprotokolle .....	240

4.6.4	Das Domain Name System (DNS) .....	246
4.6.5	Verschiedene Internet-Anwendungsprotokolle .....	250
4.7	Andere Protokollstapel .....	266
4.7.1	Die AppleTalk-Protokollfamilie .....	266
4.7.2	Novell IPX/SPX .....	269
4.7.3	NetBEUI/SMB .....	270
4.8	Zusammenfassung .....	271
<b>5</b>	<b>Betriebssystemgrundlagen .....</b>	<b>273</b>
5.1	Entwicklung der Betriebssysteme .....	274
5.1.1	Die Geschichte von UNIX .....	276
5.1.2	PC-Betriebssysteme .....	277
5.2	Aufgaben und Konzepte .....	282
5.2.1	Allgemeiner Aufbau von Betriebssystemen .....	282
5.2.2	Prozessverwaltung .....	289
5.2.3	Speicherverwaltung .....	294
5.2.4	Dateisysteme .....	296
5.3	Zusammenfassung .....	303
<b>6</b>	<b>Windows .....</b>	<b>305</b>
6.1	Allgemeine Informationen .....	305
6.1.1	Die verschiedenen Windows-Versionen .....	305
6.1.2	Windows-Dateisysteme .....	309
6.2	Windows im Einsatz .....	310
6.2.1	Die Windows-Benutzeroberfläche .....	310
6.2.2	Die Windows-Konsole .....	319
6.2.3	Die Windows PowerShell .....	323
6.2.4	Windows-Konfiguration .....	334
6.3	Windows-Netzwerkkonfiguration .....	341
6.3.1	Allgemeine Einstellungen .....	341
6.3.2	TCP/IP-Dienstprogramme .....	342
6.3.3	Datei- und Druckserver unter Windows .....	346
6.3.4	Windows-Server .....	347
6.4	Zusammenfassung .....	349
<b>7</b>	<b>Linux und UNIX .....</b>	<b>351</b>
7.1	Arbeiten mit der Shell .....	353
7.1.1	Booten und Login .....	353
7.1.2	Virtuelle Terminals .....	356
7.1.3	Grundfunktionen der Shell .....	357
7.1.4	Hilfefunktionen .....	363
7.1.5	Pipes und Ein-/Ausgabeumleitung .....	366

7.1.6	Die wichtigsten Systembefehle .....	369
7.2	Konfigurations- und Administrationsaufgaben .....	383
7.2.1	Syslog und Logdateien .....	383
7.2.2	Programme automatisch starten .....	384
7.2.3	Software installieren .....	386
7.3	Automatisierung .....	388
7.3.1	Shell-Skripte .....	389
7.3.2	Weitere Hilfsmittel .....	392
7.4	Editoren .....	395
7.4.1	vi .....	395
7.4.2	Emacs .....	398
7.5	Grafische Benutzeroberflächen .....	401
7.5.1	Der X-Server .....	401
7.5.2	Desktops .....	403
7.6	Netzwerkconfiguration unter Linux .....	407
7.6.1	Grundeinstellungen .....	407
7.6.2	TCP/IP-Dienstprogramme .....	409
7.6.3	Datei- und Druckserver unter Linux .....	409
7.7	Mac OS X .....	415
7.7.1	Mit Aqua arbeiten .....	417
7.7.2	Systemkonfiguration .....	421
7.7.3	Mac OS X-Netzwerkconfiguration .....	423
7.7.4	Serverdienste unter Mac OS X .....	425
7.8	Zusammenfassung .....	426
<b>8</b>	<b>Grundlagen der Programmierung .....</b>	<b>429</b>
8.1	Die Programmiersprache C .....	430
8.1.1	Das erste Beispiel .....	431
8.1.2	Elemente der Sprache C .....	434
8.1.3	Die C-Standardbibliothek .....	452
8.2	Java .....	457
8.2.1	Grundlegende Elemente der Sprache Java .....	459
8.2.2	Objektorientierte Programmierung mit Java .....	463
8.2.3	Dateizugriffe in Java .....	469
8.3	Perl .....	471
8.3.1	Das erste Beispiel .....	472
8.3.2	Elemente der Sprache Perl .....	473
8.4	Ruby .....	484
8.4.1	Das erste Beispiel .....	485
8.4.2	Ruby-Grundelemente .....	487
8.4.3	Objektorientierung in Ruby .....	501
8.5	Zusammenfassung .....	512

<b>9</b>	<b>Konzepte der Programmierung .....</b>	<b>515</b>
9.1	Algorithmen und Datenstrukturen .....	515
9.1.1	Ein einfaches Praxisbeispiel .....	515
9.1.2	Sortier-Algorithmen .....	518
9.1.3	Such-Algorithmen .....	522
9.1.4	Ausgewählte Datenstrukturen .....	524
9.2	Reguläre Ausdrücke .....	529
9.2.1	Muster für reguläre Ausdrücke .....	530
9.2.2	Programmierung mit regulären Ausdrücken .....	534
9.3	Systemnahe Programmierung .....	537
9.3.1	Prozesse und Pipes .....	537
9.3.2	Threads .....	541
9.4	Einführung in die Netzwerkprogrammierung .....	545
9.4.1	Die Berkeley Socket API .....	545
9.4.2	Ein praktisches Beispiel .....	552
9.4.3	Ein Ruby-Webserver .....	556
9.5	Verteilte Anwendungen mit J2EE .....	572
9.5.1	Enterprise Java Beans (EJB) .....	573
9.5.2	Java Servlets .....	578
9.5.3	Web Services .....	579
9.6	GUI- und Grafikprogrammierung .....	582
9.6.1	Zeichnungen und Grafiken erstellen .....	583
9.6.2	Animation .....	589
9.6.3	Programmierung fensterbasierter Anwendungen .....	592
9.6.4	Java-Applets .....	606
9.7	Die Entwicklungsumgebung Eclipse .....	608
9.7.1	Überblick .....	608
9.7.2	Java-Entwicklung mit Eclipse .....	609
9.8	Zusammenfassung .....	614
<b>10</b>	<b>Software-Engineering .....</b>	<b>617</b>
10.1	Überblick .....	618
10.1.1	Der Entwicklungszyklus .....	619
10.1.2	Planung und Analyse .....	620
10.1.3	Entwurf .....	626
10.1.4	Implementierung und Test .....	627
10.1.5	Dokumentation .....	628
10.1.6	Konkrete Entwicklungsverfahren .....	629
10.2	Werkzeuge .....	632
10.2.1	UML .....	633
10.2.2	Entwurfsmuster .....	639

10.2.3	Unit-Tests .....	646
10.3	Zusammenfassung .....	651

**11 Datenbanken ..... 653**

11.1	Die verschiedenen Datenbanktypen .....	654
11.1.1	Einzeltabellendatenbanken .....	655
11.1.2	Relationale Datenbanken .....	657
11.1.3	Objektorientierte Datenbanken .....	664
11.2	MySQL – ein konkretes DBMS .....	667
11.2.1	MySQL installieren und konfigurieren .....	667
11.2.2	Erste Schritte mit dem mysql-Client .....	671
11.3	SQL-Abfragen .....	671
11.3.1	Datenbanken und Tabellen erzeugen .....	672
11.3.2	Auswahlabfragen .....	676
11.3.3	Einfüge-, Lösch- und Änderungsabfragen .....	680
11.3.4	Transaktionen .....	682
11.4	MySQL-Administration .....	683
11.4.1	mysqladmin .....	683
11.4.2	Benutzerverwaltung .....	684
11.4.3	Import und Export von Daten, Backups .....	689
11.4.4	Konfigurationsdateien .....	692
11.4.5	Logdateien .....	693
11.4.6	Replikation .....	695
11.5	Grundlagen der Datenbankprogrammierung .....	697
11.6	Zusammenfassung .....	700

**12 Server für Webanwendungen ..... 703**

12.1	HTTP im Überblick .....	703
12.1.1	Ablauf der HTTP-Kommunikation .....	704
12.1.2	HTTP-Statuscodes .....	707
12.1.3	HTTP-Header .....	710
12.2	Der Webserver Apache .....	714
12.2.1	Apache im Überblick .....	714
12.2.2	Apache-Module .....	716
12.2.3	Apache installieren .....	717
12.2.4	Apache-Konfiguration .....	721
12.3	PHP installieren und einrichten .....	734
12.3.1	Installation .....	734
12.3.2	Die PHP-Konfigurationsdatei php.ini .....	738
12.3.3	phpMyAdmin einrichten .....	741
12.4	Zusammenfassung .....	745

<b>13</b>	<b>Weitere Internet-Serverdienste .....</b>	<b>747</b>
13.1	Namens- und Verzeichnisdienste .....	747
13.1.1	Der DNS-Server BIND .....	747
13.1.2	Der Verzeichnisdienst OpenLDAP .....	753
13.2	Sonstige Server .....	763
13.2.1	vsftpd, ein FTP-Server .....	763
13.2.2	inetd und xinetd .....	764
13.3	Zusammenfassung .....	768
<b>14</b>	<b>XML .....</b>	<b>769</b>
14.1	Der Aufbau von XML-Dokumenten .....	771
14.1.1	Die grundlegenden Bestandteile von XML-Dokumenten .....	771
14.1.2	Wohlgeformtheit .....	779
14.2	DTDs und XML Schema .....	781
14.2.1	Document Type Definitions (DTDs) .....	781
14.2.2	Namensräume .....	792
14.2.3	XML Schema .....	793
14.3	XSLT .....	796
14.4	Grundlagen der XML-Programmierung .....	799
14.4.1	SAX .....	800
14.4.2	DOM .....	807
14.5	Zusammenfassung .....	809
<b>15</b>	<b>Weitere Datei- und Datenformate .....</b>	<b>811</b>
15.1	Textdateien und Zeichensätze .....	811
15.1.1	Das Problem des Zeilenumbruchs .....	812
15.1.2	Zeichensätze .....	814
15.1.3	Textbasierte Dateiformate .....	821
15.2	Binäre Dateiformate .....	824
15.2.1	Bilddateiformate .....	827
15.2.2	Multimedia-Dateiformate .....	831
15.2.3	Archivdateien verwenden .....	833
15.3	Zusammenfassung .....	836
<b>16</b>	<b>Webseitenerstellung mit (X)HTML und CSS .....</b>	<b>839</b>
16.1	HTML und XHTML .....	840
16.1.1	Die Grundstruktur von HTML-Dokumenten .....	841
16.1.2	Textstrukturierung und Textformatierung .....	843
16.1.3	Listen und Aufzählungen .....	849
16.1.4	Hyperlinks .....	852

16.1.5	Bilder in Webseiten einbetten .....	857
16.1.6	Tabellen .....	860
16.1.7	Formulare .....	869
16.1.8	Einbetten von Multimedia-Dateien .....	875
16.1.9	Meta-Tags und Suchmaschinen .....	877
16.2	Cascading Stylesheets (CSS) .....	880
16.2.1	Platzieren von Stylesheets .....	881
16.2.2	Stylesheet-Wertangaben .....	883
16.2.3	Stylesheet-Eigenschaften .....	885
16.2.4	Layer erzeugen und positionieren .....	888
16.3	Zusammenfassung .....	891

## **17 Webserveranwendungen ..... 893**

17.1	PHP .....	893
17.1.1	Sprachgrundlagen .....	894
17.1.2	Klassen und Objekte .....	904
17.1.3	Webspezifische Funktionen .....	909
17.1.4	Zugriff auf MySQL-Datenbanken .....	914
17.2	Ruby on Rails .....	920
17.2.1	Grundlagen .....	921
17.2.2	Ein Praxisbeispiel .....	922
17.3	Weitere Technologien im Überblick .....	933
17.3.1	Content-Management, Weblogs & Co. ....	934
17.3.2	Blogs und Wikis .....	935
17.4	Zusammenfassung .....	937

## **18 JavaScript und Ajax ..... 939**

18.1	JavaScript .....	939
18.2	JavaScript im HTML-Dokument .....	940
18.2.1	Erstes Beispiel: Ausgabe ins Dokument .....	941
18.3	Formulare und Event-Handler .....	945
18.3.1	Erstes Beispiel .....	945
18.3.2	Zugriff auf Formulare und ihre Elemente .....	946
18.4	Datums- und Uhrzeit-Funktionen .....	954
18.4.1	Datums- und Uhrzeit-Methoden .....	955
18.4.2	Timeout – die JavaScript-»Stoppuhr« .....	956
18.5	Manipulation von Bildern .....	957
18.5.1	Erstes Beispiel: Austauschen eines Bildes auf Knopfdruck .....	957
18.5.2	Vorausladen von Bildern .....	959
18.5.3	Eine gut funktionierende Rollover-Lösung .....	959
18.6	Browser- und Fenster-Optionen .....	964
18.6.1	Browser-Eigenschaften .....	964

18.6.2	Automatische Hyperlinks – History und Location .....	966
18.6.3	Neue Browserfenster öffnen .....	966
18.7	DHTML und DOM .....	971
18.7.1	W3C-DOM im Überblick .....	971
18.7.2	Eine DOM-Baum-Anzeige .....	974
18.7.3	DOM-Anwendung in der Praxis .....	977
18.7.4	Dokumentinhalte verändern und austauschen .....	979
18.8	Ajax .....	981
18.8.1	Die erste Ajax-Anwendung .....	982
18.8.2	Datenaustauschformate: XML und JSON .....	988
18.8.3	Größeres Beispiel: Eine interaktive Länderliste .....	989
18.9	Zusammenfassung .....	999

## 19 Computer- und Netzwerksicherheit ..... 1001

19.1	PC-Gefahren .....	1002
19.1.1	Viren und Würmer .....	1002
19.1.2	Trojaner und Backdoors .....	1007
19.1.3	Weitere Schädlinge .....	1009
19.2	Netzwerk- und Serversicherheit .....	1014
19.2.1	Servergefahren .....	1014
19.2.2	Wichtige Gegenmaßnahmen .....	1017
19.2.3	Kryptografie .....	1023
19.3	Zusammenfassung .....	1025

## Anhang ..... 1029

A	Glossar .....	1031
B	Zweisprachige Wortliste .....	1047
B.1	Englisch-Deutsch .....	1047
B.2	Deutsch-Englisch .....	1049
C	Kommentiertes Literatur- und Linkverzeichnis .....	1053
C.1	Allgemeine Einführungen und Überblicke .....	1053
C.2	Mathematische und technische Grundlagen .....	1054
C.3	Hardware .....	1054
C.4	Netzwerktechnik .....	1055
C.5	Betriebssystemgrundlagen .....	1055
C.6	Windows .....	1056
C.7	Linux und UNIX .....	1056
C.8	Grundlagen der Programmierung .....	1057
C.9	Konzepte der Programmierung .....	1059
C.10	Software-Engineering .....	1059
C.11	Datenbanken .....	1061
C.12	Server für Webanwendungen .....	1061

## Inhalt

C.13	Weitere Internet-Serverdienste .....	1062
C.14	XML .....	1062
C.15	Webseitenerstellung mit HTML und CSS .....	1062
C.16	Webserveranwendungen .....	1063
C.17	JavaScript und Ajax .....	1064
C.18	Computer- und Netzwerksicherheit .....	1064
Index	.....	1065

## Geleitwort des Fachgutachters

Gute IT- Bücher sind – gemessen an der Anzahl der veröffentlichten Werke – eher selten. Gute Bücher für den Fachinformatiker – sogar eine Rarität. Umso erfreulicher ist das vorliegende Buch von Sascha Kersken, das gleich in mehrfacher Hinsicht ein Glücksfall ist. Fachinformatiker in der Ausbildung stehen immer wieder vor der Schwierigkeit, dass sie in einem sehr spezifischen IT-Umfeld arbeiten und dort oft sehr tief eindringen müssen. Was dann allerdings fehlt und von den Berufsschulen nicht immer vermittelt werden kann, ist ein allgemeiner Überblick, der zukünftigen Fachinformatikern eine Orientierung an die Hand gibt, ohne sich im Allgemeinen zu verlieren.

Während meiner mehrjährigen Zeit als Dozent und Ausbilder für Fachinformatiker in den Richtungen Systemintegration und Anwendungsentwicklung und in meiner ehrenamtlichen Tätigkeit in einem IHK-Prüfungsausschuss habe ich nur wenige kompetente Bücher kennengelernt, die speziell für die Ausbildung zum Fachinformatiker geschrieben wurden. Manche dieser Bücher sind reine Formel- und Tabellensammlungen, andere, im Taschenbuchformat, behandeln die Themen oftmals eher oberflächlich.

In dieser neuen Auflage ist Sascha Kersken das Kunststück gelungen, wesentliche Elemente der Ausbildung kompetent in einem Buch zu vereinen. In dem Buch werden zukünftigen Fachinformatikern Fakten nicht einfach tabellarisch angeboten, sondern anschaulich erklärt. Die technischen und mathematischen Grundlagen, wichtige Konzepte der Betriebssysteme und Programmiersprachen werden ebenso wie Grundlagen der Netzwerke, IT-Sicherheit, Datenbanken, XML und Webanwendungen behandelt. Dabei versteht es der Autor, komplexe und schwierige Themen, wie beispielsweise die Bedienung klassischer UNIX-Editoren oder den Umgang mit dem Subnetting, didaktisch leicht verständlich zu vermitteln.

Was mir besonders gut gefällt ist die Tatsache, dass Beispiele aus zahlreichen Programmiersprachen verwendet werden und dabei der Schwerpunkt nicht auf die einzelne Sprache, sondern auf die Prinzipien und Techniken der Programmierung gesetzt wird. Hier verbirgt sich in den anwendungsorientierten Kapiteln ein auf die Zielgruppe zugeschnittenes Werk, das Programmieren konzeptorientiert vermittelt und sich dabei nicht an eine spezielle Sprache bindet. Schon aus diesem Grund ist dieses Buch sehr empfehlenswert, nicht nur für Anwendungsentwickler. Grundkenntnisse der Programmtechniken sind für alle Fachinformatiker notwendig und prüfungsrelevant.

Dass sich der in der Open-Source-Szene beheimatete Autor in seinem Werk besonders auf lizenzfreie Server, Betriebssysteme und Programmierumgebungen stützt, ist ein Umstand, der sich für die Zielgruppe besonders günstig auswirkt. Gleich ob ein angehender Student, ein Schüler, der sich auf seinen zukünftigen Beruf vorbereiten möchte, oder ein Auszubildender, der über den Rahmen seines Betriebes hinaus Kenntnisse erlangen möchte, – die Beispiele lassen sich mit wenig Aufwand und ohne teure Lizenzen mithilfe Open Source oder anderer frei verfügbarer Software nachvollziehen. Darüber hinaus sind Kenntnisse von Open Source im modernen Berufsbild des Fachinformatikers unverzichtbar geworden und werden in Zukunft vermutlich noch wichtiger werden.

Dieses Buch sollte meiner Ansicht nach in keinem Regal eines zukünftigen Fachinformatikers fehlen. Ebenso ist es für Informatik-Schüler in Gymnasien, Informatik-Studenten in den ersten Semestern oder sonstige IT-Spezialisten ausgesprochen empfehlenswert.

**Dipl. Biol. G. A. Wilm**

Geschäftsführer der degaw IT-Beratungen,  
langjähriger IT-Dozent für Fachinformatiker,  
Mitglied im Prüfungsausschuss für Fachinformatiker  
Richtung Systemintegration in der IHK Nord Westfalen

*Computer der Zukunft werden vielleicht einmal nicht mehr  
als 1,5 Tonnen wiegen.  
– Popular Mechanics, 1949*

## Vorwort

Natürlich wiegt der durchschnittliche Computer heute nur noch wenige Kilogramm, und ist dennoch einige Tausendmal leistungsfähiger als die Ungetüme der 1940er-Jahre. Trotzdem zeigt dieses Zitat aus der »guten, alten Zeit« eines ganz deutlich: die einzige Konstante im Bereich der Informationstechnik ist der Wandel. Gerade deshalb ist es so wichtig, in der IT-Ausbildung einen gesunden Mittelweg zwischen der Vermittlung zeitloser, allgemeiner Konzepte und der Erarbeitung tagesaktueller Technologien zu finden.

Die vorige Auflage des vorliegenden Buches erschien vor gut zweieinhalb Jahren. Dies ist, gemessen am Lebenszyklus der PC-Hardwaregenerationen und Softwareversionen, eine recht lange Zeit. So ist es kein Wunder, dass die meisten technischen Detailinformationen in diesem Buch angepasst werden mussten. Einige der hier vermittelten Grundkonzepte gelten dagegen nahezu unverändert seit den 1970er-Jahren; die mathematischen Grundlagen, auf die die Informatik aufbaut, sind größtenteils sogar bereits seit der Antike bekannt.

Dennoch ist diese dritte Auflage mehr als ein einfaches Versionsupdate: Dank der tatkräftigen Hilfe des technischen Gutachters Georg Wilm, der sich hervorragend im Bereich der IT-Ausbildung auskennt und auch das Geleitwort geschrieben hat, wurde jedes Kapitel einer genauen Prüfung hinsichtlich der Eignung für die Ausbildung zum Fachinformatiker und verwandte Berufe unterzogen. So wurde vieles ergänzt, manches aktualisiert, und einige Themen mussten aus Platzgründen weichen.

Ich hoffe, dass die Mischung von Theorie- und Praxisthemen, von allgemeinen Konzepten und aktuellen Technologien, noch besser gelungen ist als in den bisherigen Auflagen, und wünsche Ihnen viel Spaß beim Lesen und viel Erfolg bei Ihrer IT-Ausbildung.

**Sascha Kersken**  
Köln

## Worum geht es in diesem Buch?

Dieses Buch beschreibt die wesentlichen Ausbildungsinhalte im Fachkunde-Unterricht in den Ausbildungsberufen Fachinformatiker Anwendungsentwicklung und Systemintegration, IT-Systemelektroniker, IT-Systemkaufleute und Informatikkaufleute. Anders als herkömmliche Bücher handelt es sich jedoch nicht um eine Tabellensammlung zum Auswendiglernen, sondern um praxisorientierte Anleitungen, die Sie weit über Ihre Ausbildung hinaus bei Ihrer alltäglichen Arbeit mit Computersystemen einsetzen können.

Sie lernen in den nachfolgenden Kapiteln zahlreiche Geräte, Betriebssysteme, Programmiersprachen und Anwendungsprogramme kennen und erfahren viel Wissenswertes über Netzwerke und das Internet. Bei allen Themen habe ich versucht, die Balance zwischen Theorie und Praxis zu wahren: Es werden weder die technischen und theoretischen Details verschwiegen, wie in zahlreichen Büchern für absolute Einsteiger, noch kommen die praktischen Anwendungsbeispiele zu kurz, was in manchen akademischen Lehrwerken der Fall ist.

Soweit es möglich ist, habe ich versucht, konkrete Software aus dem Open-Source-Bereich auszuwählen. Zum Einen, weil ich selbst seit Jahren Open-Source-Software einsetze und entwickle, vor allem aber auch, weil Sie auf diese Weise fast jedes in diesem Buch behandelte Programm kostenlos und ohne jegliche Einschränkungen herunterladen, installieren und benutzen können.

- [»] Die Ausbildung zum Fachinformatiker oder für andere IT-Berufe enthält neben den fachspezifischen Inhalten auch Aspekte wie Wirtschafts- und Sozialwissenschaften, Rechtskunde oder Betriebsorganisation. Solche Themen können in diesem Buch nicht berücksichtigt werden. Für das Fach Technisches Englisch finden Sie in Anhang B immerhin eine Stichwortliste in beiden Richtungen.

## Was ist neu in dieser Auflage?

Die eingangs erwähnte Gutachtertätigkeit hat Aufbau und Inhalte dieses Buches stark beeinflusst; mehr denn je wird es den Anforderungen der Fachinformatiker-Ausbildung gerecht. Hier die wichtigsten Neuerungen im Überblick:

- ▶ Grundlegende Informationen zur Netzwerktechnik wurden weit vorgezogen; sie wird gleich nach der Hardware und noch vor den Betriebssystemgrundlagen behandelt, da sie einer der wichtigsten Aspekte der modernen Betriebssysteme ist.
- ▶ Das bisherige Kapitel »Betriebssysteme« wurde in die drei Einzelkapitel Grundlagen, Windows sowie Linux und UNIX unterteilt; vor allem die Praxisteile wurden stark erweitert:
  - ▶ Mehr technische Details des Betriebssystems Windows, unter anderem ein Einstieg in die neue PowerShell

- ▶ Noch mehr Informationen zur UNIX-Administration, einschließlich Cronjobs
- ▶ Detaillierte Praxisanleitungen zur Netzwerkkonfiguration in beiden Betriebssystemfamilien
- ▶ Als vierte Programmiersprache neben C, Java und Perl wird die objektorientierte Skriptsprache Ruby ausführlich behandelt.
- ▶ Neben der praktischen Datenbankverwendung wird nun auch die Administration eines Datenbankservers beschrieben, und zwar ebenfalls am Beispiel von MySQL.
- ▶ Die Informationen über Serverdienste wie Apache wurden stark erweitert; neue wie etwa der Verzeichnisdienst OpenLDAP sind hinzugekommen.
- ▶ Zusätzlich zur bewährten PHP-Einführung lernen Sie eine zweite Technologie für Webanwendungen kennen: das Web-Framework Ruby on Rails.
- ▶ Das JavaScript-Kapitel wurde um einen ausführlichen Praxiseinstieg in Ajax ergänzt.

Natürlich kommen so viele Erweiterungen nicht ohne Kompensation aus; drei Kapitel aus der Voraufgabe, die sich mit Mediengestaltungsthemen beschäftigten, wurden ganz entfernt, auch der Themenblock Webdesign einschließlich Flash ist bis auf eine eher technische HTML- und CSS-Übersicht weggefallen. Ich werde diese Inhalte nach und nach im Web verfügbar machen; behalten Sie die Webseite zum Buch (<http://www.galileocomputing.de/1234>) im Blick. [«]

## Kapitelübersicht

Die einzelnen Kapitel dieses Buches widmen sich den folgenden Themen:

- ▶ **Kapitel 1, *Einführung***, behandelt die Geschichte und grundlegende Funktionsweise des Computers. Außerdem werden einige wichtige Grundlagen der Informationstechnik erläutert; sie bilden die Voraussetzung für das Verständnis späterer Kapitel.
- ▶ In **Kapitel 2, *Mathematische und technische Grundlagen***, werden zunächst die mathematischen und logischen Prinzipien erläutert, auf denen der Computer basiert. Anschließend lernen Sie die wichtigsten elektrotechnischen Grundbausteine kennen. Abgerundet wird das Kapitel durch die Vorstellung von Automatentheorien und Rechnersimulationen.
- ▶ **Kapitel 3, *Hardware***, beschäftigt sich mit den diversen Bauteilen, aus denen ein Computer besteht, sowie mit zahlreichen wichtigen Peripheriegeräten. Sie erfahren nicht nur die wichtigsten technischen Details über Elemente wie den Mikroprozessor, verschiedene Laufwerke und Datenträger oder andere Ein- und Ausgabegeräte, sondern es werden auch praktische Informationen vermittelt. Zum Beispiel werden Einbau und Anschluss von Komponenten oder das BIOS-Setup angesprochen.

- ▶ In **Kapitel 4**, *Netzwerktechnik*, wird die Entwicklung der Netzwerke und des Internets beschrieben, und Sie erhalten eine Einführung in die Begriffswelt der Netzwerke. Beispielsweise werden Schichtenmodelle und Netzwerkarchitekturen vorgestellt. Danach wird die genaue Funktionsweise verschiedener Arten von Netzwerkkarten und -anschlüssen erläutert; außerdem werden diverse Netzwerkprotokolle beschrieben. Den Schwerpunkt bildet die TCP/IP-Protokollfamilie, die für das Internet entwickelt wurde und inzwischen der wichtigste Kommunikationsstandard für alle Arten von Netzwerken ist.
- ▶ In **Kapitel 5**, *Betriebssystemgrundlagen*, werden allgemeine Konzepte des Betriebssystemaufbaus erklärt, zum Beispiel die Verwaltung von Prozessen, das Speichermanagement oder die Dateiverwaltung.
- ▶ **Kapitel 6**, *Windows*, beschreibt die theoretischen und praktischen Grundlagen des verbreitetsten PC-Betriebssystems der Welt; den Schwerpunkt bildet die aktuelle Version Vista. Neben der grafischen Benutzeroberfläche wird auch die Arbeit mit der klassischen Windows-Kommandozeile sowie mit der neuen Windows PowerShell behandelt. Den Abschluss bildet ein Einstieg in die Windows-Konfiguration.
- ▶ In **Kapitel 7**, *Linux und UNIX*, wird eine zweite Betriebssystemfamilie praktisch beleuchtet. Schwerpunktmäßig wird die Arbeit mit der UNIX-Shell bash und den zahlreichen Systemprogrammen behandelt; daneben werden die Grundlagen der Administration und des Umgangs mit klassischen UNIX-Texteditoren beschrieben. Es geht vornehmlich um Linux; im letzten Abschnitt wird jedoch auch Apples Mac OS X behandelt.
- ▶ In **Kapitel 8**, *Grundlagen der Programmierung*, werden vier verschiedene wichtige Programmiersprachen eingeführt, die unterschiedliche Entwicklungsstufen und Aspekte des Programmierens abdecken: C, Java, Perl und Ruby.
- ▶ **Kapitel 9**, *Konzepte der Programmierung*, baut auf diese Grundlagen auf und erläutert verschiedene Aspekte, die die Programmierung in der Praxis ausmachen: Algorithmen und Datenstrukturen, Reguläre Ausdrücke (mächtige Suchmuster), Elemente der systemnahen Programmierung oder die Entwicklung von Programmen für grafische Oberflächen. Zudem erfahren Sie in Grundzügen, wie Netzwerkanwendungen programmiert werden, und Sie erhalten eine Einführung in die Programmierung verteilter Anwendungen mit der Java 2 Enterprise Edition und in Web Services. Den Abschluss bildet ein Überblick über die freie Entwicklungsumgebung Eclipse.
- ▶ Das in **Kapitel 10** behandelte *Software-Engineering* geht weit über reine Programmier-techniken hinaus: Sie lernen viele unterschiedliche Methoden zur Bearbeitung und Verwaltung von Softwareprojekten kennen. Unter anderem werden die Anwendungsmodellierung mit der UML, Grundlagen des Projektmanagements und verschiedene Ansätze des Software-Engineerings wie der Unified Process oder Extreme Programming vorgestellt. Für die in den IT-Ausbildungsberufen fällige Projektarbeit sind diese Themen überlebenswichtig.

- ▶ **Kapitel 11, Datenbanken**, befasst sich mit einer der wichtigsten Funktionsgrundlagen vieler Programmierprojekte und Anwendungen. Nach der üblichen Erläuterung von Begriffen und Konzepten wird als konkretes Datenbanksystem der weit verbreitete Open-Source-Datenbankserver MySQL eingeführt.
- ▶ In **Kapitel 12, Server für Webanwendungen**, wird zunächst das Web-Protokoll HTTP beschrieben. Danach geht es um die Installation und Konfiguration des verbreiteten Open Source-Webservers Apache 2 mitsamt der Web-Programmiersprache PHP.
- ▶ **Kapitel 13, Weitere Internet-Serverdienste**, befasst sich mit diversen weiteren Servern für TCP/IP-basierte Netzwerke: dem Nameserver BIND, dem Verzeichnisdienst-Server OpenLDAP, dem FTP-Server vsftpd sowie dem Universal- oder Meta-Server xinetd.
- ▶ **Kapitel 14, XML**, stellt die eXtensible Markup Language vor, eine Sprache, die der Definition beliebiger hierarchisch gegliederter Dokumentformate dient. In zahlreichen Anwendungen wird XML inzwischen eingesetzt, so dass es nützlich ist, die Konzepte dieses Formats zu kennen. Sie erfahren das Wichtigste über wohlgeformte Dokumente, über Formatbeschreibungen mit Hilfe von DTDs und XML Schema, die Umwandlung von XML-Dokumenten mit XSLT sowie über die Programmierung XML-basierter Anwendungen.
- ▶ **Kapitel 15, Weitere Datei- und Datenformate**, widmet sich den wichtigsten Formaten für Text, Bild und Multimedia. Es geht zunächst um Text und Zeichensätze, anschließend werden verschiedene textbasierte und binäre Dateiformate konkret erläutert.
- ▶ In **Kapitel 16, Webseitenerstellung mit (X)HTML und CSS**, wird zunächst HTML beziehungsweise XHTML vorgestellt, die Sprache, in der Webseiten verfasst werden. Hier werden verschiedene konkrete Aspekte der Webseitenerstellung erläutert, etwa die Textstrukturierung, der Listen- und Tabellensatz, das Einbetten von Bildern, Hyperlinks und Webformulare. Der zweite Teil beschreibt das konsistente Layout von Webseiten mit Cascading Style Sheets (CSS).
- ▶ In **Kapitel 17, Webserveranwendungen**, erfahren Sie, wie man Websites erstellt, die nicht nur aus statischen HTML-Dokumenten, sondern auch aus dynamisch generierten Inhalten bestehen. Konkret wird zuerst die Programmiersprache PHP 5 behandelt. Neben den allgemeinen Konzepten wird die Programmierung konkreter, datenbankbasierter Anwendungen beschrieben. Danach geht es um das moderne Web-Framework Ruby on Rails.
- ▶ **Kapitel 18, JavaScript und Ajax**, stellt die wichtigste clientseitige Programmiersprache vor, mit der Sie die Inhalte einer Webseite »zum Leben erwecken« können. Zunächst werden die klassischen Anwendungen wie die Ausgabe ins Dokument selbst, die Verarbeitung von Formularen oder der Austausch von Bildern behandelt, anschließend erfahren Sie das Wichtigste über DOM, die Standardtechnik, mit der Sie die Elemente eines Dokuments nachträglich modifizieren können, sowie Ajax, um die Inhalte für diese Änderungen ohne Neuaufbau der Seite vom Server nachzuladen.

- ▶ **Kapitel 19**, *Computer- und Netzwerksicherheit*, befasst sich mit verschiedenen Themen der immer wichtiger werdenden Sicherheit: Der Schutz vor Viren, Würmern und Trojanern wird ebenso behandelt wie Kryptographie, Firewalls oder Fragen der Datensicherheit.

Nach den Kapiteln folgen noch drei Anhänge:

- ▶ **Anhang A**, *Glossar*, enthält kurze Beschreibungen der wichtigsten IT-Stichwörter.
- ▶ In **Anhang B**, *Zweisprachige Wortliste*, finden Sie ein deutsch-englisches und ein englisch-deutsches Verzeichnis wichtiger Fachbegriffe.
- ▶ **Anhang C**, *Kommentiertes Literatur- und Linkverzeichnis*, empfiehlt weiterführende Bücher und Web-Ressourcen zu den Themen der verschiedenen Kapitel.

## Für wen ist dieses Buch geeignet?

In erster Linie können Sie mit diesem Handbuch etwas anfangen, wenn Sie eine Ausbildung im IT-Bereich oder verwandten Berufen absolvieren. Es wurde insbesondere für den im Titel genannten Ausbildungsberuf des Fachinformatikers in den beiden Fachrichtungen Anwendungsentwicklung und Systemintegration geschrieben, dürfte aber auch für IT-Systemelektroniker oder IT-Systemkaufleute gut geeignet sein. Genau wie in Ihrer Ausbildung werden auch in diesem Buch viele verschiedene Themenbereiche behandelt: Da Computer sehr komplexe Maschinen sind, ist es erforderlich, sich selbst dann ein großes Spektrum verschiedener Kenntnisse anzueignen, wenn Sie nur in einem bestimmten Fachbereich arbeiten möchten.

Auch für Studenten im Grundstudium der Informatik oder in den Informatik-Kursen anderer Studienrichtungen ist das Buch durchaus geeignet. Gerade der Mittelweg zwischen Darstellung der theoretischen Grundlagen und praktischer Anleitung dürfte für Sie eine wichtige Lücke schließen.

Zu guter Letzt ist dieses Buch aber natürlich auch für alle anderen geeignet, die an Computern, Programmierung oder Netzwerken interessiert sind. Es eignet sich nicht nur als Unterrichtsbegleiter, sondern auch zum Selbststudium einzelner Themen. Die einzige Voraussetzung ist im Grunde genommen, dass Sie einen Computer zur Verfügung haben und grundsätzlich wissen, wie man damit umgeht.

Zum Komplexitätsniveau ist noch anzumerken, dass dies weder ein Buch für absolute Neueinsteiger noch ein Begleiter für vollkommene Experten ist. Wenn Sie noch nie mit einem Computer gearbeitet haben, benötigen Sie eine grundlegendere Anleitung; im vorliegenden Buch erfahren Sie nicht, wie man das Gerät in Betrieb nimmt, seine Arbeit als Datei speichert, einen Ordner anlegt oder einen Webbrowser bedient. All diese Dinge (und noch einige mehr) müssen klar sein, bevor Sie etwas Sinnvolles mit diesem Handbuch an-

fangen können. Wenn Sie dagegen bereits Erfahrung haben, gibt es wahrscheinlich einige Themen, über die Sie noch nicht Bescheid wissen. In diesem Fall finden Sie möglicherweise hier, was Sie suchen.

### Icons

Um das Buch für Sie übersichtlicher und damit noch nützlicher zu machen, finden Sie an manchen Stellen die folgenden zwei Symbole in der Randspalte:

Dieses Icon markiert einen hilfreichen Tipp, eine Abkürzung oder eine nützliche Zusatzinformation. [«]

Mit diesem Symbol werden Warnungen gekennzeichnet. Sie sollten sie ernst nehmen, um keine unnötigen Datenverluste oder Fehlfunktionen zu erleiden. [!]

#### IHK-Fragen und Beispiellistings im Internet

Unter <http://www.buecher.lingoworld.de/fachinfo/fragen.html> finden Sie alle im Buch abgedruckten Beispiellistings, so dass Sie diese nicht abtippen müssen. Außerdem finden Sie dort Fragen, mit denen Sie sich auf die IHK-Prüfung vorbereiten können, sowie jeweils aktualisierte Literatur- und Linklisten.

## Danksagungen

Zunächst einmal, wie immer, vielen Dank an das Team von Galileo Press. Dass dieses Buch so gut aussieht, angenehm in der Hand liegt, ordentlich gebunden ist und sogar ein Lesebändchen besitzt, ist den Bemühungen des Verlags zu verdanken. Besonderer Dank gebührt wieder einmal meinem Lektor Stephan Mattescheck, der für die inhaltliche Seite zuständig ist und dieser Verantwortung stets mit Bravour gerecht wird.

Den größten Einzelbeitrag zu dieser Neuauflage hat der technische Gutachter Georg Wilm geleistet: Zunächst hat er die vorige Auflage gründlich durchgesehen und zu jedem Kapitel eine ausführliche Bewertung verfasst. Auf dieser Grundlage wurde die Neuauflage konzipiert, die er in der Rohfassung ebenfalls akribisch durchgearbeitet hat, woraus sich nochmals zahlreiche Ergänzungen und einige Berichtigungen ergaben. Kurz: Wenn Sie finden, dass diese Auflage mehr als je zuvor alles bietet, was Sie für Ihre IT-Ausbildung benötigen, dann ist dies zu einem großen Teil auch Herrn Wilm zu verdanken – meinen herzlichsten Dank dafür, und natürlich auch für das freundliche Geleitwort.


Danke auch an den Korrektor, René Wiegand, für einige wertvolle Verbesserungsvorschläge zu den mathematischen Themen in Kapitel 2.

Ich danke auch den zahlreichen Lesern der ersten beiden Auflagen, die mir nette Kommentare und auch Hinweise auf Fehler zugeschickt haben. Durch diesen wertvollen

## Vorwort

Beitrag konnte ich viele Stellen noch entscheidend verbessern oder klarer formulieren. Die fast durchweg positive Resonanz auf die beiden Voraufagen hat nicht nur zu meinem Wohlbefinden beigetragen, sondern auch die Motivation zum Schreiben des vorliegenden Buches gesteigert.

Eine Danksagung wäre nicht komplett, wenn sie nicht auch die wichtigsten Menschen in meinem Leben einschließen würde: meine Familie. Vor allem mein Sohn und meine Frau haben wieder einmal unzählige Stunden auf mich verzichten müssen und waren selbst immer für mich da. Meine Frau hat zudem diverse Infografiken in diesem Buch angefertigt.



*In der Informatik geht es genauso wenig um Computer  
wie in der Astronomie um Teleskope.  
– Edsger W. Dijkstra*

# 1 Einführung

In diesem Kapitel erhalten Sie einen Überblick über die wichtigsten Grundlagen der Informationstechnik: eine Abgrenzung der verschiedenen Fachrichtungen und Ausbildungsgänge der Informatik, die Entwicklungsgeschichte der Computer sowie eine allgemeine Übersicht über die digitale Speicherung verschiedener Arten von Informationen.

## 1.1 Informationstechnik, Informatik und EDV

Allgemein gesprochen geht es in diesem Buch um **Informationstechnik** (englisch: Information Technology oder kurz IT). Der traditionelle Begriff für diese Art der Technik lautet **Elektronische Datenverarbeitung** (EDV). **Daten** oder **Informationen** sind Werte, die im Zusammenhang mit beliebigen Sachverhalten angelegt werden oder die im Rahmen von Mess- oder Rechenvorgängen anfallen. **Datenverarbeitung** ist der Vorgang der Sammlung, Speicherung und Manipulation dieser Informationen. Im Lauf der Zeit haben sich unterschiedliche Verfahren der Datenverarbeitung entwickelt:

- ▶ Die **manuelle Datenverarbeitung** führt Berechnungen und Datenmanipulationen ohne jegliche Hilfsmittel durch, sie basiert auf Kopfrechnen und Auswendiglernen. Das äußerste erlaubte Hilfsmittel ist ein Schreibblock, um Daten oder Zwischenergebnisse zu notieren.
- ▶ Die **mechanische Datenverarbeitung** verwendet mechanische Hilfsmittel für die Verarbeitung von Informationen, beispielsweise eine mechanische Schreibmaschine oder einen Rechenschieber.
- ▶ Bei der **elektrischen Datenverarbeitung** werden elektrisch betriebene Geräte als Hilfsmittel eingesetzt, zum Beispiel elektrische Schreibmaschinen oder Registrierkassen.
- ▶ Die **elektronische Datenverarbeitung** verwendet schließlich elektronisch gesteuerte Arbeitsmittel, also Elektronenrechner oder Computer.

Arten der  
Datenver-  
arbeitung

### 1.1.1 Fachrichtungen der Informatik

Die wissenschaftliche Fachrichtung, die sich mit den verschiedenen Aspekten der Computertechnik auseinandersetzt, wird seit den 60er-Jahren des 20. Jahrhunderts als **Informatik** (englisch: *computer science*) bezeichnet; es handelt sich um ein Kunstwort aus Information und Mathematik. Die akademische Informatik wird üblicherweise in vier Fachrichtungen unterteilt:

- ▶ Die **theoretische Informatik** betrachtet insbesondere die mathematisch-logischen Grundlagen, die der Verwendung und Programmierung von Computern zugrunde liegen. Es geht beispielsweise um die **Berechenbarkeit** (Ist ein Problem überhaupt durch Berechnung lösbar?) und um **Automatentheorien** – die mathematisch-formalen Modelle, auf denen Rechner unabhängig von der elektronischen Machbarkeit aufbauen.
- ▶ Die **technische Informatik** beschreibt die elektronisch-technischen Eigenschaften der Bauteile, aus denen Computer zusammengesetzt sind. Ein wichtiges Teilgebiet der technischen Informatik ist die **Schaltalgebra**, die Umsetzung logischer Operationen durch elektronische Schaltungen.
- ▶ In der **praktischen Informatik** geht es im Großen und Ganzen um die **Programmierung** von Computern und die Mittel, die dazu erforderlich sind. Die Erforschung des Aufbaus von Betriebssystemen und Programmiersprachen-Compilern sowie deren Implementierung (praktische Umsetzung) sind die wichtigsten Teilgebiete.
- ▶ Die **angewandte Informatik** kümmert sich gewissermaßen um alles andere, nämlich um sämtliche Nutzenanwendungen von Computern. Das reicht von Datenbanken über die Netzwerkkommunikation bis hin zu Grafik, Animation und Audio-/Videobearbeitung.

Da es sich bei diesem Buch um ein Praxisbuch handelt, das nicht für das trockene Auswendiglernen von Lehrsätzen geschrieben wurde, sondern für die alltägliche Nutzung von Computern, ist es kaum verwunderlich, dass sich fast alle Kapitel mit Aspekten der angewandten Informatik beschäftigen. In den Kapiteln 5 bis 9 werden auch die wichtigsten Ansätze der praktischen Informatik, nämlich Betriebssysteme und diverse Themen der Programmierung, behandelt.

Einige grundlegende Aspekte der theoretischen Informatik lernen Sie in Kapitel 2, *Mathematische und technische Grundlagen*, kennen: Dort werden die wichtigsten mathematischen und logischen Operationen besprochen, die Computer ausführen. Außerdem wird beispielhaft auf die Realisierung einiger dieser Funktionen durch elektronische Bauteile eingegangen, also auf einige Ansätze der technischen Informatik.

Im Übrigen gibt es noch ein eigenes Kapitel zum Thema Hardware, die ebenfalls dem Gebiet der technischen Informatik zugeordnet werden kann (Kapitel 3, *Hardware*).



### 1.1.2 Überblick über die IT-Ausbildung

Das Berufsfeld der Informationstechnik ist vielfältig. Deshalb gibt es zahlreiche unterschiedliche Möglichkeiten einer Ausbildung in diesem Bereich. Sie lassen sich zum einen nach Fachgebieten wie Anwendungsentwicklung, Systemintegration oder kaufmännischem EDV-Einsatz unterscheiden. Zum anderen gibt es zwei grundlegende Ausbildungsformen: Berufsausbildung und Studium.

Aus diesen beiden Unterteilungen ergeben sich unter anderem folgende konkreten Ausbildungsgänge:

Überblick  
Ausbildungs-  
gänge

- ▶ Ausbildungsberufe
  - ▶ Fachinformatiker
  - ▶ IT-Systemelektroniker
  - ▶ IT-Systemkaufmann
  - ▶ Informatikkaufmann
- ▶ Studiengänge
  - ▶ Informatik
  - ▶ Informatik (FH)
  - ▶ Wirtschaftsinformatik
  - ▶ Medieninformatik
  - ▶ Bioinformatik
  - ▶ Medizinische Informatik

#### Ausbildungsberufe

Bis vor wenigen Jahren galt die Informationstechnik als zu komplex, um in einer praxisorientierten Berufsausbildung im klassischen dualen System (Ausbildungsbetrieb – Berufsschule) gelehrt zu werden. Die beiden angebotenen Ausbildungen zum EDV-Kaufmann und Büromaschinenelektroniker hatten andere Schwerpunkte, nämlich einen kaufmännischen beziehungsweise elektrotechnischen.

Erst 1996 wurden die IT-Berufe in der Bundesrepublik durch ein Übereinkommen von Arbeitgeberverbänden und Gewerkschaften unter Beratung des Bundesinstituts für Berufsbildung (BiBB) neu geordnet. Seitdem gibt es vier verschiedene IT-Ausbildungsberufe. Sie bereiten auf Tätigkeiten im Bereich der Informations- und Telekommunikationstechnik vor, die früher Bewerbern mit Studienabschluss vorbehalten waren oder in Einzelfällen an qualifizierte Quereinsteiger vergeben wurden.

IT-Berufe

Im Einzelnen handelt es sich um folgende Ausbildungsgänge:

- ▶ Fachinformatiker (Fachrichtungen Anwendungsentwicklung und Systemintegration)
- ▶ IT-Systemelektroniker
- ▶ IT-Systemkaufmann
- ▶ Informatikkaufmann

Fach-  
informatiker

Aufgabe der **Fachinformatiker** ist es gemäß offizieller Definition, »fachspezifische Anforderungen in komplexe Hard- und Softwaresysteme« umzusetzen. Diese recht ungenaue Beschreibung läuft in der Praxis auf sehr vielfältige Anforderungen hinaus, da sich nicht nur die beiden Fachrichtungen Anwendungsentwicklung und Systemintegration, sondern vor allem auch Ausbildungsbetriebe und Einsatzumgebungen oft stark voneinander unterscheiden. Gemeinsam ist ihnen allen lediglich der Umgang mit Computersystemen, mit Netzwerken, mit der unterschiedlichsten Software und dem komplexen Zusammenspiel dieser Komponenten.

Die beiden Berufsbilder sollen sich im Laufe der drei Ausbildungsjahre allmählich auseinander entwickeln. Bei Ausbildungsbeginn stehen gemeinsame Grundlagen im Vordergrund. Dazu gehören nicht nur fachspezifische Themen, wie sie in diesem Buch behandelt werden, sondern auch wichtige Informationen zum Arbeitsablauf und zum betrieblichen Umfeld. Dies sind insbesondere Grundlagen der Betriebswirtschaftslehre, der Projekt- und Betriebsorganisation, des Arbeits- und Ausbildungsrechts sowie der betrieblichen Buchführung. Diese Kenntnisse werden vor allem in der Berufsschule (60 Tage pro Ausbildungsjahr) vermittelt.

Anwen-  
dungsent-  
wicklung

Der Schwerpunkt des **Fachbereichs Anwendungsentwicklung** ist die Erstellung von Software für den eigenen Betrieb oder für Kunden. Der erste Schritt ist die Entwicklung neuer oder die Anpassung vorhandener Programme gemäß den Anforderungen der späteren Anwender. Es folgt die Einrichtung der Software auf den gewünschten Systemen, die gegebenenfalls dafür angepasst werden müssen. Anschließend sollen die Benutzer informiert oder gar geschult werden. In das Umfeld des Berufsbildes gehört demzufolge auch die Erstellung brauchbarer Dokumentationen für Entwickler, Administratoren und Endanwender.

System-  
integration

Im **Fachbereich Systemintegration** liegen die Hauptaufgaben im Bereich der Hardware-, Software- und Netzwerkeinrichtung. Hier sind gute Kenntnisse der verschiedenen Hardwarekomponenten, Betriebssysteme sowie der Netzwerkgeräte, -dienste und -protokolle gefragt. Auch die Schnittstellen zwischen der Informations- und der Telekommunikationstechnik spielen in diesem Beruf eine wichtige Rolle. Die Fachinformatiker im Fachbereich Systemintegration müssen das projektorientierte Arbeiten beherrschen und dabei alle Aufgaben planen, durchführen, testen und dokumentieren.

Ein weiteres, sehr wichtiges Arbeitsgebiet ist die Automatisierung administrativer Aufgaben. Dazu benötigen die Fachinformatiker Systemintegration umfangreiche Kenntnisse der Shells (Kommandozeileninterpreter) der jeweiligen Betriebssysteme sowie in verschiedenen Skriptsprachen. Unter Windows beginnt dies mit DOS-Batchdateien und ihren Nachfolgern, den Windows-CMD-Dateien; für komplexere Aufgaben sind auch der Windows Scripting Host und die neue Windows PowerShell nützlich. Im UNIX-Bereich ist die bash-Shell das grundlegende Werkzeug; größere Aufgaben werden dagegen mit Editor- und Skriptsprachen wie sed, awk und Perl gelöst, wobei Perl die wichtigste dieser Sprachen ist und bei fortgeschrittenen Kenntnissen alle anderen ersetzen kann. In Kapitel 6 bis 9 dieses Buches werden die wichtigsten Kenntnisse zu diesen Shells und Sprachen vermittelt.

Der Beruf des **IT-Systemelektronikers** ist elektrotechnischer orientiert als der des Fachinformatikers. Mehr als beim Fachinformatiker der Fachrichtung Systemintegration liegt der Schwerpunkt hier im Bereich der Hardware. Zu den wichtigsten Aufgaben gehört die Verkabelung von Computern und Netzwerken, angefangen bei der Stromversorgung über den Anschluss von Peripheriegeräten bis hin zu Besonderheiten der Netzwerkinfrastruktur. Dazu benötigen IT-Systemelektroniker ein umfangreiches Know-how über Hardwarekomponenten und Treiber sowie über deren Zusammenarbeit mit Betriebssystemen und Anwendungsprogrammen. Hinzu kommen Kenntnisse über ISDN und andere Telekommunikationstechniken. Zu den wichtigsten Arbeitgebern für Systemelektroniker gehören große Telefongesellschaften, sodass deren spezifische Anforderungen dieses Berufsbild wesentlich geprägt haben.

IT-Systemelektroniker

**IT-Systemkaufleute** sind im Wesentlichen die Kundenberater für Hard- und Softwareprojekte. Sie verfügen über technisches Know-how, sodass sie die Einzelheiten von IT-Projekten verstehen und den Kunden erklären können, und benötigen umfangreiche betriebswirtschaftliche und kaufmännische Kenntnisse, um die Kalkulation für ein solches Projekt durchführen zu können. Sie sind für alle Phasen der Kundenbetreuung für IT-Lösungen zuständig: angefangen bei der Marktanalyse über Beratung und Marketing bis hin zum Vertrieb sowie Einarbeitung und Schulung.

IT-Systemkaufleute

Bei den **Informatikkaufleuten** stehen die kaufmännischen Aufgaben stärker im Vordergrund, während die technischen Aspekte etwas weniger ausgeprägt sind. Sie bilden das Bindeglied zwischen wirtschaftlichen und technischen Abteilungen. Typischerweise vermitteln sie zwischen den Kunden, die vor allem wirtschaftliche Fragen haben, und den Beratern der Hard- und Softwarehersteller (oft den oben genannten IT-Systemkaufleuten), die eine eher technische Sicht auf Vorgänge benötigen. Beispielsweise könnten Informatikkaufleute die technische Umsetzung von Geschäftsprozessen anleiten und entsprechende Pflichtenhefte erstellen. Eine weitere Aufgabe wäre es, die Unterschiede verschiedener IT-Lösungen in kaufmännisch kalkulierbaren Zahlen auszudrücken.

Informatikkaufleute

Alle IT-Ausbildungen – mit Ausnahme zweijähriger Umschulungsmaßnahmen – beinhalten eine (ausschließlich theoretische) Zwischenprüfung. Es besteht Teilnahmepflicht, aber das Ergebnis ist rein informativ – sowohl für die Auszubildenden als auch für die Betriebe.

**Abschlussprüfung** Die Abschlussprüfung besteht aus einer selbst gewählten Projektarbeit, die von der jeweils zuständigen Industrie- und Handelskammer genehmigt werden muss. Ihre Dauer beträgt bei den Fachinformatikern des Fachbereichs Anwendungsentwicklung 72 Stunden, wobei auch ein Pflichtenheft erstellt werden muss. Bei den anderen Ausbildungsgängen beträgt die Dauer lediglich 36 Stunden und der Projektablauf beschränkt sich hier auf die vier Phasen Planung, Durchführung, Test und Abnahme.

Innerhalb der veranschlagten Zeit müssen sowohl das Projekt selbst als auch eine umfangreiche Dokumentation dazu erstellt werden; Letzteres sollte etwa acht Stunden der Projektdauer ausmachen. Die Dokumentation wird bei der IHK eingereicht; zudem müssen die Auszubildenden ihre Projekte dort präsentieren und ein Fachgespräch dazu führen. Wichtige Hinweise zum projektorientierten Arbeiten erhalten Sie in Kapitel 10, *Software-Engineering*.

Die zweite Säule der Abschlussprüfung bildet die bundeseinheitliche schriftliche Prüfung. Sie besteht aus drei Teilen:

- ▶ Zwei **ganzheitliche Aufgaben** (GA) beschreiben je ein virtuelles IT-Szenario und stellen sechs komplexe Fragen dazu. Die Auszubildenden müssen eine dieser Fragen, mit deren Thema sie sich am wenigsten auskennen, streichen, und die übrigen fünf ausführlich beantworten. Maximal sind dabei für jede beantwortete Frage 20 Punkte, das heißt insgesamt 100 für jede GA erreichbar. Die GA 1 enthält fachspezifische Fragen für den jeweiligen Ausbildungsgang, während die GA 2 die für alle IT-Ausbildungsberufe identische Kernqualifikation abfragt. Für jede GA sind 90 Minuten angesetzt.
- ▶ Der dritte Teil ist eine 60 Minuten dauernde Prüfung zu den Themen Wirtschaft und Soziales einschließlich Arbeits- und Ausbildungsrecht. Diese Prüfung besteht aus isolierten Fragen, wobei einige offen, viele dagegen im Multiple-Choice-Verfahren zu beantworten sind.

**Ausbildung und dieses Buch** In Tabelle 1.1 sehen Sie, welche Kapitel dieses Buches für die Auszubildenden der einzelnen Fachrichtungen besonders nützlich (x) beziehungsweise unerlässlich (+) sind. Wenn ein Kapitel für Ihren Beruf nicht angekreuzt wurde, bedeutet dies nicht, dass Sie sich gar nicht mit diesem Thema beschäftigen sollten! Wie bereits im Vorwort erwähnt, ist ein möglichst gründliches Allgemeinwissen eine der wichtigsten Voraussetzungen für den beruflichen Erfolg. Zudem verstehen Sie Zusammenhänge der IT- und Medienwirtschaft umso besser, je größer Ihr Überblick ist.

Kapitel	Fachinf. Anwend.	Fachinf. Systemint.	Kaufl.	IT-Systemelektr.
1	+	+	+	+
2	+	+	×	+
3	×	+	×	+
4	+	+	×	+
5	+	+	×	+
6	+	+	×	+
7	+	+	×	+
8	+	+	×	+
9	+	×		×
10 <sup>1</sup>	+	+	+	+
11	+	+	×	×
12	×	+		×
13	×	+		×
14	+	+	×	×
15	+	+	×	×
16	+	×	×	×
17	+	×		
18	+	×		
19	×	+	×	+

**Tabelle 1.1** Die einzelnen Kapitel dieses Buches und ihre Relevanz für die verschiedenen Ausbildungsberufe

### Studiengänge

Die Studiengänge im Bereich der Informationstechnik sind wesentlich gründlicher, dafür aber auch viel theoretischer orientiert als die betrieblichen Ausbildungsgänge. Neben den hier behandelten Studienfächern der Informatik gibt es übrigens auch angrenzende Fachgebiete wie Telematik, Elektrotechnik, Mechatronik oder angewandte Mathematik.

Das allgemeinste Spektrum bietet das Studium der **Informatik**. Das Grundstudium beinhaltet Einführungen in alle vier Disziplinen des Faches (theoretische, technische, praktische und angewandte Informatik). Im Hauptstudium ist dagegen eine Spezialisierung auf einen bestimmten Bereich innerhalb eines dieser Zweige vorgesehen. Denkbar wären etwa Betriebssysteme aus dem Gebiet der praktischen Informatik oder Datenbanken, die zur angewandten Informatik zählen.

Informatik

<sup>1</sup> Für Fachinformatiker Anwendungsentwicklung ist das gesamte Kapitel relevant, für die anderen Ausbildungsberufe dagegen vor allem die Ausführungen zum Projektmanagement.

Das Informatikstudium wird sowohl an Universitäten als auch an Fachhochschulen angeboten. Naturgemäß ist Letzteres etwas praktischer orientiert; das Lernen erfolgt besonders im Grundstudium schulähnlicher. In beiden Lehreinrichtungen ist der typische Abschluss das klassische Diplom; dazu muss eine Diplomarbeit über ein bestimmtes, selbst gewähltes Forschungsthema verfasst werden. Daneben bieten immer mehr Hochschulen und Fachhochschulen auch internationale Abschlüsse wie Bachelor und Master an, die allerdings erst allmählich durch (insbesondere deutsche) Arbeitgeber anerkannt werden.

**Nebenfächer** Wie in jedem mathematisch-naturwissenschaftlichen Studium müssen auch bei der Informatik Nebenfächer belegt werden. Die Auswahl ist so vielfältig wie die Anwendungsbereiche der Informationstechnik: Für theoretisch ausgerichtete Forschungen wäre etwa die Mathematik interessant; ein Fach aus dem Bereich der Wirtschaftswissenschaften eröffnet den Zugang zur Entwicklung kaufmännischer Systeme und Anwendungen. In diversen Bereichen der angewandten Informatik könnten auch Fächer wie Biologie oder Physik von Interesse sein. Spannend ist auch die Kombination mit philologischen Fächern, besonders mit den Sprachwissenschaften oder der philosophischen Logik.

Umgekehrt kann Informatik auch für Studiengänge in allen genannten Fächern als Nebenfach belegt werden.

**Spezielle Studiengänge** Einen Schritt weiter als die Wahl eines bestimmten Nebenfaches gehen gezielte Kombinationsstudienfächer. Die wichtigsten von ihnen sind folgende:

- ▶ **Wirtschaftsinformatik.** Eine gezielte Kombination aus Themen der (insbesondere angewandten) Informatik und der Betriebswirtschaftslehre. Wirtschaftsinformatiker können in allen Bereichen der kaufmännischen IT-Anwendung arbeiten, vor allem in Banken, Versicherungen oder den Fachabteilungen großer Unternehmen, aber auch in der Entwicklung kaufmännischer Softwarelösungen.
- ▶ **Medieninformatik.** Dieser Studiengang stellt den Bereich der multimedialen IT-Anwendung in den Mittelpunkt. Neben den Grundlagen der Informatik lernen die Studierenden vor allem den praktischen IT-Einsatz kennen: Grafik- und Bildbearbeitung, Audio- und Videoschnitt und die Erstellung multimedialer Präsentationen. Von allen IT-Studiengängen erfordert dieser das größte kreative Potenzial.
- ▶ **Bioinformatik.** Die Forschungsergebnisse der modernen Biologie, besonders der Mikrobiologie und der Gentechnik, könnten ohne umfangreiche IT-Unterstützung nicht ausgewertet werden. Bioinformatiker bilden die Schnittstelle zwischen Biologie und Informationstechnik; sie erstellen die Anwendungen zur Analyse biologischer Forschungen, etwa der Sequenzanalyse von Genen. Die Bioinformatik ist möglicherweise der wachstumstärkste Sektor in der gesamten IT-Landschaft.
- ▶ **Medizinische Informatik.** Auch in der modernen Medizin spielt Computertechnik eine überaus wichtige Rolle: Von der Verwaltung der Patienten- und Diagnosedaten über die automatisierte Analyse von Laborproben bis hin zum halbautonom arbei-

tenden OP-Roboter ist der Praxis- und Klinikalltag ohne IT nicht mehr denkbar. In der medizinischen Informatik erwerben die Studierenden sowohl fundierte medizinische als auch wichtige informationstechnische Fachkenntnisse, sodass sie in den verschiedensten Bereichen der Medizintechnik und -praxis arbeiten können.

Die Tiefe des für ein Studium erforderlichen Wissens geht weit über das hinaus, was ein einzelnes allgemeines Grundlagenbuch wie das vorliegende leisten kann. Dennoch kann dieses Buch auch im Studium ein unentbehrlicher Begleiter sein: Hier können Sie die Grundlagen der Theorie und Praxis der unterschiedlichen Themen schnell und übersichtlich nachschlagen – als letzte Absicherung vor der entscheidenden Fachklausur oder auch als Anleser für die Veranstaltungsplanung des kommenden Semesters.

Studium und  
dieses Buch

## 1.2 Die Geschichte der Rechenmaschinen und Computer

In diesem Buch ist von Computern die Rede, und natürlich wissen Sie ganz genau, was das ist. Wenn Sie allerdings den Versuch machen sollten, einen Computer mit allen Dimensionen seiner heutigen Möglichkeiten griffig zu definieren – wie würde diese Definition lauten? Vergangene Generationen sprachen etwa von **programmgesteuerten Rechenautomaten**, und gerade Fachleute scheinen noch heute lieber Rechner als Computer zu dieser Maschine zu sagen.

Aber ist Rechnen heutzutage die wichtigste Aufgabe der Computer? Es scheinen schließlich mehr Leute einen Webbrowser zu verwenden als die Tabellenkalkulation Excel, und das Bildbearbeitungsprogramm Photoshop ist erheblich populärer als spezielle Mathematiksoftware wie Mathematica oder Maple.

Sind Com-  
puter nur  
Rechner?

Trotzdem ist ein Computer ein Gerät, das Probleme durch Berechnungen löst: Er kann nur diejenigen Sachverhalte »verstehen«, die man in Form von Zahlen und mathematischen Formeln darstellen kann. Dass es sich dabei heute auch um Bilder, Töne, Animationen, 3D-Welten oder Filme handeln kann, liegt einfach an der enormen Rechengeschwindigkeit und Kapazität moderner Rechner.

Sehen Sie sich den Begriff »programmgesteuerter Rechenautomat« noch einmal genau an: Ein »Rechenautomat« ist ein Gerät, das automatisch etwas berechnet, sodass man dies nicht manuell erledigen muss. Das kann auch ein Taschenrechner oder sogar eine mechanische Rechenmaschine. Das Besondere, das ein Computer zu bieten hat, beschreibt der Begriff »programmgesteuert«. Ein Computerprogramm ist eine Abfolge von Rechenvorschriften, die aufeinander aufbauen können und Schritt für Schritt ausgeführt werden.

Mit anderen Worten ist ein Computer nicht nur ein Rechenautomat, sondern ein

Algorithmen

**Mohamed Ibn Musa Al Chwarismi**, ist eine Schritt-für-Schritt-Anleitung zum Lösen mathematischer Probleme. Jeder Computer versteht eine oder mehrere formale Sprachen, in denen man ihm solche Algorithmen einprogrammieren kann. Ist ein Algorithmus erst einmal im Computer gespeichert, kann er immer wieder mit anderen Daten ausgeführt werden. Betrachten Sie etwa den folgenden Algorithmus aus dem Alltagsleben, der die Internet-Nutzungskosten verschiedener Kunden eines Providers berechnet:

1. Eingabe Tarif: Flatrate oder minutenbasiert?
2. War es die Flatrate? Macht 19,99 €. Berechnung beendet.
3. Minutenbasiert: Eingabe der Minuten.
4. Multipliziere die Minuten mit 0,01 €.
5. Addiere die Grundgebühr von 2,49 € hinzu. Berechnung beendet.

Dieser einfache Algorithmus kann natürlich ohne Weiteres von einem Menschen abgearbeitet werden. Er wird zwar langsamer rechnen als ein heutiger Computer, aber durchaus innerhalb einer annehmbaren Zeit damit fertig werden. Andererseits kann der Computer Millionen solcher Berechnungen in der Sekunde ausführen. Zwar wird er gewisse Leistungen des menschlichen Geistes wahrscheinlich niemals erreichen, aber schneller rechnen kann er allemal.

Darüber hinaus gibt es Algorithmen, an denen wir Menschen schlichtweg verzweifeln würden: Oder hätten Sie Lust, jeden einzelnen Pixel eines  $10 \times 10$  cm großen hochauflösenden Bildes anhand der Farben seiner acht umgebenden Pixel neu zu berechnen, um das Bild zu vergrößern oder zu verkleinern? Ohne Sie entmutigen zu wollen: Ein solches Bild besteht, wenn es für den Druck geeignet sein soll, aus etwa 1.392.400 Pixeln – viel Spaß beim Rechnen!<sup>2</sup>

So macht der Computer sich selbst immer unentbehrlicher: Sobald Computersysteme eine gewisse Komplexität erreichen, wird eine Anwendung erfunden, die ohne sie nicht zu bearbeiten wäre. Um diese Anwendung dann schneller und effizienter ausführen zu können, entsteht die nächste Computergeneration, für die dann wieder neue Anwendungsgebiete gefunden werden. Auf diese Weise ist der Fortschritt in der Computerentwicklung nicht aufzuhalten, und was auch immer als unüberwindbare Grenze galt, wurde von cleveren Ingenieuren irgendwann überschritten. Der vorliegende Abschnitt versucht, diesen Weg ein wenig konkreter zu beschreiben.

---

<sup>2</sup> Zugrunde gelegt wurde hier die gängige Druckauflösung von 300 Pixel/Inch, was etwa 118 Pixeln/cm entspricht (1 Inch = 2,54 cm).

### 1.2.1 Die Vorgeschichte

Seit Menschen überhaupt sesshaft in größeren Gemeinschaften zusammenleben, sind sie zum Rechnen gezwungen, um diese Gemeinschaften zu organisieren. Zu diesem Zweck wurden im Laufe der Zeit immer komplexere und ausgeklügeltere Geräte erfunden.

Das erste Rechenhilfsmittel in der Geschichte war die Rechentafel oder der Abakus. Ähnliche Geräte wurden unabhängig voneinander zum Beispiel in China und im alten Rom entwickelt. Es handelte sich um eine Tafel mit verschiebbaren Steinen, die in mehreren Spalten angeordnet waren. Das Ganze ähnelte den heutigen kugelbestückten Rechenhilfen, wie sie etwa in Grundschulen eingesetzt werden. Interessant ist, dass die Römer gar nicht erst versuchten, ihr recht umständliches Zahlensystem auf den Abakus zu übertragen, sondern ein modernes Stellenwertsystem darauf verwendeten.

Überhaupt kann die Erfindung der Stellenwertsysteme zum Schreiben von Zahlen als eine der wichtigsten Errungenschaften auf dem Weg zum Computer betrachtet werden: Die brillante Idee, den Wert einer Ziffer von ihrer Position innerhalb der ganzen Zahl abhängig zu machen, hatten die Inder; die heutigen »arabischen« Zahlen wurden in Indien erfunden und später von den Arabern – erstmals durch den bereits genannten Al Chwarismi im 9. Jahrhundert – übernommen. Die wichtigste Erfindung überhaupt ist in diesem Zusammenhang die Null. Die indischen Mathematiker notierten sie zunächst als Punkt und dann bis heute als Kreis. Ohne die Null ist es nicht möglich, den Wert einer einzelnen Ziffer zu vervielfachen: Auch wenn sie für sich allein keinen mathematischen Wert besitzt, werden Sie zugeben, dass 2.000 etwas völlig anderes bedeutet als 2.

Stellenwert-  
systeme

Die Erfindung der Stellenwertsysteme war der Ausgangspunkt für die Konstruktion immer ausgefeilterer mechanischer Rechenmaschinen: Im 17. Jahrhundert konstruierte der französische Mathematiker und Philosoph **Blaise Pascal** eine Addiermaschine, 1722 erfand **Gottfried Wilhelm Leibniz** eine komplexere Maschine, die bereits sämtliche Grundrechenarten beherrschte. Solche mechanischen Rechenapparaturen arbeiteten mit einem komplexen Gefüge aus Zahnrädern und ähnlichen Bauteilen, die aus der Uhrmacherei übernommen wurden.

Erst im 19. Jahrhundert entstand die Idee zu einer Apparatur, die mit heutigen Computern vergleichbar ist. Der britische Wissenschaftler **Charles Babbage** hatte bereits eine klassische mechanische Rechenmaschine namens **Differential Engine** gebaut und plante um 1850 den Bau eines weiteren mechanischen Geräts namens **Analytical Engine**. Es sollte der erste frei programmierbare Rechenautomat der Welt werden. Ohne es zu ahnen, nahm Babbage einige der wichtigsten Designprinzipien des Computers vorweg: Das Gerät sollte aus einem Rechenwerk namens »mill«, einem Programm- und Datenspeicher und einer Ein- und Ausgabereinheit bestehen. Auch wenn spätere Rekonstruktionen auf der Basis von Babbages Aufzeichnungen ergaben, dass die Maschine funktioniert hätte, war Babbage selbst nicht in der Lage, sie fertig zu stellen, weil die damalige Mechanik keine Bauteile in ausreichender Präzision liefern konnte.

Babbages  
Analytical  
Engine

Interessant ist dagegen, dass es Grundideen einer Programmiersprache für die Analytical Engine gab. Babbage engagierte **Ada Lovelace**, die Tochter des berühmten Dichters Lord Byron, für die Entwicklung einer solchen Sprache. Ihr zu Ehren wurde eine in den 1970er-Jahren entwickelte Programmiersprache **Ada** genannt.

**Die Zuse Z3** Während die Rechenmaschinen in der zweiten Hälfte des 19. Jahrhunderts immer leistungsfähiger und komplexer wurden, beachtete dennoch niemand den Gedanken der Programmierbarkeit. Alle Rechenmaschinen bis etwa 1940, auch die elektrischen, waren auf die Berechnung einzelner eingegebener Rechenaufgaben beschränkt. Erst um diese Zeit erfand der Bauingenieur **Konrad Zuse** in Berlin programmierbare elektromechanische Rechenautomaten. Sein dritter Versuch, die Z3, funktionierte tatsächlich. Die Maschine arbeitete mit Relais, elektromagnetischen Schaltern aus der Telefontechnik. Sie verwendete eine binäre Fließkommaarithmetik mit einer Wortbreite von 22 Bit: 1 Bit für das Vorzeichen, 14 Bit für die Mantisse und 7 Bit für den Exponenten. Der Hauptspeicher besaß eine Kapazität von 64 Maschinenwörtern, also  $64 \times 22$  Bit.

#### Elektrizität, Elektromechanik und Elektronik

Bereits Anfang des 20. Jahrhunderts wurden die ersten elektrischen Rechenmaschinen konstruiert. Die Rechner, die Konrad Zuse seit den 1930er-Jahren baute, waren elektromechanisch. In den 1940er-Jahren begann man in den USA und Großbritannien mit der Entwicklung elektronischer Rechner.

In diesem Zusammenhang ist es wichtig, den Unterschied zwischen Elektrizität, Elektromechanik und Elektronik herauszustellen:

- ▶ Ein **elektrisches** Gerät ist jede Maschine, bei der mechanische Bauteile durch elektrischen Strom angetrieben werden.
- ▶ **Elektromechanisch** werden alle Schaltelemente genannt, die durch einen Elektromagneten gesteuert werden, beispielsweise die Relais, die Zuse für seine ersten Computer verwendet hat. Mit anderen Worten geschieht die Steuerung des Geräts durch elektrischen Strom, allerdings indirekt durch mechanische beziehungsweise magnetische Bauteile.
- ▶ **Elektronische Geräte** schließlich werden allein durch elektrischen Strom gesteuert – Bauteile wie die klassische Elektronenröhre oder der neuere Transistor dienen als elektrisch gesteuerte Schalter ohne mechanische Teile.

### 1.2.2 Die Entwicklung der elektronischen Rechner

Computer im heutigen Sinne sind alle programmierbaren Rechner, die elektronisch arbeiten. Die elektronischen Computer lassen sich grob in vier Generationen einteilen:

- ▶ 1. Generation: **Röhrenrechner**  
Seit den 1940er-Jahren wurden Rechner auf der Basis von **Elektronenröhren** entwickelt.

► 2. Generation: **Transistorrechner**

Ab den 1950er-Jahren wurden die teuren und stromhungrigen Röhren durch **Transistoren** ersetzt.

► 3. Generation: **Rechner mit integrierten Schaltkreisen**

In den 1960er-Jahren gelang es, durch fotolithografische Verfahren große Mengen von Transistorschaltungen auf Halbleiterplatten unterzubringen – es begann die Entwicklung der **integrierten Schaltkreise** (englisch: Integrated Circuits, abgekürzt ICs oder Chips).

► 4. Generation: **Rechner mit Mikroprozessor**

Die frühen ICs waren fest verdrahtet – sie konnten nur eine einzelne, festgelegte Aufgabe erfüllen. Solche Chips für spezielle Anwendungszwecke gibt es noch heute. Anfang der 1970er-Jahre wurden zusätzlich frei programmierbare ICs entwickelt, die man **Mikroprozessoren** nannte. Es dauerte allerdings noch über fünf Jahre, bis die ersten Computer mit diesen Prozessoren konstruiert wurden.



**Abbildung 11** Eine kleine Auswahl verschiedener Elektronenröhren

### Röhrenrechner

Auf der Grundlage der Forschungen von **John von Neumann**, der das theoretische Modell eines Computers formulierte, wurden in den USA Geräte wie Harvard Mark I und Mark II oder der berühmte ENIAC gebaut. Diese erste Generation elektronischer Computer arbeitete mit **Elektronenröhren** als Schaltelementen. Diese mit den Glühlampen verwandten Vakuumbauteile wurden seit Ende des 19. Jahrhunderts entwickelt und dienten verschiedenen Zwecken – denken Sie beispielsweise an das Röhrenradio, in dem die Elek-

tronenröhre als Verstärkerelement eingesetzt wird. Abbildung 1.1 zeigt eine Auswahl verschiedener Elektronenröhren, die zwischen 1927 und 1960 hergestellt wurden. Das Bild stammt übrigens aus dem virtuellen Physik-Museum der Universität Innsbruck (<http://exphys.uibk.ac.at/museum/>); Herr Professor Denoth stellte es mir freundlicherweise zur Verfügung. Auf der angegebenen Website finden Sie noch viele weitere interessante Bilder und Informationen zur Physikgeschichte, darunter auch weitere Röhren, die seit 1870 entwickelt wurden.

Das für die Computertechnik interessanteste Röhrenmodell war die **Triode**, die mit ihren drei Anschlüssen die früheste Verwirklichung eines rein elektronischen Schalters darstellt: Einer der drei Anschlüsse dient der Steuerung; wenn dort Spannung anliegt, fließt Strom durch die beiden anderen Anschlüsse.

Programmiert wurden die meisten Röhrenrechner durch Schalter und Steckverbindungen an großen Schalttafeln. Die einzige Möglichkeit, ein bestimmtes Programm für die spätere erneute Ausführung zu »speichern«, bestand darin, den Zustand der Schalttafel aufzumalen oder zu fotografieren. So dauerte es oft mehrere Stunden, den Computer in die Lage zu versetzen, komplexe Aufgaben zu erfüllen.

Erst allmählich begann man mit der Verwendung von Lochkarten zur Programm- und Dateneingabe. Die Lochkarte selbst wurde im 19. Jahrhundert erfunden, um mechanische Webstühle zu steuern. Der Ingenieur **Hermann Hollerith**, ein Mitbegründer der späteren IBM, setzte sie um 1900 zur Speicherung von Daten einer US-Volkszählung ein, was die Berechnung der Ergebnisse mithilfe von Rechenmaschinen von den erwarteten Jahren auf wenige Wochen reduzierte.

### Transistorrechner

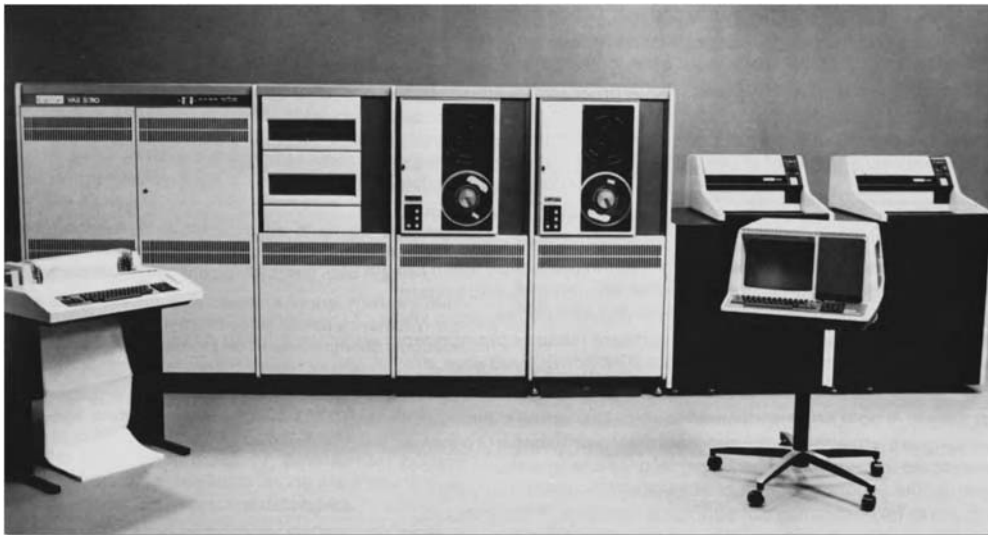
Röhrenrechner hatten einige gravierende Nachteile: Sie waren zu groß, zu stromhungrig und wegen der gewaltigen Hitzeentwicklung zu störanfällig. Aus diesen Gründen wurde bald der 1947 erfundene **Transistor** für die Computerentwicklung eingesetzt. Transistoren sind **Halbleiterbauteile**, die prinzipiell dieselben Schaltaufgaben erledigen können wie die verschiedenen Arten von Röhren, aber kleiner, billiger und weniger stromhungrig sind. Der Begriff »Halbleiter«, der im Zusammenhang mit Computern immer wieder genannt wird, bezeichnet übrigens ein Material, dessen elektrische Leitungsfähigkeit etwa in der Mitte zwischen den Leitern (vielen Metallen) und Isolatoren liegt. Das beliebteste chemische Element zur Fertigung von Halbleiterbauteilen ist Silizium, was der berühmten Gegend in Kalifornien, in der sich in den 1960er-Jahren zahlreiche Elektronikfirmen ansiedelten, den Namen Silicon Valley eingebracht hat.

Miniaturisierung durch Transistoren

Erst durch die Einführung des Transistors gelangte die Elektronik zu ihrer vollen Blüte. Dies zeigte sich vor allem an den Transistorradios, die seit den 1950er-Jahren zuhauf verkauft wurden. Auch für die entstehende Computerindustrie ergaben sich neue Impulse:

Durch die Transistortechnik ließen sich kleinere, leistungsfähigere und weniger störanfällige Rechner konstruieren. Natürlich ist »klein« und »leistungsfähig« relativ. Angesichts eines heutigen PCs oder gar Notebooks waren auch die Transistorrechner monströs, mindestens jedoch so groß wie eine ganze Reihe gewaltiger Kleiderschränke.

Auch die Transistorrechner wurden anfangs vor allem durch Lochkarten gesteuert. Mitte der 1960er-Jahre begann man allerdings mit der Entwicklung von **Terminals** für den direkten Dialog mit dem Computer. Ein Programmierer konnte über eine Tastatur Befehle eingeben und ein unmittelbares Feedback erhalten, anfangs über einen Fernschreiber-Endlosdrucker, später über einen Monitor. Die Geschichte der Steuerung von Computern wird in Kapitel 5, *Betriebssystemgrundlagen*, genauer behandelt.



**Abbildung 1.2** Eine VAX von Digital Equipment mit Terminals verschiedener Generationen.  
Quelle: Wikipedia

### Computer mit integrierten Schaltkreisen

Der Übergang von Transistorrechnern zu Computern mit ICs verlief unspektakulär und allmählich. Bereits in den frühen 1970er-Jahren waren deshalb Computer in verschiedenen Größen verfügbar: Die Großrechner oder Mainframes – vor allem von IBM produziert – bildeten die Rechenzentren von großen Behörden, Versicherungskonzernen oder Universitäten. Daneben kamen die sogenannten Kleincomputer auf (noch immer größer als die meisten Kleiderschränke); einer der führenden Hersteller war die Digital Equipment Corporation (DEC). Die Kleincomputer waren zwar nicht ganz so leistungsfähig wie Mainframes, dafür aber flexibler und sogar – mit entsprechend großen LKWs – transportfähig. Besonders wichtige Beispiele für Kleincomputer sind die Geräte der DEC-PDP-Baureihe, die untrennbar mit der Geschichte des Betriebssystems UNIX verknüpft ist.

Durch die Verwendung von ICs kam eine weitere Klasse von Geräten hinzu: die sogenannten Minicomputer. Sie waren etwa so groß wie eine größere Kommode und verhältnismäßig leicht, aber durch die fortschreitende Miniaturisierung nicht viel weniger leistungsfähig als Kleincomputer. Ein herausragendes Beispiel für den Minicomputer sind spätere Ausgaben der ebenfalls von DEC stammenden VAX-Baureihe (ein älteres Modell sehen Sie in Abbildung 1.2).

Für Klein- und Minicomputer wurden übrigens die ersten standardisierten Betriebssysteme und Anwendungsprogramme entwickelt. Dies erforderte die Entwicklung leicht kopierbarer Speichermedien. Ein wichtiger Schritt in diese Richtung war die Verwendung von Magnetbandspulen als Datenspeicher, ihr Aussehen und ihre Funktionsweise entsprachen den etwa zur selben Zeit verbreiteten Tonbändern.

### Mikrocomputer

1971 wurde der erste programmierbare Mikrochip entwickelt, genannt **Mikroprozessor**. Allgemein wird der Intel 4004 als erster Prozessor betrachtet; später stellte sich aber heraus, dass ein anderes Unternehmen bereits einige Monate vor Intel einen ähnlichen Chip entwickelt hatte. Allerdings wussten die Intel-Entwickler nichts davon.

Der 4004 war ein 4-Bit-Mikroprozessor. Er konnte also Informationen verarbeiten, die aus einer Abfolge von vier Einsen oder Nullen bestanden. Mit dieser **Wortbreite** lassen sich sechzehn verschiedene Werte darstellen, zum Beispiel die Zahlen 0 bis 15. Der Prozessor verstand verschiedene Arten von grundlegenden Befehlen: Er beherrschte arithmetische Operationen, also Berechnungen in den Grundrechenarten, und konnte logische Verknüpfungen und Vergleiche durchführen und auf der Basis ihrer Ergebnisse die »Entscheidung« treffen, an einer anderen Stelle im Programm fortzufahren. Die mathematisch-logischen Grundlagen und ihre Verwirklichung durch elektronische Bauteile werden im nächsten Kapitel, *Mathematische und technische Grundlagen*, vorgestellt.

Die ersten  
Personal  
Computer

Intel unterschätzte zu Anfang die Möglichkeiten des Mikroprozessors; er wurde in Rechenmaschinen und Ampelanlagen eingebaut, aber nicht in einen Computer. Erst 1975 baute die Rechenmaschinenfirma MITS einen einfachen Mikrocomputer-Bausatz, den Altair 8800. Er war mit einer Reihe von DIP-Schaltern für die Eingabe der einzelnen Bits und einer Reihe von Leuchtdioden zur Anzeige der Ergebnisse ausgestattet. Der verwendete Mikroprozessor war der Intel 8080, der bereits 8 Bit verarbeiten konnte. Für dieses zunächst nicht besonders nützliche Gerät entwarfen Bastler Schnittstellen für Monitor und Tastatur, und ein junger Programmierenthusiast schrieb einen Interpreter (zeilenweisen Übersetzer) für die einfache Großrechner-Programmiersprache BASIC, der auf dem Gerät lief. Der Programmierer war **Bill Gates**. Die Entwicklung von Programmiersprachen-Paketen für die beginnende Personal- und Homecomputer-Industrie war das erste Geschäftsfeld seiner 1976 gegründeten Firma **Microsoft**.

Große Verdienste im Zusammenhang mit der PC-Entwicklung haben übrigens die Forscherinnen und Forscher des **XEROX PARC** (der Name PARC steht für Palo Alto Research Center). Diese Forschungseinrichtung in Kalifornien hat eine interessante Geschichte: Zu Beginn der 1970er-Jahre bekamen die Manager der renommierten Papier-, Druckmaschinen- und Kopierfirma XEROX langsam Angst, denn sie sahen angesichts der schnellen Weiterentwicklung der Computer das Schreckgespenst des »papierlosen Büros« auf sich zukommen. Natürlich weiß man es heute besser; der Papierverbrauch in den Büros hat sich seitdem vervielfacht, weil es einfach und billig ist, Dokumente mal eben auszudrucken und dann auf Papier zu kontrollieren.

Aber damals konnte das natürlich niemand ahnen. Und so beschloss die Firma XEROX: Wenn die Leute im Büro kein Papier mehr brauchen, dann müssen wir ihnen eben das liefern, was sie statt dessen dort benötigen. Dies war der Grundgedanke für die Einrichtung dieses Forscherparadieses. Einige der brilliantesten Köpfe der Computertechnik und Informatik wurden eingeladen, jeweils fünf Jahre lang, mit fast unbegrenzten finanziellen Mitteln ausgestattet, zu erforschen, was immer sie wollten. So kam es, dass hier schon Mitte der 1970er-Jahre Neuerungen wie eine grafische Benutzeroberfläche, die Dokumentenbearbeitung nach dem Prinzip **WYSIWYG** (»What You See Is What You Get«) oder ein Laserdrucker entwickelt wurden.

Im Prinzip existierte dort ein marktreifes Personalcomputer-System, dessen technische Fähigkeiten erst über 15 Jahre später zum allgemeinen Standard wurden. Der einzige Nachteil dieses Systems, des **ALTO**, war sein außerordentlich hoher Preis von ca. 100.000 US-Dollar. Jedenfalls ist XEROX PARC so auch die Wiege des lokalen Netzwerkes geworden: Die Vision war, dass jeder Mitarbeiter eines Unternehmens ein solches Gerät auf seinem Schreibtisch stehen hat und der Austausch von Daten ganz einfach ist. So entstand Ethernet, diejenige Form des lokalen Netzes, die heute am häufigsten genutzt wird.

Sollten Sie sich jetzt wundern, warum keine dieser bahnbrechenden Entwicklungen heute den Namen XEROX trägt, so liegt dies daran, dass die Führungsetage des Unternehmens im fernen New York deren Brillanz und zukunftsweisende Eigenschaften nicht erkannte. So wurden die Entwicklungsergebnisse der meisten Forscher von diesen mitgenommen, jahrelang in die Schublade gelegt und später oftmals zur Gründung anderer Firmen verwendet.

John Warnock etwa, eine Zeit lang Forschungsleiter von PARC, gründete die Firma **Adobe** und konzentrierte sich besonders auf die Bereiche WYSIWYG und den Laserdrucker. Andere Ergebnisse wurden freimütig an Fremde weitergegeben, so etwa die Geheimnisse der grafischen Oberfläche an den **Apple**-Mitbegründer Steve Jobs, der darin die Grundlage für den Computer der Zukunft erkannte. Zusammen mit dem Bastelgenie **Steve Wozniak** entwarf er den ersten verbreiteten »richtigen« Personal Computer auf Mikroprozessorbasis: **Apple II** hieß das 1977 entstandene Gerät, das sich in seinen verschiedenen

Versionen bis 1984 Millionen Mal verkaufte. Dieser Computer definierte für fast zehn Jahre, was ein Homecomputer haben musste: einen eingebauten BASIC-Interpreter, eine fest in das Gerät integrierte Tastatur, gewöhnliche Audiokassetten und später Disketten als Datenspeicher sowie einen Anschluss für die Bildausgabe auf einem gewöhnlichen Fernseher (die privaten Anwender oder kleine Unternehmen konnten sich nicht auch noch einen Monitor leisten, zumal es ihn zu dem Zeitpunkt nur einfarbig gegeben hätte).

**Der IBM-PC** 1981 stieg auch der Großrechner-Multi **IBM** in das Geschäft mit Personal Computern ein. Es war zwar untypisch für dieses Unternehmen, nicht alle Bauteile eines Computers selbst zu entwickeln, aber aus Gründen des Zeitdrucks kauften sie sich die Bestandteile ihres Geräts auf dem freien Markt zusammen. Der IBM-PC war nicht unbedingt besser als seine Vorgänger, aber allein der gute Name des Herstellers schuf das Vertrauen der Wirtschaft und der Industrie, dass PCs eine solide und zukunftssträchtige Technologie seien.

Einige Jahre war IBM Marktführer im Bereich der **Personal Computer**. Da jedoch immer mehr Hersteller kompatible Nachbauten auf den Markt brachten, die zunehmend billiger wurden, verlor der Konzern irgendwann den ersten Platz. Zudem zerstritten sie sich 1990 mit ihrem Betriebssystemlieferanten Microsoft. Seitdem wird nicht mehr von IBM definiert, wie ein »richtiger PC« auszusehen hat, sondern von Intel und Microsoft – wegen der Kombination aus Intel-Prozessoren und Microsoft-Windows-Betriebssystemen werden PCs heute mitunter als »WinTel«-PCs bezeichnet.

**8-Bit-Homecomputer** Parallel blühte in der ersten Hälfte der 1980er-Jahre der Markt mit reinen Homecomputern ohne professionelle Ambitionen. Interessant war auf diesem Gebiet zunächst der ZX81 (siehe Abbildung 1.3), den der Brite Clive Sinclair entwickelt hatte. Trotz seiner bescheidenen Ausstattung mit 1 kByte Arbeitsspeicher, Schwarzweißgrafik und Folientastatur verkaufte sich das Gerät vorzüglich, weil es der erste Computer war, der weniger als 100 £ kostete.



**Abbildung 1.3** Der Sinclair ZX81. Quelle: Wikipedia

Der beliebteste Homecomputer der 8-Bit-Generation wurde allerdings der 1983 auf den Markt gebrachte **Commodore C64**. Eines der wichtigsten Verkaufsargumente war die reichhaltige Auswahl an Software, vor allem Computerspielen. In Abbildung 1.4 wird dieses Gerät gezeigt. Ein nicht ganz so beliebtes, aber durchaus konkurrenzfähiges Gerät war der **Atari 800 XL**; für diesen gab es zwar weniger Spiele, aber dafür war er besser programmierbar. Die Unterschiede zwischen diesen beiden Geräten waren allerdings relativ gering. Beide waren mit dem 8-Bit-Prozessor 6502 von MosTek ausgestattet, verfügten über 64 kByte RAM, eine Grafik mit 320 × 240 Pixeln und 16 Farben (Atari bei einer geringeren Auflösung sogar 256) sowie mehrstimmigen Synthesizer-Sound, der an das angeschlossene Fernsehgerät übertragen wurde. Obwohl die beiden Rechner sich also technisch sehr ähnlich waren, tobten zu jener Zeit »Systemkriege«, die sich ohne Weiteres mit den heutigen PC/Macintosh- oder Linux/Windows-»Glaubenskriegen« vergleichen lassen.<sup>3</sup>

Die nächste Generation von Homecomputern, die in der zweiten Hälfte der 1980er-Jahre erschien, basierte auf dem 16-Bit-Prozessor 68000 von Motorola und seinen Nachfolgern und war mit »richtigen« Betriebssystemen mit grafischer Benutzeroberfläche ausgestattet. Wieder stritten sich Commodore und Atari um den ersten Platz; auch die verschiedenen Modelle von Commodore Amiga und Atari ST waren mit vergleichbaren Features ausgestattet. Schon früher, nämlich 1984, war der ähnlich gestaltete **Apple Macintosh** erschienen; allerdings war er nicht für Heimanwender konzipiert.

16-Bit-Home-  
computer



**Abbildung 1.4** Commodore C64, der berühmte »Brotkasten«.  
Quelle: Wikipedia

<sup>3</sup> Um mich selbst zu outen: Mein eigener erster Rechner war ein ZX81, dem ein Atari 800 XL folgte. Später hatte ich dann sehr lange nur noch »IBM-kompatible« PCs; zwischenzeitlich gehörte auch ein inzwischen leider verstorbener Mac zur Familie. Sämtliche in diesem Buch beschriebenen Betriebssysteme waren oder sind auf diesen Rechnern im Einsatz und kommen prima miteinander aus.

Bemerkenswert ist, dass Standard-PCs erst viele Jahre später mit Multimedia-Fähigkeiten ausgestattet wurden, die auch nur ansatzweise mit Geräten wie Amiga oder ST vergleichbar waren. Noch heute sind diese Rechner bei manchen Künstlern oder Musikern beliebt.

**32-Bit-Rechner** Zu Beginn der 1990er-Jahre wurden die Home- und Personalcomputer nach und nach mit 32-Bit-Prozessoren ausgestattet. Den Anfang machte der IBM-PC/AT mit Intels 80386-Prozessor, dem wie bei den Vorgängermodellen wieder zahlreiche kompatible Nachbauten folgten. Apple ersetzte dagegen die Motorola 680xx-Prozessoren der frühen Macs durch Power PC-Prozessoren, was die Familie der PowerMacs einleitete. Bemerkenswerterweise wurde die Macintosh-Prozessorbasis vor kurzem nochmals gewechselt; statt Power PC werden nun Intel Dual-Core-Prozessoren eingesetzt.

Die Leistungssteigerung der PC- und Mac-Prozessoren sorgte dafür, dass diese Rechner – besonders die günstigeren IBM-kompatiblen PCs – die klassischen Homecomputer verdrängten und nach den Büros auch den Heimbereich eroberten. Dazu wurden insbesondere im Bereich der Multimedia-Fähigkeiten große Fortschritte erzielt. Ein gewisses Problem bestand allerdings darin, dass die Betriebssysteme der frühen 1990er-Jahre nicht für die modernen 32-Bit-Architekturen ausgelegt waren. 1995 brachte Microsoft mit Windows 95 ein massentaugliches Kompromiss-System auf den Markt, das zwar noch immer einen MS-DOS-Unterbau besaß, aber innerhalb der grafischen Oberfläche echtes 32-Bit-Multitasking beherrschte. Erst im Jahr 2001 verabschiedete sich Microsoft mit Windows XP ganz von der MS-DOS-Vergangenheit im Heimsektor; im selben Jahr erschien die erste Version von Apples Neuentwicklung Mac OS X.

Beide Systemwelten hatten übrigens viel weniger Schwierigkeiten beim Wechsel von 32- auf die immer weiter verbreiteten 64-Bit-Architekturen, genauso wenig wie das freie, für viele Plattformen verfügbare Betriebssystem Linux. Näheres zur Geschichte der Betriebssysteme lesen Sie übrigens in Kapitel 5, *Betriebssystemgrundlagen*.

### **Ausblick in die nähere Zukunft**

Offiziell zählen alle heutigen Computer zur vierten Generation, zu den Computern mit Mikroprozessoren. Selbst mächtige Servermaschinen oder Großrechner werden durch zahlreiche parallel betriebene Prozessoren realisiert. Eine offizielle fünfte Generation ist niemals ausgerufen worden.

Dennoch zeichnen sich zurzeit verschiedene Entwicklungen ab, die in Zukunft zu einem Wandel auf dem Computermarkt führen könnten. Die wichtigsten sind Folgende:

- ▶ Es entstehen immer mehr spezielle mobile Geräte, die Teile der Funktion des universellen PCs übernehmen. Dazu gehören Mobiltelefone mit Internet- und Multimedia-Funktionen, PDAs oder spezielle Geräte, die wie eine Armbanduhr oder ein Gürtel getragen werden können oder in Kleidungsstücke eingebaut sind (»Wearable Compu-

ters«). Im gleichen Maße wird das Internet, besonders in seiner drahtlosen Form, immer wichtiger.

- ▶ Über die klassische binäre Elektronik hinaus entstehen neue Ansätze für den Bau von Computern. Dazu gehören beispielsweise DNA-basierte »Bio-Rechner« oder Computer auf der Basis von **Lichtwellenleitern**. Ein weiteres interessantes Projekt ist der **Quantencomputer**: Da die Quantenmechanik besagt, dass ein Teilchen mehrere Zustände zur selben Zeit aufweisen kann, können sogenannte **QBits (Quanten-Bits)** codiert werden, die »1 und 0 gleichzeitig« beinhalten – das Durchprobieren zahlreicher verschiedener Fälle muss nicht mehr nacheinander geschehen, sondern kann gleichzeitig erfolgen.
- ▶ Überraschend ruhig ist es dagegen in den letzten Jahren in der Öffentlichkeit um die »künstliche Intelligenz« geworden. Inzwischen hat sich die Forschung vorläufig von dem Gedanken verabschiedet, das komplexe menschliche Gehirn nachzuahmen. Stattdessen betreibt man erst einmal Grundlagenforschung und simuliert beispielsweise das Zusammenspiel weniger, einzelner Nervenzellen in sogenannten **neuronalen Netzen**.

### 1.2.3 Entwicklung der Programmiersprachen

Damit ein Computer nützliche Aufgaben erledigen kann, muss er programmiert werden. Da in der kurzen Übersicht über die Computergeschichte bereits von Programmiersprachen die Rede war, folgt hier ein kurzer Abriss über ihre Entwicklung.

#### Die Maschinensprache des Prozessors

Die einzige Sprache, die ein Mikroprozessor wirklich versteht, ist seine **Maschinensprache**. Sie besteht aus nichts weiter als Zahlen: Jeder Befehl, den der Prozessor »versteht«, besitzt einen bestimmten numerischen Code. Je nach Art des Befehls folgen auf die Befehlsnummer ein oder mehrere Argumente verschiedener Länge.

Ein Maschinenprogramm ist für Menschen so gut wie unlesbar und schon gar nicht schreibbar. Wenn Sie eine binäre Programmdatei mit einem Texteditor öffnen, werden die gespeicherten Zahlen als Zeichen interpretiert; es erscheint merkwürdiger Zeichensalat, an manchen Stellen unterbrochen von kleinen Textblöcken, wenn das Programm normalen Text enthält.

Angenommen, ein Programm enthält den folgenden Befehl:

```
65 0 0 0 98
```

Bei dem Befehl 65 könnte es sich beispielsweise um die Anweisung handeln, einen bestimmten Wert auf einen Speicherstapel zu legen; der Wert wird als 32-Bit-Ganzzahl angegeben, hier 98. Im Texteditor würde dies etwa so aussehen:

```
A  b
```

Das große A besitzt den Zeichencode 65, das kleine b den Code 98. Die drei Null-Bytes werden in manchen Editoren als merkwürdige Sonderzeichen, in anderen als Leerzeichen angezeigt. Ein Rückschluss auf die tatsächlichen Befehle ist so gut wie unmöglich.

Wenn Sie überhaupt jemals gezwungen sein sollten, Maschinensprachdateien von Hand zu modifizieren (zum Beispiel, um ein Computerspiel zu überlisten), verwenden Sie besser einen **Hex-Editor**, der die einzelnen Werte nicht nur als ASCII-Zeichen, sondern zusätzlich auch hexadezimal darstellt. Hier könnte das Ganze folgendermaßen aussehen:

```
41 00 00 00 52 A b
```

### Assembler – die »benutzerfreundliche« Maschinensprache

Um Maschinensprache halbwegs benutzbar zu machen, wurde der **Assembler** entwickelt. Statt die Befehle mit ihren tatsächlichen Zahlencodes zu schreiben, werden sie durch Kürzel dargestellt, die man sich mehr oder weniger gut merken kann – daher auch der Name »Mnemonics«. In der Regel werden diese Namen für die jeweiligen Assembler-Befehle unmittelbar vom Prozessorhersteller selbst festgelegt, um jegliches Chaos zu vermeiden.

Die Assembler-Sprache ist von Prozessor zu Prozessor völlig verschieden. Jede Prozessorarchitektur versteht ihre ganz eigenen Arten von Befehlen, die entsprechend unterschiedlich in Assembler umgesetzt werden.

Assembler ist sowohl der Name für diese vereinfachte Schreibweise der Maschinensprache als auch der Name für das Programm, das diese Sprache in die eigentliche Maschinensprache umsetzt (im Englischen wird die Sprache allerdings eher als *Assembly Language* bezeichnet).

Das Assembler-Programm führt gegenüber der eigentlichen Maschinensprache oft eine Reihe von Erleichterungen ein. Viele Assembler beherrschen etwa die Definition sogenannter **Makros**: Immer wiederkehrende Abfolgen von Befehlen erhalten einen eindeutigen Namen und können dann später unter diesem Namen aufgerufen werden.

**Assembler – praktischer Einsatz** Assembler wird heutzutage kaum noch zur Programmierung verwendet, zumindest nicht zur Erstellung vollständiger Programme. Wichtige Ausnahmen sind folgende:

1. In Betriebssystemen sind einige der besonders hardwarenahen Kernroutinen im Assembler des jeweiligen Prozessors geschrieben, und zwar vor allem deswegen, damit der gesamte Rest des Systems so weit von der Hardware abstrahiert wird, dass er vollständig in einer höheren Sprache – meist C – geschrieben werden kann. Diese Vorgehensweise wurde in den 1970er-Jahren bei der Implementierung von UNIX entwickelt und gilt noch heute.
2. Auch Gerätetreiber, die zu den wichtigsten Bestandteilen der Betriebssysteme gehören, müssen manchmal in Assembler geschrieben werden.

3. Bestimmte Teile von Computerspielen werden hin und wieder in Assembler geschrieben. Spiele-Programmierern, besonders von schnellen 3-D-Spielen, kommt es vor allem auf Geschwindigkeit an.
4. Besonders systemnahe Computerviren (Bootsekturviren, die den Startbereich eines Datenträgers infizieren, und Programmviren, die ausführbare Programme befallen) sind meistens vollständig in Assembler geschrieben.

### Die ersten höheren Programmiersprachen (Fortran, Cobol, BASIC)

Als praktischen Ersatz für die maschinenorientierten Sprachen wurden seit Mitte der 1950er-Jahre die problem- oder benutzerorientierten Programmiersprachen eingeführt. Ihr Vorrat an möglichen Befehlen und ihre Syntax orientierten sich eher an den Bedürfnissen der Programmierer als an denen des Rechners.

Diese Programmiersprachen müssen in die Maschinsprache des konkreten Prozessors übersetzt werden. Dazu wurden zwei grundlegende Vorgehensweisen entwickelt:

- ▶ Der **Compiler** erzeugt ein dauerhaft lauffähiges Maschinsprachprogramm und speichert es als ausführbares Programm (*binary executable*) ab.
- ▶ Der **Interpreter** übersetzt den Quellcode dagegen Zeile für Zeile; der Code wird also während der Ausführung (zur Laufzeit) übersetzt. Interpretierte Sprachen werden häufig auch als **Skriptsprachen** bezeichnet.

Die erste Generation der höheren Programmiersprachen war sehr einfach. Beispielsweise gab es noch keine echte Programmstrukturierung. Das ursprüngliche **BASIC** verwendete etwa Zeilennummern, zu denen gesprungen werden konnte; Fortran benutzte spezielle Sprungmarken.

Jede dieser frühen Sprachen hatte eine spezielle Ausrichtung oder einen besonderen Verwendungszweck:

- ▶ **Fortran**, entwickelt in den 1950er-Jahren, ist die Abkürzung für »Formula Translator«. Es handelt sich um eine besonders im Hinblick auf mathematische Bedürfnisse geschriebene Sprache. Sie wird zum Teil auch heute noch von Ingenieuren oder Mathematikern gern verwendet.
- ▶ **Cobol** wurde ebenfalls in den 1950er-Jahren entwickelt. Der Name ist die Abkürzung für »Common Business-Oriented Language«; es handelt sich also um eine Sprache für kaufmännische Anwendungszwecke, für Handel und Wirtschaft. Cobol ist eine ziemlich »geschwätzige« Sprache. Es braucht relativ viele Worte, um verhältnismäßig kurze Anweisungen auszudrücken. Beispielsweise können Sie in fast jeder Programmiersprache folgendermaßen den Wert der Variablen *b* durch 7 teilen und das Ergebnis in einer zweiten Variablen namens *a* speichern:

$a = b / 7$

In Cobol wird dagegen folgende Anweisung verwendet:

```
DIVIDE B BY 7 GIVING A
```

- ▶ **BASIC** wurde 1960 am britischen Dartmouth College entwickelt. Der Name steht für »Beginners All-Purpose Symbolic Instruction Code«. Es handelt sich also um eine eher einfache Sprache für Anfänger. Die Sprache fand seit der zweiten Hälfte der 1970er-Jahre, als die neu gegründete Firma Microsoft sie zum ersten Mal für Personal Computer anpasste, eine gewaltige Verbreitung. So gut wie jeder Homecomputer in den 1980er-Jahren hatte irgendeine BASIC-Variante im ROM eingebaut.

Ein BASIC-  
Beispiel

Zur Verdeutlichung sehen Sie hier ein kleines Beispielprogramm in »allgemeinem« BASIC, das fast jeder klassische BASIC-Interpreter verstehen würde:

```
10 PRINT "Wie heißt du?"
20 INPUT A$
30 PRINT A$; " ist ein interessanter Name."
40 PRINT "Noch mal (j/n)?"
50 INPUT J$
60 IF J$ = "j" THEN GOTO 10
```

Die einzelnen Programmzeilen bedeuten Folgendes:

- ▶ Ausgabe des Textes »Wie heißt du?«.
- ▶ Eingabe der Variablen A\$, die durch das Dollarzeichen als String-Variable (Textinhalt) gekennzeichnet wird.
- ▶ Der eingegebene Name wird ausgegeben, gefolgt von »ist ein interessanter Name«.
- ▶ Der Benutzer wird gefragt, ob er einen weiteren Durchgang wünscht.
- ▶ Eingabe der Variablen J\$.
- ▶ Hat J\$ den Wert »j«, geht es weiter bei Zeile 10; das Programm wird erneut ausgeführt.

### Imperative oder prozedurale Programmiersprachen (Pascal, C)

Diese Programmiersprachen erlauben eine **Strukturierung** von Programmen, darüber hinaus ist eine gewisse **Modularisierung** möglich: Programme können in kleinere logische Einheiten eingeteilt werden, Prozeduren oder Funktionen genannt. Diese sind bis zu einem gewissen Grad wiederverwendbar.

**Pascal** wurde seit 1968 von dem Schweizer Mathematikprofessor **Niklaus Wirth** ausdrücklich als Lehrsprache entwickelt. Noch heute ist Pascal eine der beliebtesten Sprachen, um Schülern oder Studenten das Programmieren beizubringen, weil die Sprache zu einer klaren Programmstrukturierung zwingt.

Die Programmiersprache **C** wurde 1971 von **Dennis Ritchie** und **Brian Kernighan** bei AT&T entwickelt, insbesondere, um eine portierbare (auf andere Rechnerplattformen

übertragbare) Version des Betriebssystems UNIX zu schreiben. Ursprünglich wurde C für einen ganz bestimmten Computer und dessen Besonderheiten entwickelt, den DEC PDP-7. Aus diesem Grund ist C erstaunlich nah an den Fähigkeiten von Assembler, ohne so benutzerunfreundlich zu sein wie dies.

Hier sehen Sie zunächst ein einfaches Pascal-Programm:

Ein Pascal-  
Beispiel

```
PROGRAM tagesgruss;
VAR name: STRING;
    zeit: INTEGER;

BEGIN
  writeln ('Hallo. Gib deinen Namen ein!');
  readln (name);
  writeln ('Gib die Uhrzeit ein - nur Stunde!');
  readln (zeit);
  IF zeit < 12 THEN
    writeln ('Guten Morgen ', name)
  ELSE
    IF zeit < 18 THEN
      writeln ('Guten Tag ', name)
    ELSE
      writeln ('Guten Abend', name);
END.
```

Das Programm begrüßt den Benutzer freundlich mit »Hallo«, und dieser wird aufgefordert, seinen Namen einzugeben. Daraufhin wartet es auf die Eingabe, die in der Variablen `name` gespeichert wird. Als Nächstes fragt das Programm nach der Uhrzeit beziehungsweise genauer nach der Stunde.

In den verschachtelten IF-ELSE-Bedingungen wird dann je nach Tageszeit »Guten Morgen«, »Guten Tag« oder »Guten Abend« ausgegeben.

Die Programmiersprache C wird in Kapitel 8, *Grundlagen der Programmierung*, ausführlich behandelt. Trotzdem sehen Sie hier als Kontrast zu Pascal bereits ein kleines C-Beispielprogramm:

Ein C-Beispiel

```
#include <stdio.h>

int main ()
{
  int a, b;
  printf ("Geben Sie die erste Zahl ein: ");
  scanf ("%d", &a);
  printf ("Geben Sie die zweite Zahl ein: ");
  scanf ("%d", &b);
```

```

    if (a < b)
        printf ("%d ist kleiner als %d.\n", a, b);
    else if (a > b)
        printf ("%d ist groesser als %d.\n", a, b);
    else
        printf ("Zweimal die Zahl %d.\n", a);
    return 0;
}

```

Es wird die Eingabe zweier Zahlen erwartet. Anschließend werden die beiden Zahlen verglichen; je nach Größe der beiden Zahlen wird eine entsprechende Meldung ausgegeben.

Die wichtigste Besonderheit ist, dass es kein **Hauptprogramm** gibt wie in Pascal, sondern nur die spezielle Funktion `main()`, die vom Betriebssystem aufgerufen wird. Die weiteren Eigenheiten von C werden, wie gesagt, in Kapitel 8, *Grundlagen der Programmierung*, erläutert.

### Objektorientierte Programmiersprachen (Smalltalk, C++, Java, C#)

In einer objektorientierten Sprache wird in wiederverwendbaren Paketen programmiert, den sogenannten **Klassen**. Eine Klasse ist eine allgemeine Vorlage für die Konstruktion von Objekten. Ein Objekt ist eine Datenstruktur, die selbst Funktionen enthält, um sich auf die gewünschte Art und Weise zu »verhalten«.

Die wichtigsten Vorteile dieser Art der Programmierung sind die Folgenden:

- ▶ **Kapselung.** Datenstrukturen außerhalb eines Objekts können nicht direkt dessen innere Daten manipulieren, sondern nur seine offiziellen Methoden (Schnittstellen nach außen) benutzen. Dies sorgt für ein klares Programmdesign und schützt vor vielen Fehlern.
- ▶ **Vererbung.** Klassen können ihre Eigenschaften und Methoden an »Kind-Klassen« abgeben, in denen nur noch die Unterschiede programmiert werden müssen. Dies beschleunigt die Softwareentwicklung und macht Programme noch einmal erheblich übersichtlicher.

Zu den wichtigsten objektorientierten Programmiersprachen gehören folgende:

- ▶ **Smalltalk** war die erste objektorientierte Sprache überhaupt. Sie wurde in den 1970er-Jahren zur Programmierung der ersten grafischen Benutzeroberfläche entwickelt.
- ▶ **C++** ist die objektorientierte Erweiterung der Programmiersprache C und wurde von **Bjarne Stroustrup** entwickelt. Da C++ abwärts kompatibel mit C ist, wurde die Sprache bald von C-Programmierern eingesetzt; ihre besonderen Vorteile als objektorientierte Sprache wurden aber erst langsam angenommen.

- ▶ **Java** besitzt vor allem die Besonderheit, dass es sich um eine plattformunabhängige Sprache handelt. Sie brauchen ein Java-Programm nur einmal zu kompilieren, es läuft innerhalb eines speziellen Programms für die verschiedenen Plattformen, der virtuellen Java-Maschine (JVM).
- ▶ **C#** ist eine recht neu entwickelte Sprache von Microsoft. Es handelt sich um eine der Sprachen, mit denen Anwendungen für das .NET-Framework entwickelt werden können. Zwar betont Microsoft aus Marketing-Erwägungen stets die Verwandtschaft mit C++, aber dennoch hat C# mehr mit Java gemeinsam als mit C++.

### Logische Programmiersprachen (LISP, Prolog, Logo)

Einen völlig anderen Ansatz als die bisher behandelten Sprachen versprechen die **logischen Programmiersprachen**: Die Grundidee besteht darin, nicht den fertigen Algorithmus einzutippen und mit wechselnden Wertbelegungen vom Computer berechnen zu lassen, sondern das Grundproblem selbst zu formulieren und die Erzeugung einer möglichst optimalen Lösung dem Compiler zu überlassen.

Die Idee der logischen Programmierung stammt bereits aus den 1960er-Jahren. Wegweisend waren die Arbeiten des Linguisten **Noam Chomsky**, der als Erster eine Verbindung zwischen Sprachwissenschaft und Informationstechnik herstellte. Auf dieser Grundlage entwickelten sich sogenannte Expertensysteme, die Fragen durch logische Schlussfolgerungen aus bereits bekannten Informationen beantworten können. Die logischen Programmiersprachen erwiesen sich als ideales Hilfsmittel zur Implementierung solcher Systeme.

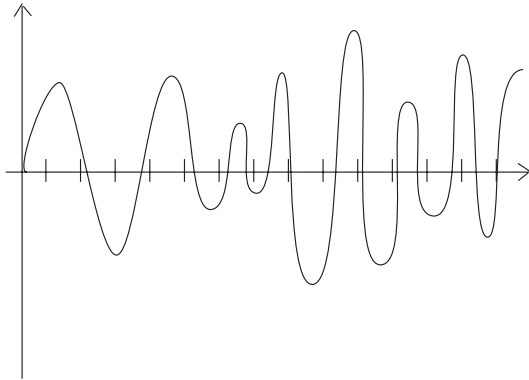
Logische Sprachen im engeren Sinne verwenden die **Prädikatenlogik** zur Formulierung von Ausdrücken oder Termen; der wichtigste Vertreter ist **Prolog**. Sprachen mit ähnlichem Ansatz aber anderer Syntax werden auch als **funktionale Sprachen** bezeichnet; dazu gehört vor allem **Lisp**, aber auch die Sprache **Logo**, die hauptsächlich entworfen wurde, um Kindern die Denkweise der Computerprogrammierung beizubringen. Als gemeinsamen Oberbegriff verwendet man manchmal die Bezeichnung **deklarative Programmiersprachen**.

## 1.3 Digitale Speicherung und Verarbeitung von Informationen

In diesem Abschnitt wird kurz erläutert, wie ein Computer Informationen speichert und verarbeitet. Es besteht hierbei ein grundsätzlicher Unterschied zwischen analogen und digitalen Daten: Analoge Informationen lassen sich in einer kontinuierlichen Wellenform wie in Abbildung 1.5 darstellen, die in immer kleinere Einheiten unterteilt werden können.

In der Natur liegen alle Informationen zunächst in analoger Form vor: Das Bild, das Sie sehen, oder der Ton, den Sie hören, besitzt prinzipiell keine kleinste Informationseinheit oder Auflösung. Mit dieser Art von Informationen kann ein Computer heutiger Bauart nichts anfangen. Die besonderen Eigenschaften der Elektronik haben dazu geführt, dass Computer digital entworfen wurden. »Digital« stammt vom englischen Wort *digit* (Ziffer); dieses Wort ist wiederum vom lateinischen »digitus« (Finger) abgeleitet, da die Finger von jeher zum Zählen eingesetzt wurden.

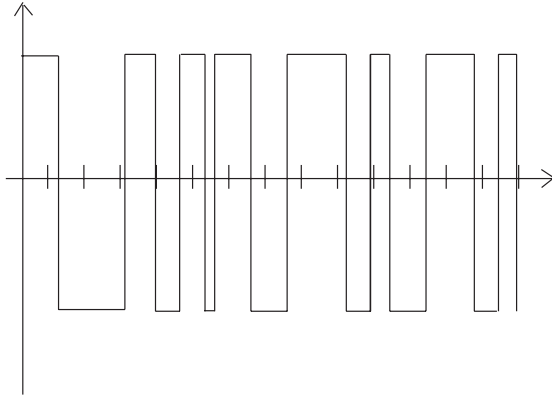
Digital sind Informationen also immer dann, wenn sie in Form von Zahlen dargestellt werden können. Genauer gesagt werden die Daten **binär** gespeichert, das heißt als Abfolge von Einsen und Nullen. Das ist nicht genau dasselbe wie **dual**. Das Dualsystem ist das mathematische Zweiersystem, während binär allgemein die Speicherung beliebiger Daten durch zwei verschiedene Zustände bezeichnet.



**Abbildung 1.5** Schematische Darstellung von Analogdaten als Welle

Die Speicherung von Zahlen erfolgt übrigens in der Tat dual, solange es sich um ganze Zahlen handelt. Eine Besonderheit gilt dabei für vorzeichenbehaftete (positive oder negative) Zahlen, bei denen das vorderste Bit für das Vorzeichen steht. Kompliziert wird es dagegen bei Fließkommazahlen, die in Exponentialschreibweise gespeichert werden. Auf die Darstellung von Zahlen im Computer wird in Kapitel 2, *Mathematische und technische Grundlagen*, näher eingegangen.

Charakteristisch für digitale Daten ist, dass es eine kleinste Informationseinheit gibt und dass die Information nicht mehr weiter aufgelöst werden kann. Während Analogdaten also durch die Wellenform gekennzeichnet sind, lassen sich Digitaldaten durch eine rechteckige Form darstellen (Abbildung 1.6). In der Mathematik werden Objekte mit einem solchen »Mindestabstand« als **diskrete Mengen** bezeichnet. Ein Alltagsbeispiel ist die Folge der ganzen Zahlen – im Unterschied zu den rationalen oder gar reellen Zahlen, bei denen zwischen zwei beliebigen Elementen immer noch unendlich viele Zwischenwerte liegen.



**Abbildung 1.6** Schematische Darstellung binärer Digitaldaten

Die Umwandlung der analogen Eindrücke aus der Realität in computereignete digitale Daten wird als **Digitalisierung** bezeichnet. Je nach Datenart wird sie zum Beispiel von einem Scanner oder einer Digitalkamera bei Bildern oder durch eine Soundkarte bei Tönen durchgeführt. Die folgenden Unterabschnitte sollen einen groben Eindruck davon vermitteln, wie verschiedene Arten von Daten grundsätzlich gespeichert werden.

Digitalisierung

### 1.3.1 Digitale Bilddaten

Wie Hardware zur Bilddigitalisierung funktioniert, also Scanner und Digitalkameras, wird kurz in Kapitel 3, *Hardware*, angeschnitten. Beachten Sie, dass es zwei grundlegende Arten von Computergrafik gibt. Die **Pixelgrafik** (auch **Bitmap-Grafik** genannt), von der hier die Rede ist, speichert ein Bild als rechteckiges Raster quadratischer Farbinformationen ab, den sogenannten **Pixeln**. Die **Vektorgrafik** speichert dagegen Umrisslinien und Kurven von Zeichnungen in Form mathematischer Formeln.

Die Qualität eines gespeicherten Pixelbildes lässt sich durch die folgenden Sachverhalte charakterisieren:

- ▶ Die **Auflösung** gibt die Größe der einzelnen Pixel an. Die Angabe besagt, wie viele Pixel pro Zentimeter beziehungsweise Inch gespeichert werden. Beachten Sie, dass für den Druck erheblich höhere Auflösungen erforderlich sind (etwa 300 Pixel pro Inch) als für eine gleich große Bildschirmfläche (gerechnet wird hier – unabhängig von der tatsächlichen Monitorgröße – mit 72 Pixeln pro Inch).
- ▶ Die **Farbtiefe** gibt an, wie viele Bits zur Speicherung der Informationen eines einzelnen Pixels verwendet werden. Je nach Farbtiefe kann eine bestimmte Anzahl verschiedener Farben eingesetzt werden. Beispielsweise ermöglicht eine Farbtiefe von 8 Bit nur 256 verschiedene Farben, 16 Bit bieten 65.536 Farben und 24 Bit sogar mehr als 16,7 Millionen (genauer gesagt 16.777.216).

- ▶ In der Regel werden die einzelnen Farben nicht stur durchnummeriert, sondern aus einzelnen Grundfarben zusammengesetzt. Ohne hier näher darauf einzugehen, gibt es die **additive** Farbmischung der Lichtfarben Rot, Grün und Blau (RGB) sowie die **subtraktive** Mischung der Druckfarben Cyan, Magenta, Gelb und Schwarz (CMYK). Normalerweise wird für die Intensität jeder einzelnen Grundfarbe ein **Farbkanal** gespeichert. Die Farbtiefe wird dann pro Kanal angegeben, bei einem RGB-Bild mit 24 Bit Farbtiefe also beispielsweise 8 Bit (oder 256 Intensitätsstufen) pro Kanal.

### 1.3.2 Digitale Audiodaten

Töne werden mithilfe eines Verfahrens digitalisiert, das man **Sampling** nennt. Das Audiosignal wird in bestimmten Zeitabständen immer wieder abgetastet, genauer gesagt wird die Amplitude (das Volumen) zum jeweiligen Zeitpunkt gemessen. Die Frequenz (Tonhöhe) ergibt sich dabei aus der zeitlichen Verteilung der Amplituden. Jeder einzelne Abtastvorgang (*Sample*) wird als numerischer Wert abgespeichert.

In der Folge erweist sich die digitalisierte Tonabfolge als diskret in zweifacher Hinsicht: auf der X-Achse **zeitdiskret** durch das gleichbleibende Intervall zwischen den Messungen; auf der Y-Achse **wertdiskret** durch die Beschränkung auf einzelne herausgegriffene (Mittel-)Werte.

Genau wie bei Bildern gibt es auch hier verschiedene Merkmale, die die Datenmenge und die Qualität der gespeicherten Daten betreffen:

- ▶ Die **Sampling-Rate** gibt die Anzahl der Messvorgänge pro Sekunde an. Je höher die Frequenz dieser Messungen, desto höher ist die Tonqualität. Audio-CDs haben beispielsweise eine Sampling-Rate von 44,1 kHz (Kilohertz), es wird also 44.100 mal pro Sekunde gemessen. Multimedia-Produktionen wie Computerspiele oder Infotainment-Titel verwenden dagegen häufig die halbe Sampling-Rate von 22,05 kHz.
- ▶ Die **Sampling-Tiefe** gibt die Datenbreite des einzelnen gespeicherten Tons an, legt also fest, wie viele verschiedene Amplituden unterschieden werden. Bei Audio-CDs sorgt eine Sampling-Tiefe von 16 Bit (65.536 unterschiedliche Werte) für guten Ton; niedrigere Sampling-Tiefen klingen nicht besonders gut.
- ▶ Die Anzahl der **Tonkanäle** besagt, ob irgendeine Art von Raumklang gespeichert wird oder nicht. Wenn Audiodaten mono gespeichert werden, gibt es nur einen einzigen Kanal. Stereo verwendet zwei getrennte Kanäle, die über einen linken und einen rechten Lautsprecher ausgegeben werden können. Eine noch höhere Anzahl von Kanälen wie bei Quadrophonie, Dolby Surround oder 5.1-Sound bewirkt ein noch realistischeres Hörerlebnis.

Tabelle 1.2 stellt die verschiedenen Merkmale von Bilddaten den vergleichbaren Eigenschaften von Tondaten gegenüber, um die entsprechenden Beziehungen zwischen verschiedenen Arten von Digitaldaten zu verdeutlichen.

Informationsart	Bilddaten	Audiodaten
Auflösung	Bildaauflösung, in Pixeln pro Zentimeter oder Inch	Sampling-Rate in Samples pro Sekunde (kHz)
Datenbreite	Farbtiefe	Sampling-Tiefe
Anzahl Kanäle	Farbkanäle	Tonkanäle

**Tabelle 1.2** Vergleich der Merkmale von Bild- und Audiodaten

### 1.3.3 Digitale Speicherung von Text

Reiner Text (nicht der formatierte Text in Textverarbeitungsprogrammen!) wird als Abfolge nummerierter Zeichen eines **Zeichensatzes** gespeichert. Je nach Datenbreite des verwendeten Zeichensatzes können unterschiedlich viele verschiedene Zeichen verwendet werden. Der grundlegende Computerzeichensatz ist noch heute **ASCII**. Es handelt sich um einen 7 Bit breiten Zeichensatz; er enthält also 128 Zeichen. Es sind sämtliche Zeichen für die Darstellung englischsprachiger Texte verfügbar.

Um auch Texte mit den Sonderzeichen der diversen europäischen Sprachen darstellen zu können, werden verschiedene Erweiterungen des ASCII-Codes verwendet. Da die 7 Bit der ASCII-Zeichen gewöhnlich in 8 Bit breiten Feldern gespeichert werden, steht Platz für weitere 128 Zeichen zur Verfügung. Auf diese Weise lassen sich lateinisch geschriebene Sprachen mit Sonderzeichen wie deutschen Umlauten darstellen, aber auch andere Buchstabenalphabete wie Arabisch, Russisch oder Griechisch. Silbenschriften wie Chinesisch oder Japanisch lassen sich dagegen auf diese Weise nicht speichern.

Um viele verschiedene Zeichensätze unter »einen Hut« zu bringen, wurde der **Unicode**-Standard eingeführt. Es handelt sich um einen Zeichensatz mit 16 bis 32 Bit breiten Zeichen; es können also mindestens 65.536 verschiedene Zeichen, in neueren Versionen sogar noch mehr, gespeichert werden. Auf diese Weise bietet Unicode genügend Platz für die Schriftzeichen der meisten Sprachen der Welt sowie für mathematische, technische und andere Sonderzeichen.

Die verschiedenen Zeichensätze werden in Kapitel 15, *Weitere Datei- und Datenformate*, genauer besprochen.

## 1.4 Zusammenfassung

Die Wissenschaft vom Umgang mit Computersystemen, um die es in diesem Buch geht, wird allgemein Informatik genannt. Der praktische Einsatz dieser Systeme heißt dagegen meist Informationstechnik (IT). Es gibt verschiedene Ausbildungsgänge, die auf Berufe in diesem Fachbereich vorbereiten: Zum einen praktische Berufsausbildungen wie die im Titel dieses Buches erwähnten Fachinformatiker, zum anderen die erheblich umfangreicheren, aber theoretischeren Studiengänge.

Die Entwicklung des Computers war letztlich eine Folge des wissenschaftlichen Fortschritts: Die seit Jahrtausenden bekannten Gesetzmäßigkeiten der Mathematik und der Logik wurden mit modernen Erkenntnissen der Elektrotechnik beziehungsweise Elektronik kombiniert, um einen frei programmierbaren Rechenautomat zu erhalten. Die fortschreitende Miniaturisierung und eine effiziente industrielle Fertigung sorgten dafür, dass die Computer sich im Laufe von etwa sechzig Jahren von beinahe unbezahlbaren Einzel-exemplaren an Universitäten zum allgegenwärtigen Werkzeug für fast jeden Anwendungszweck und die meisten Berufe entwickeln konnten.

Parallel zu den Computern selbst hat sich auch die Art und Weise ihrer Programmierung weiterentwickelt: Anfangs musste die reine numerische Maschinensprache mithilfe von Schalttafeln und Steckverbindungen eingegeben werden; nach und nach wurden sowohl die Eingabemethoden als auch der Funktionskomfort der Sprachen selbst erheblich verbessert.

Interessant ist in diesem Zusammenhang, dass sich die innere Funktionsweise der Rechner von Anfang an kaum verändert hat: Bis heute werden elektronisch gesteuerte Schalter verwendet, die die beiden Zustände 0 und 1 darstellen können. Sämtliche Informationen – ob mathematische Informationen, Texte oder Multimedia-Daten – sowie die Manipulation dieser Daten müssen mithilfe dieser Schaltungen codiert werden. Wie dies im Einzelnen geschieht, erfahren Sie im Verlauf dieses Buches noch wesentlich genauer.

# Index

- !, Operator 439
- !=, Operator 440
- !~, Perl-Operator 534

---

- #, Perl-Kommentar 478
- #define, Präprozessor-Direktive 456
- #endif, Präprozessor-Direktive 456
- #ifdef, Präprozessor-Direktive 456
- #include, C-Präprozessor-Direktive 433
- #include, Präprozessor-Direktive 456
- \$, Operator in RegExp 533
- \$, Perl-Variablen 473–474
- \$, PHP-Variablen 895
- \$\_, Perl-Variable 814
- \$\_, Perl-Variable 476
- \$0, UNIX-Systemvariable 357
- %, Operator 439
- %, Perl-Variablen 474
- &&, Operator
  - als Perl-Fallunterscheidung 480
  - C 439
  - Perl 480
- &, Dereferenzierungs-Operator 449
- &, Operator, C 439
- \*, Operator 439
  - RegExp 531
- ++, Operator 441
- +, Operator 439
  - JavaScript-String-Verkettung 943
  - RegExp 531
  - String-Verkettung, Java 461–462
- , Operator 439, 452
- , Operator 441
- ., Perl-Operator 479
- .bashrc, UNIX-Konfigurationsdatei 358
- .NET-Framework 51
- /, Operator 439
- //, Java-Kommentar 463
- /etc/exports, NFS-Konfigurationsdatei 410
- /etc/passwd, Konfigurationsdatei 355
- /etc/profile, UNIX-Konfigurationsdatei 358
- /etc/shadow, Konfigurationsdatei 356
- <a>, HTML-Tag 852
- <applet>, HTML-Tag 877
- <area>, HTML-Tag 859
- <b>, HTML-Tag 848
- <body>, HTML-Tag 842
- <br />, HTML-Tag 843
- <caption>, HTML-Tag 861
- <col>, HTML-Tag 865
- <colgroup>, HTML-Tag 864
- <dl>, HTML-Tag 851
- <dt>, HTML-Tag 851
- <embed>, HTML-Tag 875
- <form>, HTML-Tag 869
- <h1> bis <h6>, HTML-Tags 846
- <head>, HTML-Tag 842
- <html>, HTML-Tag 842
- <i>, HTML-Tag 848
- <img>, HTML-Tag 857
- <input>, HTML-Tag 870
- <li>, HTML-Tag 849
- <map>, HTML-Tag 859
- <meta>, HTML-Tag 843, 877
- <object>, HTML-Tag 876
- <ol>, HTML-Tag 849
- <option>, HTML-Tag 872
- <p>, HTML-Tag 846
- <pre>, HTML-Tag 847
- <script>, HTML-Tag 940
- <select>, HTML-Tag 872
- <style>, HTML-Tag 882
- <table>, HTML-Tag 860
- <tbody>, HTML-Tag 864
- <td>, HTML-Tag 861
- <textarea>, HTML-Tag 872
- <tfoot>, HTML-Tag 864
- <th>, HTML-Tag 861
- <thead>, HTML-Tag 864
- <title>, HTML-Tag 842
- <tr>, HTML-Tag 860
- <u>, HTML-Tag 848
- <ul>, HTML-Tag 849
- =, Operator 440
- ==, Operator 440
- ==~, Operator 534
- ==~, Perl-Operator 483
- ?., Operator 441
- ?, Operator
  - RegExp 531
- @, Perl-Variablen 474
- @\_, Perl 481
- ^, Operator
  - C 440
  - in RegExp 531, 533
- ~, Operatoren 539
- |, Operator 439
  - in RegExp 533
- ||, Operator 439
  - als Perl-Fallunterscheidung 480
  - Perl 480
- ~, Operator 440
- 0900-Dialer 1009
- 1:1-Relation, RDBMS 658
- 1:n-Relation, RDBMS 658
- 1000BaseFL, Ethernet-Standard 197
- 1000BaseTX, Ethernet-Standard 197
- 100BaseT, Ethernet-Standard 196
- 10Base2, Ethernet-Standard 194
- 10Base5, Ethernet-Standard 194
- 10BaseT, Ethernet-Standard 196
- 16-Bit-Anwendungen, unter Win32 307
- 3D Now! (CPU-Befehlserweiterung) 118
- 8.3 (MS-DOS-Dateinamensschema) 303

---

- A**

---

- Abakus 35
- Abfrage, RDBMS
  - Änderungsabfrage 672
  - Auswahlabfrage 660, 672

## Index

- Abfrage, RDBMS (Forts.)  
  *Einfügeabfrage* 672  
  *Löschabfrage* 672  
Abfrage, RDBMS, SQL 661  
Abgeleitete Klasse 466  
Absätze  
  HTML 846  
Absoluter Pfad 298, 301  
Abstract Factory, Entwurfsmuster 642  
accept(), Perl-Funktion 551  
ACCEPT, iptables-Regel 1020  
Access Point (WLAN) 201, 1031  
Access, Datenbank 663  
ACID (Transaktionen) 682  
Action Mailer (Ruby on Rails) 921  
Action Pack (Ruby on Rails) 921  
Action Web Service (Ruby on Rails) 921  
ActionListener 595  
actionPerformed(), AWT-Methode 595  
Active Directory 347, 1031  
Active Record (Ruby on Rails) 921  
Active Server Pages → ASP.NET  
Active Support (Ruby on Rails) 921  
ActivePerl 471  
ActiveX 1031  
ActiveX-Objekte 876  
Ada, Programmiersprache 36  
Adapter, Entwurfsmuster 642  
Addierer (Schaltung) 87  
Addiermaschinen 35  
Addierwerk (Schaltung) 87  
Addition, Operator 439  
Administrator, Windows-Benutzer 335  
Administratordokumentation 629  
Admin-Rechte 1007  
Adobe  
  *PostScript* 823  
Adobe PostScript 161  
Adressbus 114  
  *Wortbreite* 116  
Adressierung, Speicher 78  
ADSL 209  
  *Anschließen* 210  
ADSL2(+) 209  
Advanced Data Guarding (RAID) 145  
AdWare 1010  
afpd, netatalk-Dienst 414  
Aggregatfunktionen, SQL 678  
Agile Softwareentwicklung 631  
AGP 134, 1031  
AI → Künstliche Intelligenz  
AIFF, Audio-Dateiformat 831  
AIX, Betriebssystem 277  
Ajax 981  
  *Antwort verarbeiten* 985  
  *Aspekte* 982  
  *Bibliotheken für* 999  
  *DOM-Einsatz für* 985  
  *JSON* 988, 997  
  *Komplexes Beispiel* 989  
  *Objekt erzeugen* 982  
  *onreadystatechange, Eigenschaft* 984  
  *open(), Methode* 983  
  *PHP-Skript (Server-Antwort)* 986  
  *readyState, Eigenschaft* 984  
  *responseText, Eigenschaft* 985  
  *responseXML, Eigenschaft* 989, 991  
  *send(), Methode* 984  
  *Server-Antwort* 986  
  *XML* 988, 990–991  
  *XMLHttpRequest* 982  
Akteure (UML) 635  
Aktivitätsdiagramme (UML) 638  
Akustikkoppler 170  
Al Chwarismi (arab. Mathematiker) 34  
Algebra  
  *boolesche* 60, 91  
  *Definition* 91  
  *lineare* 91  
  *zur Algorithmandarstellung* 91  
Algorithmen 33, 90, 515  
  *algebraische Darstellung* 91  
  *Anschaulich-sprachliche Darstellung* 91  
  *Berechenbarkeit* 92  
  *Binäre Suche* 523  
  *BubbleSort* 518  
  *Diagrammdarstellung* 91  
  *entwickeln* 515  
  *Komplexität* 93  
Algorithmen (Forts.)  
  *Lineare Suche* 93, 522  
  *O-Notation der Komplexität* 94  
  *Permutationen* 94  
  *Pseudocode-Darstellung* 92  
  *QuickSort* 521  
  *Sortieren* 518  
  *Suche* 522  
Algorithmus 1031  
  *Größter gemeinsamer Teiler (GGT)* 516  
Alias (MacVerknüpfung) 421  
alias, UNIX-Befehl 393  
Aliase (UNIX-Shell) 393  
ALOHANet 193  
Alpha, Prozessor 116, 119  
ALRM, Signal 291  
Altair 8800, früherer Mikrocomputer 40  
ALU 113, 1031  
  *FPU* 113  
AMD 113  
Amiga 43  
Amigos, drei 633  
Amplitude, Audio 54  
Analog, Unterschied zu Digital 51  
Analyse, Software-Engineering 623  
  *Lastenheft* 624  
  *Objektorientierte Analyse* 624  
  *Pflichtenheft* 625  
  *Strukturierte Analyse* 624  
Analytical Engine 35  
Änderungsabfrage 672  
AND-Schaltung 85  
  *Aufbau mit Transistoren* 85  
  *mit einfachen Mitteln nachbauen* 82  
AND-Verknüpfung 61  
Anführungszeichen  
  *PHP* 897  
  *verschiedene, Bedeutung in Perl* 478  
Angewandte Informatik 26  
Animation  
  *Doble Buffering* 589  
  *Java, AWT* 589  
ANSI 1031  
  *C-Standard* 452  
ANSI-C 431  
ANSI-Zeichensatz 816

- Antivirenprogramme 1005
- Antville, Blog-Tool 936
- Anweisungen, C 435
- Anweisungsblock 443
- Anwenderdokumentation 629
- Anwendung
  - DDN-Modell-Schicht* 177
  - OSI-Schicht* 175
- Anwendungsfalldiagramme (UML) 635
- Anwendungsfälle 630
- Anwendungsserver 189
  - Verteilte Anwendungen* 190
- Apache
  - Installation, Windows* 720
  - Xalan* 797
  - Xerces* 800
- API 1031
- Apple
  - Mac OS X* 415
  - Macintosh* 279
  - QuickTime* 416
- Apple II 41, 278
- Apple Macintosh 43
- Apple-Menü
  - Mac OS X* 418
  - Wichtige Befehle* 418
- AppleTalk 266, 1031
  - Einrichten, Mac OS X* 424
  - Protokollstapel* 267
  - Vergleich zu TCP/IP* 268
- Applets, Java 457, 583, 606
- Application Gateway Firewall 1017
- Application Server 189, 1031
  - Java* 573
- apt, Linux-Paketmanager 387
- Aqua, Mac OS X-Oberfläche 277, 417
- Arabische Zahlen 35
- Arbeitsspeicher 111
  - des virtuellen Prozessors* 100
- Arbeitsverzeichnis 298
  - anzeigen, UNIX* 372
  - wechseln, UNIX* 372
  - wechseln, Windows* 322
- Archivdateien 833
  - BZip2* 835
  - GNUZip* 835
  - tar* 387, 833
  - ZIP* 833
- ARCNet
  - Netzwerk* 204
- A-Record (DNS) 751
- ArgoUML, Tool 634
- Arithmetic-Logical Unit → ALU
- Arithmetische Operatoren
  - C* 439
  - SQL* 678
- arithmetisch-logische Einheit → ALU
- ARP 1031
- ARP, TCP/IP-Netzzugang 213
- ARPA 167, 1032
- ARPANet 167
  - Anwendungen* 168
  - MilNet* 169
  - technische Grundidee* 168
  - ursprüngliche Aufgabe* 168
- Arrays 450, 1032
  - Anzahl der Elemente, Perl* 519
  - aus Strings, Perl* 476
  - C* 450
  - Deklaration, C* 450
  - mehrdimensionale, PHP* 899
  - Perl* 474
  - PHP* 896
  - zur C-String-Darstellung* 451
  - zusammenfassen, Perl* 476
- Artificial Intelligence → Künstliche Intelligenz
- AS, SQL-Klausel 678
- ASCII 1032
  - Zeichensatz* 55
- ASCII-Art 847
- ASCII-Code 815
  - Erweiterungen* 816
  - IBM-Erweiterung* 816
  - Steuerzeichen* 815
  - Tabelle* 815
- ASCII-Modus 1032
- ASP.NET 934, 1032
- Assembler 46, 1032
  - Mnemonics* 46
  - Nicht-x86* 121
  - Praktische Anwendung* 46
  - x86-Beispiele* 121
- Asymmetrische Verschlüsselung 1024
- AT&T 48, 276
  - UNIX System V* 276
- AT&T Bell Laboratories 48
- atalkd, netatalk-Dienst 414
- Atari 43, 278
- Atari 800XL 43, 278
- Atari ST 43
- AT-Befehlssatz 1032
- Athlon, Prozessor 113, 118
- ATM (Asynchronous Transfer Mode) 1032
- ATM, Asynchronous Transfer Mode 203
- Atom, Newsfeed-Format 936
- atomar, Information in RDBMS 661
- Attachment (E-Mail) 258
- attrib, Windows-Befehl 322
- Attribute (Windows-Dateien) 302
- Attribute, HTML 841
- Audio
  - zeit- und wertdiskret* 54
- Audio-CD 148
  - über Soundkarte abspielen* 162
- Audio-Dateiformate 831
  - AIFF* 831
  - MP3* 831
  - MP4* 832
  - OGG Vorbis* 832
  - WAV* 831
- Audiodateien 54
  - Sampling* 54
  - Sampling-Rate* 54
  - Sampling-Tiefe* 54
  - Tonkanäle* 54
- Auflicht-Scanner 154
- Auflösung
  - bei der Digitalkamera* 155
  - Bild* 53
  - der Grafikkarte* 156
- Ausbildung 27
  - Fachinformatiker* 28
  - Informatikkaufleute* 29
  - IT-Systemelektroniker* 29
  - IT-Systemkaufleute* 29
  - Prüfung* 30
  - Studienfächer* 31
- Ausdruck 1032
  - Bedingungen* 443
  - C* 437
  - Perl* 477
- Ausgabe
  - in Dateien, C* 454
  - Konsole, C* 453

## Index

Ausgabe (Forts.)  
  *Perl* 473  
Ausgabeeinheit 109  
Ausgabegeräte 139, 156  
  *Drucker* 159  
  *Grafikkarte* 156  
  *Monitor* 157  
Ausgabesteuerung, durch das  
  Betriebssystem 273  
Ausgabeumleitung  
  *in UNIX-Shells* 367  
  *Windows* 321  
Auslagerungsdatei 295  
Ausnahmen  
  *auslösen* 470  
  *FileNotFoundException, Java*  
    470  
  *IOException, Java* 470  
  *Java* 461, 470  
  *Java, IOException* 461  
Aussage 1032  
Aussageformen 59  
Aussagen 58  
  *falsche* 58  
  *mathematische* 58  
  *wahre* 58  
Aussagenlogik 57  
Auswahlabfrage 660, 672  
Authentifizierung  
  *MySQL* 684  
AUTO\_INCREMENT, SQL-Feldop-  
  tion 676  
Autoflush-Modus 549  
Automatentheorien 26, 90  
  *Registermaschine* 99  
  *Turing-Maschine* 96  
Automatisch starten  
  *System V Init* 384  
  *UNIX* 384  
Automatische Variablen 437  
Autonomes System 1032  
Autonomes System (AS), Routing  
  234  
Average Case (Komplexität) 93  
AVI, Video-Dateiformat 832  
awk 471, 529  
AWT 1032  
AWT, Java 582  
  *Ereignisbehandlung* 583

## B

Babbage, Charles 35  
Back Orifice, Backdoor 1008  
Backdoors 1008  
background-attachment, CSS-  
  Angabe 888  
background-color, CSS-Angabe  
  888  
background-image, CSS-Angabe  
  888  
background-repeat, CSS-Angabe  
  888  
Bactick, Operator 539  
Backups 1006  
Band, der Turing-Maschine 96  
bash  
  *.bashrc, Konfigurationsdatei*  
    358  
  */etc/profile, Konfigurationsda-  
    tei* 358  
  *alias* 393  
bash (Borne Again Shell) 358  
BASIC 48  
Basic Input/Output System →  
  BIOS  
Basic Service Set (WLAN) 201  
BASIC, Programmiersprache 40,  
  48  
Basis, bei Stellenwertsystemen 73  
Bastion-Host 1017  
Batch Processing → Stapelverar-  
  beitung  
Batch-Datei, Windows 323  
Baum, Datenstruktur 527  
  *Binärbaum* 527  
Baum-Topologie, Netzwerk 184  
BCD → binary coded decimals  
BEA WebLogic-Server 573  
Beck, Kent 631  
Bedingter Sprung 120  
Befehle  
  *Dateiverwaltung, UNIX* 370  
  *des virtuellen Prozessors* 101  
  *Systemverwaltung, UNIX* 377  
  *Textmanipulation, UNIX* 374  
  *UNIX* 369  
  *Windows-Konsole* 322  
Befehlstabelle 114  
  *der CPU* 114  
Befehlszeiger (CPU-Register) 113

Behavioral Patterns → Verhal-  
  tensmuster  
Bell Laboratories 276  
belongs\_to, Rails-Relation 925  
Bemer, Robert 815  
Benutzer  
  *Administrator, Windows* 335  
  *an- und abmelden, Windows*  
    316  
  *entfernen, Linux* 380  
  *Gruppe hinzufügen, UNIX* 380  
  *hinzufügen, Linux* 380  
  *Home-Verzeichnis* 298, 302  
  *Passwort ändern, UNIX* 381  
  *root* 292, 355  
  *Verwaltung, Mac OS X* 422  
  *Verwaltung, Windows* 335  
  *Zugriffsrechte* 300  
Benutzermodus 284, 290  
Benutzeroberfläche 274  
  *grafische* 274, 279, 289  
  *Konsole* 274, 288  
Benutzerrechte, ändern, UNIX  
  373  
Berechenbarkeit 26, 1032  
Berechenbarkeit von Algorith-  
  men 92  
Berechenbarkeit, Halteproblem  
  92  
Berkeley Socket API 545, 1032  
Berkeley, Universität 276, 395,  
  545  
  *UNIX-Version* 276  
Berners-Lee, Tim 170  
Beschaffungsmanagement 622  
Beschreibbare DVDs 152  
Besitzer, wechseln (Datei) 373  
Best Case (Komplexität) 93  
Betriebssystem  
  *Aufbau* 282  
  *Aufgaben* 273  
  *Benutzermodus* 284  
  *Benutzeroberfläche* 274  
  *Bibliothek* 287  
  *Booten* 284  
  *BSD-UNIX* 276  
  *CP/M* 278  
  *Darwin* 277, 416  
  *Dateiverwaltung* 274  
  *Dialogverarbeitung* 275  
  *Ein- und Ausgabesteuerung* 273

- Betriebssystem (Forts.)
- FreeBSD* 277
  - Gerätetreiber* 285
  - Geschichte* 274
  - herunterfahren* 382
  - HP UX* 277
  - IBM AIX* 277
  - ITS (Incompatible Timesharing System)* 275
  - Kernel* 283
  - Kernelmodus* 284
  - Konsole* 288
  - Linux* 277, 280
  - Mac OS X* 277, 415
  - Minix* 280
  - MS-DOS* 278
  - MULTICS* 276
  - Multitasking* 285, 289
  - Neustart (UNIX)* 382
  - OS/2* 279
  - Prozessmanagement* 273
  - Shell* 288
  - Speichermanagement* 273
  - Stapelverarbeitung* 275
  - Sun Solaris* 277
  - Systemaufrufe* 122, 284, 287
  - Systemprogramme* 286
  - Task Scheduler* 284
  - Threads* 283, 293
  - Timesharing* 275
  - Unicode-Unterstützung* 819
  - UNIX* 49, 276
  - UNIX System V* 276
  - Verwaltungsbefehle, UNIX* 377
  - VMS* 279
  - Win32 API* 288
  - Windows* 305
  - Windows 2000* 280, 306
  - Windows 95* 279, 305
  - Windows 98* 280, 305
  - Windows Me* 280, 305
  - Windows NT* 279, 306
  - Windows Server* 306
  - Windows Vista* 280
  - Windows XP* 280, 306
  - Windows, Versionsübersicht* 306
  - Zeichensatzeinstellung* 819
- Bewegungsdaten 654
- Bezeichner 1032
- C* 436
  - in PHP* 895
- BGP 1032
- BGP, Routing-Protokoll 237
- Bibliothek des Betriebssystems 287
- Big-Endian-Architektur 1033
- Big-Endian-Plattformen 825
- BIGINT, SQL-Datentyp 674
- Bilddateiformate 827
- BMP* 830
  - GIF* 829
  - JPEG* 829
  - Photoshop* 827
  - PICT* 831
  - PNG* 830
  - PSD* 827
  - TIFF* 828
- Bilddaten
- Auflösung* 53
  - digitale* 53
  - Farbkanal* 54
  - Farbtiefe* 53
- Bilddatenbank 655
- Bilder in HTML einbetten 857
- Bildwiederholrate (Monitor) 158
- bin, UNIX-Verzeichnis 298
- Binärbaum, Datenstruktur 527
- C* 527
- Binärdaten 52, 72
- Binäre Suche 523, 1033
- Java* 523
- Binärmodus 1032
- Binary Coded Decimals 1033
- binary coded decimals 82
- BINARY, SQL-Feldoption 676
- bind(), Perl-Funktion 550
- BIND, DNS-Serversoftware 247
- BIND-Nameserver 747
- A-Record* 751
  - CNAME-Record* 752
  - Installation* 748
  - Konfiguration* 748
  - MX-Record* 753
  - NS-Record* 752
  - PTR-Record* 751
  - Reverse-Lookup-Zone* 749
  - SOA-Record* 751
  - Zonendaten-Dateien* 750
  - Zonendefinition* 748
- Bio-Computer 45
- Bioinformatik 32
- BIOS 111, 124, 1033
- Aufgaben* 125
  - beim Macintosh* 129
  - Betriebssystem starten* 126
  - Boot-Reihenfolge* 127
  - CMOS löschen* 126
  - CMOS-RAM* 126
  - eingebaute Routinen* 125
  - POST* 125
  - POST beschleunigen* 127
  - Power Management einstellen* 127
  - Setup* 126
  - Update* 128
- BIOS-Setup 126
- Einstellmöglichkeiten* 126
  - Einstellungen zurücksetzen* 128
- BIOS-Update 128
- Anleitung* 128
  - Motivation* 128
- Bit 78
- Bit-Komplement, Operator 440
- Bitmap-Grafik 53
- Bit-Operatoren 439
- Vergleich mit logischen Operatoren* 62
- Bitübertragung, OSI-Schicht 173
- Bit-Verschiebung, links 440
- Bit-Verschiebung, rechts 440
- Bitweise Exklusiv Oder, Operator 440
- Bitweise Oder, Operator 439
- Bitweise Und, Operator 439
- BLOB, SQL-Datentyp 675
- Block, Anweisungen 443
- Blockgeräte (block devices) 286
- Blocksatz
- in HTML* 846
- Blogs → Weblogs
- Blue Book (Mixed-Mode-CD) 148, 1033
- Bluetooth 138
- Blu-ray Disc 152
- BMP, Bilddateiformat 830
- Booch, Grady 633
- Boole, George 60
- boolean, Java-Datentyp 462
- Boolesche Algebra 60, 91, 1033
- Java-Datentyp* 462
- Booten 284
- BSD-Startskripten* 386

## Index

- Booten (Forts.)
    - Mac OS X-Startvolume* 421
    - System V Init* 384
  - Boot-Reihenfolge einstellen 127
  - Bootsektor 142
  - Bootsektor-Viren 1003
  - border, CSS-Angabe 887
  - BorderLayout 594
  - Borland 608
  - Bourne Again Shell (bash) 358
  - Bourne-Shell 358
  - Boyce-Codd-Normalform, RDBMS 661
  - Branch Prediction → Sprungvorschersage
  - break
    - C-Anweisung* 445
    - Schleifen abbrechen* 522
  - Bridge 196, 1033
  - Bridge, Entwurfsmuster 642
  - Broadcast, IP-Protokoll 215
  - Brute-Force-Attacke 356, 1033
  - BSD 1033
  - BSD-Startskripten 386
  - BSD-UNIX 276
    - FreeBSD* 277
  - BubbleSort, Algorithmus 518
    - Java* 519
  - Buffer im Emacs-Editor 399
  - BufferedReader
    - Java-Klasse* 460, 470
  - BufferedReader, Java-Klasse, readLine(), Methode 461
  - Builder, Entwurfsmuster 642
  - Bundestrojaner → Online-Durchsuchung
  - Bunte Bücher (CD-Standards) 148
  - bunzip2, UNIX-Befehl 836
  - BURN-Proof-Technologie 149
  - Bus 130
    - Bluetooth* 138
    - Definition* 112
    - der CPU* 114
    - drahtloser* 138
    - EIDE* 135
    - FireWire* 137
    - Funk-Schnittstelle* 138
    - Hot-Plugging* 137
    - Infrarot* 138
    - IrDA* 138
    - Kartensteckplätze* 134
  - Bus (Forts.)
    - Laufwerksanschlüsse* 135
    - paralleler* 138
    - PS/2* 138
    - RS-232* 138
    - SCSI* 136
    - serieller* 138
    - USB* 137
  - Bus Mastering 133, 1033
  - Bus-Topologie, Netzwerk 183
  - Button, AWT-Klasse 593
  - Byron, Ada → Lovelace, Ada
  - Byte 78
  - byte, Java-Datentyp 462
  - BZip2, Komprimierung 387
  - bzip2, UNIX-Befehl 835
- ## C
- 
- C 1033
  - C# 51, 1033
  - C++ 50, 1033
  - C, Programmiersprache 48, 276, 430, 439, 448
    - #define, Präprozessor-Direktive* 456
    - #endif, Präprozessor-Direktive* 456
    - #ifdef, Präprozessor-Direktive* 456
    - #include, Präprozessor-Direktive* 433, 456
    - , Operator* 452
    - ANSI-C* 431
    - ANSI-Standard* 452
    - Anweisungen* 433, 435
    - Anweisungsblock* 443
    - Arithmetische Operatoren* 439
    - Arrays* 450
    - Ausdrücke* 437
    - Bezeichner* 436
    - Binärbäume* 527
    - Bit-Operatoren* 439
    - break-Anweisung* 445
    - char* 437
    - Compiler* 431
    - Datentypen* 436
    - Datentypkonvertierung* 438
    - Datum und Uhrzeit* 455
    - difftime()-Funktion* 455
    - double* 437
  - C, Programmiersprache (Forts.)
    - Escape-Sequenz* 434
    - exec()-Funktion* 538–539
    - EXIT\_SUCCESS, Konstante* 434
    - Exponentialschreibweise* 438
    - Fallunterscheidungen* 443
    - fgets()-Funktion* 454
    - Fließkomma-Datentypen* 437
    - Fließkomma-Literale* 438
    - float* 437
    - Flusskontrolle* 440
    - fopen()-Funktion* 454
    - for()-Schleife* 447
    - fork()-Funktion* 537
    - fprintf()-Funktion* 454
    - fscanf()-Funktion* 454
    - Funktionen* 447
    - Funktionsaufrufe* 435, 448
    - Funktionsrückgabewert* 434
    - ganzzahlige Datentypen* 437
    - Geschichte* 431
    - getchar()* 453
    - gets()-Funktion* 434, 453
    - globale Variablen* 437
    - Header-Datei* 452
    - Hexadezimalzahlen* 438
    - int* 437
    - int, Funktionsdatentyp* 433
    - Integer-Literale* 438
    - Kommandozeilenparameter* 449
    - Kommentare* 435
    - kompilieren* 432
    - Kontrollstrukturen* 435, 442
    - Listen* 524
    - Literale* 438
    - localtime()-Funktion* 455
    - Logische Operatoren* 439
    - lokale Variablen* 437
    - long* 437
    - main()-Funktion* 433, 447
    - malloc()-Funktion* 526
    - mem.h* 526
    - NULL* 455
    - Oktalzahlen* 438
    - Präprozessor* 456
    - printf()-Funktion* 434, 453
    - puts()-Funktion* 433, 453
    - return-Anweisung* 434, 447
    - scanf()-Funktion* 445, 453
    - Schleifen* 446
    - short* 437

- C, Programmiersprache (Forts.)
  - Speicher reservieren* 526
  - Standardbibliothek* 287, 452
  - static-Deklaration* 437
  - statische Variablen* 437
  - stddef.h* 437
  - stdio.h* 433, 453
  - stdlib.h* 433
  - strcat()*, Funktion 455
  - strcmp()*-Funktion 454
  - strcpy()*-Funktion 455
  - strftime()*-Funktion 455
  - string.h* 454
  - String-Literale* 438
  - struct* 451
  - Strukturen* 451
  - switch/case* 445
  - Syntax* 434
  - sys/types.h* 538
  - time()*-Funktion 455
  - time.h* 455
  - time\_t*, Datentyp 455
  - Umgang mit Whitespace* 435
  - Variablen* 436
  - Variablen, Gültigkeitsbereich* 437
  - Variablen Deklaration* 433, 435–436
  - Vergleichsoperatoren* 440
  - void*, Funktionsdatentyp 447
  - Vorzeichen in Datentypen* 437
  - wchar\_t* 437
  - Wertzuweisung* 435
  - while()*-Schleife 446
  - Zeichen-Literale* 438
  - Zeiger* 449
- C64 43, 278
- Cache 114
  - bei Festplatten* 146
  - Level 1* 114
  - Level 2* 114
- Call by Reference 449
  - PHP* 902
- Call by Value 449
- Callback-Methode 543, 1033
- Canvas, AWT-Klasse 583
- Capistrano, Rails-Deployment-Tool 933
- CAPTCHAs 1009
- Carbon 1033
  - Mac OS X-API* 417
- Cäsar-Code 1023
- Cascading Style Sheets → CSS
- case-Befehl in Shell-Skripten 390
- CASE-Tools 634
- cat, UNIX-Befehl 374
- catch(), Java 461
- CAV → Konstante Winkelgeschwindigkeit
- CCD
  - bei der Digitalkamera* 155
  - beim Scanner* 154
- C-Compiler 431
  - GCC* 431
- CD
  - beschreibbare* 148
  - Brennsoftware* 149
  - BURN-Proof-Technologie* 149
  - Datenformate* 149
  - Disc-at-once* 149
  - Hybrid-CD* 150
  - ISO-9660-Format* 150
  - Joliet-Format* 150
  - Lead-In-Area* 149
  - Lead-Out-Area* 149
  - Multisession* 149
  - Track-at-once* 149
- cd, UNIX-Befehl 372
- cd, Windows-Befehl 322
- CD-Brennsoftware 149
- CDE, Window-Manager 403
- CDi 148
- CD-R 148
- CD-ROM 147
  - Geschwindigkeit* 148
- CD-RW 148
- CD-Standards 148
  - Blue Book* 148
  - Green Book* 148
  - Orange Book* 148
  - Red Book* 148
  - White Book* 148
  - Yellow Book* 148
- CD-Text 148
- Central Processing Unit → Prozessor
- Centronics-Anschluss 138
- CGI 933, 1033
  - Sicherheitsprobleme* 1016
- Chain of Responsibility, Entwurfsmuster 642
- Chains (iptables) 1019
- char, C-Datentyp 437
- CHAR, SQL-Datentyp 675
- charAt(), Java-Methode 463
- Chatbot Eliza 95
- chgrp, UNIX-Befehl 374
- Child-Prozess 291, 1033
- Chip 37
- Chipsatz 112
- chmod, UNIX-Befehl 373
- chomp, Perl-Funktion 473
- Chomsky, Noam 51
- chown, UNIX-Befehl 373
- chroot-Umgebung 1018
- CHS (Festplattenadressierung) 141
- CIDR, IP-Adressierung 218
- CIDR-Adressierung 1033
- CIFS-Protokoll, Microsoft-Netzwerk 270
- circuit switching 166
- CISC-Prozessor 118
  - Beispiele* 118
- class, Java-Schlüsselwort 460
- CLASSPATH, Umgebungsvariable 458
- Clear-CMOS-Jumper 126
- Client
  - bei Entwurfsmustern* 641
- Client, Netzwerk 185
- close, Perl-Anweisung 482
- closedir, Perl-Anweisung 483
- Cluster (Dateisystem) 296
- CLV → Konstante lineare Geschwindigkeit
- Cmd.exe, WinNT-Shell 319
- CmdLets, PowerShell 324
- CMOS-RAM 126
  - löschen* 126
- CMS → Content-Management
- CMYK-Farbe 54
- cn, LDAP-Attribut 755
- CNAME-Record (DNS) 752
- Cobol 47
- Cobol, Programmiersprache 47
- Cocoa 1034
- Cocoa, Mac OS X-API 416
- Code-Review 628
- Color, AWT-Klasse 586
- color, CSS-Angabe 888
- COM, Microsoft-Komponentenmodell 573

## Index

- Command, Entwurfsmuster 642
- COMMAND.COM, MS-DOS-Shell 319
- Commit (Transaktionen) 663
- COMMIT, SQL-Anweisung 683
- Commodore 43, 278
- Commodore Amiga 43
- Common Gateway Interface → CGI
- Compact Disc 147
- compareTo(), Java-Methode 463
- Compiler 47, 121
  - Java 459
- Composite, Entwurfsmuster 642
- Computer Science → Informatik
- Computer, Definition 33
- Computersystem, schematischer Aufbau 107
- Computerviren → Viren
- connect(), Perl-Funktion 549
- Connection, JDBC-Klasse 698
- console, interaktive Rails-Konsole 926
- Constant Angular Velocity → Konstante Winkelgeschwindigkeit
- Constant Linear Velocity → Konstante lineare Geschwindigkeit
- Content Management
  - papaya CMS 935
- ContentHandler, SAX-Interface 802
- Content-Management 934
  - Drupal 935
  - Joomla! 935
  - TYPO3 934
- Controller (Rails) 928
- Convention over Configuration (Rails-Prinzip) 921
- Cookies 1034
- Cookies, in PHP 912
- Coprozessor → Koprozessor
- copy, Windows-Befehl 322
- CORBA, Komponentenmodell 573
- Cosinuskurve, zeichnen, AWT 586
- COUNT, SQL-Funktion 679
- cp, UNIX-Befehl 371
- CP/M, Betriebssystem 278
- C-Programmiersprache, char, Datentyp 433
- CPU 111
  - alte Bedeutung 112
  - Dual Core 111
- CPU → Prozessor
- CR, Mac-Zeilenumbruch 813
- crack (Passwort-Knackprogramm) 355
- Crackerangriffe 1015
- Crackertools 1018
- CREATE DATABASE, SQL-Befehl 672
- CREATE TABLE, SQL-Befehl 673
- CREATE USER, MySQL-Anweisung 685
- create, Rails-Methode 926
- create\_table, Rails-Methode 924
- CreateProcess(), Windows-Systemaufruf 292
- Creational Patterns → Erzeugungsmuster
- Creator ID, HFS 423
- CRLF, Windows-Zeilenumbruch 813
- Cronjobs 393
- Crosslink-Ethernet-Kabel 196
- Cross-Site-Scripting 1016
- CRT → Röhrenmonitor
- csh (C-Shell) 358
- C-Shell 358
- CSMA/CA 1034
- CSMA/CA, Netzzugangsverfahren 201
- CSMA/CD 1034
- CSMA/CD, Netzzugangsverfahren 193
- CSS 880, 1034
  - <style>, HTML-Tag 882
  - Absatzformatierung 886
  - Abstand vom linken Rand 889
  - Abstand vom oberen Rand 889
  - Anzeigeart 886
  - Aufgabe 881
  - Ausrichtung 886
  - Außenrand 887
  - background-attachment 888
  - background-color 888
  - background-image 888
  - background-repeat 888
  - Bilder 888
- CSS (Forts.)
  - border 887
  - color 888
  - display 886
  - Einzug 886
  - Element (Tag) formatieren 881
  - Externe Dateien 883
  - Farbangaben 884
  - Farben 888
  - Feste Werte 884
  - fett 886
  - font-family 885
  - font-size 885
  - font-style 885
  - font-weight 886
  - Formate dynamisch ändern, DOM 977
  - für XML-Dokumente 796
  - Hintergrundfarbe 888
  - Hintergrund befestigen 888
  - Hintergrund kacheln 888
  - Hintergrundbild 888
  - Innenabstand 887
  - Klassen 882
  - kursiv 885
  - Laufweite 886
  - Layer 888
  - Layer, Beispiele 889
  - left 889
  - letter-spacing 886
  - line-height 886
  - Linien 887
  - margin 887
  - Numerische Werte 884
  - padding 887
  - position 889
  - Positionsart 889
  - Pseudoformate 882
  - Rahmen 887
  - Schriftart 885
  - Schriftgröße 885
  - Selektor 881
  - Stapelreihenfolge 889
  - Struktur 881
  - style, HTML-Attribut 883
  - text-align 886
  - text-decoration 886
  - Textfarbe 888
  - Textformatierung 885
  - text-indent 886
  - top 889

- CSS (Forts.)  
*Unabhängige Stile* 882  
*unterstrichen* 886  
*vertical-align* 886  
*vertikale Ausrichtung* 886  
*Wertangaben* 883  
*Zeilenhöhe* 886  
*z-index* 889
- CSS Vorteile 881
- C-Standardbibliothek 287, 452, 1033
- CUPS 1034
- CUPS, UNIX-Drucksystem 411  
*Scheduler* 411  
*Starten* 411
- Cutler, David 279
- CygWin 431
- D**
- 
- Daemon 382, 1034
- DARPA 167
- Darstellung, OSI-Schicht 175
- Darwin, Betriebssystem 277, 416
- Data Fork, HFS 422
- database.yml, Rails-Konfigurationsdatei 922
- Datagramm 1034
- Datagramm-Sockets 546
- Date, JavaScript-Klasse 954
- DATE, SQL-Datentyp 675
- date, UNIX-Befehl 379  
*Formatierung* 379
- Datei 274  
*Besitzer wechseln, UNIX* 373  
*Gruppe wechseln, UNIX* 374  
*String lesen aus, C* 454  
*Zugriffsrechte ändern* 373
- Dateiattribute 302  
*Windows, ändern* 322
- Dateien  
*Attribute ändern* 322  
*Ausgabe in, C* 454  
*bearbeiten, Perl* 482  
*Eingabe aus, C* 454  
*Kopieren* 421  
*kopieren* 317, 322, 371  
*löschen* 322  
*löschen unter UNIX* 371  
*Öffnen, C* 454  
*Schließen, C* 454
- Dateien (Forts.)  
*Umbenennen* 421  
*umbenennen* 322, 371  
*verarbeiten, Java* 469  
*Verschieben* 421  
*verschieben* 317, 322, 371  
*Verwaltung* 296
- Dateierweiterung 302  
*Anzeige einschalten* 302
- Dateiformate 811  
*AIFF* 831  
*Audio* 831  
*AVI* 832  
*Bilder* 827  
*binäre* 824  
*BMP* 830  
*GIF* 829  
*JPEG* 829  
*MP3* 831  
*MP4* 832  
*MPEG* 833  
*OGG Vorbis* 832  
*PICT* 831  
*PNG* 830  
*PostScript* 823  
*QuickTime* 832  
*Text* 811  
*textbasierte* 821  
*TIFF* 828  
*Video* 832  
*WAV* 831
- Dateihandles, Perl 482
- Dateinamen  
*Endung* 370  
*Endung sichtbar machen* 302  
*Erweiterung* 302, 370  
*Groß- und Kleinschreibung* 299  
*MS-DOS* 303  
*UNIX-Platzhalter* 370  
*unter UNIX* 299, 370  
*unter Windows* 302
- Dateiserver 186
- Dateisystem, virtuelles 297
- Dateisysteme 296  
*Benutzerrechte* 300  
*CD* 149  
*erzeugen* 379  
*ext3* 378  
*FAT 12* 309  
*FAT 16* 309  
*FAT 32* 309
- Dateisysteme (Forts.)  
*Fehlerprüfung (UNIX)* 378  
*inodes* 299  
*Journaling-Funktion* 378  
*Laufwerksbuchstaben* 300  
*Links* 299  
*Linux* 297  
*Mac OS X* 422  
*Mounten* 299  
*mounten* 377  
*NTFS* 310  
*reiserfs* 378  
*UNIX* 297  
*UNIX-Pfadangaben* 298  
*UNIX-Verzeichnisbaum* 297  
*Windows* 300, 309  
*Windows-Pfadangaben* 301  
*Zuordnungseinheiten* 296
- Dateiverwaltung 274, 296  
*unter UNIX* 370
- Dateiviren 1004
- Daten  
*Bewegungsdaten* 654  
*Ordnungsdaten* 654  
*Rechendaten* 654  
*Stammdaten* 653
- Datenanalyse 624
- Datenbank 653  
*1:1-Relation* 658  
*1:n-Relation* 658  
*Abfragen, objektorientierte* 667  
*Access* 663  
*atomare Information* 661  
*Auswahlabfrage* 660  
*Bild- und Multimedia-* 655  
*Boyce-Codd-Normalform* 661  
*Datenarten* 653  
*Datenfeld* 655  
*Datensatz* 656  
*Einzeltable* 654  
*Einzeltable, Definition* 655  
*Einzeltable, Grenzen* 656  
*Entity* 656  
*erzeugen, SQL* 672  
*FileMaker* 663  
*Filterung* 656  
*freie Server* 664  
*Fremdschlüssel* 657  
*Funktionen* 656  
*Grenzen der RDBMS* 664  
*IBM DB2* 664

## Index

- Datenbank (Forts.)
  - Index* 657
  - Join-Abhängigkeiten* 662
  - kommerzielle Server* 664
  - Konsistenz* 657
  - löschen, SQL* 673
  - m:n-Relation* 658
  - Microsoft SQL Server* 664
  - MySQL* 667
  - Normalformen* 661
  - Normalisierung* 661
  - objektorientierte* 655, 664
  - ODBC* 697
  - ODL* 665
  - OQL* 667
  - Oracle* 664
  - PostgreSQL* 664
  - Primärschlüssel* 657
  - Programmierung* 697
  - relationale* 654, 657
  - Relationen* 657
  - Schlüssel* 657
  - Sortieren* 656
  - SQL* 661
  - Suchen* 656
  - Tabelle erzeugen, SQL* 673
  - Tabelle löschen, SQL* 673
  - Transaktionen* 663
  - Typen* 654
  - Volltextdatenbank* 655
  - XML* 655
- Datenbanken
  - Transaktionen* 682
- Datenbanken, SQL 671
- Datenbus 114
  - Wortbreite* 116
- Datenfeld 655
- Datenfernübertragung → DFÜ
- Datenformate 811
  - Text* 811
- Datenkollision 193
- Datenkomprimierung 826
- Datenpakete 166
  - Frames* 192
- Datensatz 656
- Datensicherung 1006
- Datenstrukturen 90, 515, 524
  - Bäume* 527
  - Queue* 524
  - Stack* 524
- Datenträgerverwaltung, Windows 338
- Datentypen
  - boolean, Java* 462
  - ganzzahlige, C* 437
  - in der PowerShell* 329
  - Java* 461
  - Umwandlung, C* 438
  - Variablen, C* 436
- Datenübertragung
  - Paketvermittlung* 166
  - parallele* 130
  - Schaltkreisvermittlung* 166
  - serielle* 130
- Datenübertragung, Geschwindigkeitsmessung 132
- Datenverarbeitung
  - Elektrische* 25
  - Elektronische* 25
  - Manuelle* 25
  - Mechanische* 25
- DATETIME, SQL-Datentyp 675
- Datum und Uhrzeit
  - C* 455
  - date-Befehl, UNIX* 379
  - Differenzen berechnen* 455
  - EPOCH* 288
  - Ermitteln, C* 455
  - formatieren (UNIX)* 379
  - Formatieren, C* 455
  - JavaScript* 954
  - SQL* 675
- DB2, RDBMS 664
- DBMS (Database Management System) 653
- dc-Knoten (LDAP) 754
- DDN-Schichtenmodell 175
  - Anwendungsschicht* 177
  - Host-zu-Host-Transportschicht* 177
  - Internetschicht* 176
  - Netzzugangsschicht* 176
- DDR-RAM 123
- de Icaza, Miguel 403
- De Marco, Tom 624
- Deadlock 293, 1034
- Debian GNU/Linux 352
- DEC 49
- DEC → Digital Equipment Corporation (DEC)
- DECIMAL, SQL-Datentyp 674
- DE-CIX 235
- Decorator, Entwurfsmuster 642
- default
  - switch/case-Vorgabewert* 446
- Default Gateway 1034
- DEFAULT, SQL-Feldoption 676
- defined, Perl 481
- Defragmentierung 1034
- Defragmentierung (Festplatten) 142
- Deklaration 1034
- Deklarative Programmiersprachen 51
- del, Windows-Befehl 322
- DELETE, SQL-Abfrage 681
- Demilitarisierte Zone → DMZ
- DeMorgan-Theorem 64, 1034
- Denial of Service 1015, 1034
- Denial of Service, Netzwerkangriff 344
- Deployment 1034
  - Ruby on Rails* 933
- Dereferenzierung, Perl 477
- Design Patterns → Entwurfsmuster
- dev, UNIX-Verzeichnis 298
- Dezimalsystem 73
  - in Dual umrechnen* 75
  - in Hexadezimal umrechnen* 76
- DFÜ 204
  - Akustikkoppler* 170
  - Mailboxen* 170
  - Online-Dienste* 170
  - Praxis* 204
  - über ISDN* 208
- DFÜ, PPP-Protokoll 205
- DHCP 238, 1034
- DHCP-Server, Windows Server 2003 348
- DHTML 971, 1034
- DHTML-Layer, manipulieren, DOM 978
- Diagrammtypen (UML) 633
- Dialer 1009
- Dialogverarbeitung 275
- Diascanner 155
- Dienste, Windows 338
- diff, UNIX-Befehl 377
- Differential Engine 35
- Differenzmenge 70
- difftime(), C-Funktion 455

- Digital Equipment Corporation (DEC) 39
- Digital Versatile Disc → DVD
- Digital, Unterschied zu Analog 51
- Digitale Signatur 1024
- Digitalisierung 53
- Digitalkamera 155
  - Auflösung* 155
- Dijkstra, Edsger W. 236, 617
- DIMM-Module (RAM) 123
- DIN 1035
- dir, Windows-Befehl 322
- Disc-at-once 149, 1035
- Disjunktion, logische 62
- Diskettenlaufwerk 146
- diskrete Mengen 52
- display, CSS-Angabe 886
- Division, Operator 439
- DMA-Kanal 133, 1035
- DMA-Kanal, Direct Memory Access 133
- DMZ 1017
- dn, LDAP-Attribut 755
- DNA-Computer 45
- DNS 1035
  - BIND, Serversoftware* 247
  - Master- und Slave-Nameserver* 249
  - Subdomain* 247
  - Top-Level-Domain* 247
  - Zone* 247
- DNS (Domain Name System) 246
- DNS-Server, Windows Server 2003 348
- do/while()-Schleife 446
- DocBook 770
- Dock
  - Dock-Menü Mac OS X* 420
  - Mac OS X* 418–419
- DOCTYPE-Angabe
  - XML* 781
- Document Object Model → DOM
- Document Type Definition → DTD
- document.forms, JavaScript 947
- document.images, JavaScript 957
- document.write(), JavaScript-Methode 941
- dojo, Ajax-Bibliothek 999
- Dokumentation 628
  - Administratordokumentation* 629
  - Anwenderdokumentation* 629
  - Entwicklerdokumentation* 629
- Dokumentation, Software-Engineering, Javadoc 611
- Dokumentstruktur
  - HTML* 841
- DOM 807, 1035
  - Ajax-Einsatz von* 985
  - Baumstruktur anzeigen, JavaScript-Anwendung* 974
  - CSS-Formate ändern* 977
  - Document-Objekt* 808
  - Dokumenthierarchie ändern* 979
  - DOMParser* 808
  - für XML* 807
  - getChildNodes(), Methode* 809
  - in JavaScript* 971
  - Kindknoten* 809
  - Klassen importieren* 808
  - Knoten-Eigenschaften* 972
  - Knotentypen* 808
  - Parser* 808
  - Praktische Anwendung, JavaScript* 977
  - Textknoten* 973
- Domain Name System 246
- Domain Name System (DNS)
  - BIND-Nameserver* 747
  - Round-Robin-Verfahren* 752
- Domain-Namen, Schema 247
- DOMParser, Java-Klasse 808
- Don't Repeat Yourself (Rails-Prinzip) 921
- Doppelte Anführungszeichen, Perl 478
- DOS 278
- DoS → Denial of Service 1015
- Double Buffering, Animationstechnik 589
- double, C-Datentyp 437
- DOUBLE, SQL-Datentyp 674
- DoubleWord (DWord) 1035
- DoubleWord, 32 Bit 79
- Download, HTML-Hyperlinks 854
- Drag and Drop 286
- Drahtlose Netzwerke 198
  - Arten* 199
- Drahtlose Netzwerke (Forts.)
  - Gründe für den Einsatz* 199
  - Wireless LAN* 199
- Drahtlose Schnittstellen 138
- drain, Stromausgang des Transistors 83
- DRAM 122
- drawLine(), AWT-Methode 585
- drawOval(), AWT-Methode 585
- drawPolygon(), AWT-Methode 585
- drawRect(), AWT-Methode 585
- drawString, AWT-Methode 588
- Drei Amigos 633
- Dreisatz 71
- Drei-Wege-Handshake, TCP 242
- DROP DATABASE, SQL-Befehl 673
- DROP TABLE, SQL-Befehl 673
- DROP USER, MySQL-Anweisung 689
- DROP, iptables-Regel 1020
- Drucker 159
  - GDI-Drucker* 161
  - im Netzwerk freigeben, Windows* 347
  - Kugelpkopfdruker* 160
  - Laserdrucker* 160
  - LED-Drucker* 161
  - Matrixdrucker* 159
  - Nadeldruker* 159
  - Schriftarten* 161
  - Thermosublimationsdrucker* 161
  - Thermotransferdrucker* 161
  - Tintenstrahldrucker* 160
  - Treiber* 161
  - Typenraddruker* 160
- Druckserver, Print-Server 187
- Drupal, CMS 935
- DSL 209, 1035
  - ADSL* 209
  - ADSL2(+)* 209
  - Anschließen* 210
  - Einrichten, Mac OS X* 424
  - SDSL* 209
  - über Fernseekabel* 210
  - über Satellit* 210
- DSSS, WLAN-Technik 200
- DTD 781, 1035
  - Alternativen angeben* 784
  - Attributdeklaration* 788

## Index

- DTD (Forts.)
    - Attributnotwendigkeit* 790
    - Attributtypen* 788
    - definieren* 782
    - Elementdeklaration* 784
    - Entites aus externen Dateien* 791
    - Entities deklarieren* 791
    - Externe Entity-Deklarationen* 792
    - Häufigkeitsangabe* 785
    - Klammern* 784
  - du, UNIX-Befehl 378
  - Dual-Core-Prozessoren 111
  - Dualsystem 52, 73
    - in Dezimal umrechnen* 76
    - in Hexadezimal umrechnen* 77
    - in Oktal umrechnen* 77
  - Dualzahlen, mit Vorzeichen speichern 80
  - Durchlicht-Scanner 154
  - DVD 151
    - beschreibbare* 152
  - DVD+R 152
  - DVD+RW 152
  - DVD-R 152
  - DVD-RAM 152
  - DVD-ROM 151
    - Dateiformat* 152
    - Geschwindigkeit* 151
  - DVD-RW 152
  - DVI, Dateiformat 823
  - dvips, Dienstprogramm 823
  - DWord → DoubleWord, 32 Bit
  - Dynamic HTML 971
  - Dynamic RAM → DRAM
- E**
- 
- each(), PHP-Funktion 897
  - echo, UNIX-Befehl 374
  - Echte Obermenge 68
  - Echte Teilmenge 68
  - Eclipse 608
    - Installation* 609
    - Java-Entwicklung mit* 609
    - Perspektiven* 610
    - Refactoring* 613
    - Software-Engineering mit* 608
    - Sprachpakete* 609
  - ECMA 1035
  - EDO-RAM 123
  - EDV → Elektronische Datenverarbeitung
  - Effizienz der CPU 117
  - EIDE 135, 1035
    - anschießen* 135
    - im Vergleich zu SCSI* 135
  - Ein-/Ausgabe 274
    - C* 453
    - Dateien, Java* 469
    - Dateien, Perl* 482
    - Dialogverarbeitung* 275
    - Fehler* 461
    - Java* 460
    - Lochkarten* 274
    - Perl* 473
    - Stapelverarbeitung* 275
    - Terminal* 275
    - Timesharing* 275
  - Einfache Anführungszeichen, Perl 478
  - Einfügeabfrage 672
  - Eingabe
    - aus Dateien, C* 454
    - Konsole, C* 453
    - Perl* 473
    - String, C* 453
    - Zeichen, C* 453
  - Eingabeaufforderung 1035
    - UNIX* 357
    - Windows* 322
  - Eingabeaufforderung → Konsole
  - Eingabeeinheit 109
  - Eingabegeräte 139, 152
    - Digitalkamera* 155
    - Maus* 154
    - Scanner* 154
    - Tastatur* 153
  - Eingabesteuerung, durch das Betriebssystem 273
  - Eingabeumleitung
    - in UNIX-Shells* 367
    - Windows* 321
  - Eingabevervollständigung 361
  - Eingabevervollständigung, Windows-Eingabeaufforderung 319
  - Einsteckkarten 134
    - einbauen* 134
  - Einwegverschlüsselung 1024
  - Einzeltabellendatenbank 654
    - Definition* 655
    - Filterung* 656
    - Funktionen* 656
    - Grenzen* 656
    - Sortieren* 656
    - Suchen* 656
  - EJB → Enterprise Java Beans
  - Elektrische Datenverarbeitung 25
  - Elektrische Geräte 36
  - Elektrizität 36
  - Elektromechanik 36
  - Elektronenröhre 36–37
  - Elektronik 36
  - Elektronische Datenverarbeitung 25
  - Elektronische Geräte 36
  - Element einer Menge 67
  - Eliza 95
  - else, C 443
  - elsif, Perl 480
  - Elternklasse 466
  - Emacs, Texteditor 398
    - Befehlseingabe* 399
    - Befehlsschreibweise* 399
    - Buffer* 399
    - erweiterte Funktionen* 401
    - Fenster wechseln* 399
    - löschen* 400
    - markieren* 400
    - Modi* 399
    - Navigation* 400
    - Speichern* 401
    - Suchfunktionen* 400
    - Text ersetzen* 400
  - E-Mail
    - Attachment* 258
    - Entwicklung* 169
    - Hoaxes* 1014
    - HTML-Hyperlinks auf* 855
    - IMAP-Protokoll* 260
    - Kettenmails* 1014
    - MIME-Format* 257
    - Multipart-Nachrichten* 258
    - POP3-Protokoll* 259
    - RFC-822-Nachricht* 257
    - SMTP-Protokoll* 255
    - Spam* 1012
    - Verschlüsselung* 1025
  - E-Mail-Protokolle 255
  - E-Mail-Server 187

- Embedded Systems 112
- Endlicher Automat 96
- Endlosschleifen 522
- Endung (Dateiname) 302, 370
- Engelbart, Douglas 278
- ENIAC, Röhrenrechner 37
- Enterprise Java Beans 573, 1035
  - Client-Interface* 574
  - Deployment* 576
  - Deployment-Deskriptor* 576
  - Entity-Beans* 573
  - Home-Interface* 574
  - Implementierung* 574
  - Message Driven Beans* 574
  - Session-Beans* 573
  - Vorteile* 573
- Enterprise-Anwendung 1035
- Enterprise-Anwendungen 572
- Enterprise-Anwendungen → Verteilte Anwendungen
- Entity, Datenbank 656
- Entity-Referenzen
  - HTML* 844
  - XML* 777, 791
- Entwicklerdokumentation 629
- Entwicklungsprozess, Software 629
  - Agile Entwicklungsprozesse* 631
  - Extreme Programming* 631
  - Unified Process* 630
- Entwicklungsumgebung, integrierte → IDE
- Entwurf, Software-Engineering 626
  - Schnittstellen* 627
  - Standalone-Systeme* 626
  - Verteilte Systeme* 627
- Entwurfsmuster 639
  - Absicht* 640
  - Abstract Factory* 642
  - Adapter* 642
  - Alias* 640
  - Beispiel Singleton* 643
  - Bestandteile* 640
  - Beteiligte* 641
  - Bridge* 642
  - Builder* 642
  - Chain of Responsibility* 642
  - Client* 641
  - Codebeispiele* 641
  - Command* 642
- Entwurfsmuster (Forts.)
  - Composite* 642
  - Decorator* 642
  - Einordnung* 640
  - Einsatzbeispiele* 641
  - Erzeugungsmuster* 640
  - Facade* 642
  - Factory Method* 642
  - Flyweight* 642
  - Implementierung* 641
  - Interpreter* 643
  - Iterator* 643
  - Katalog* 642
  - Konsequenzen* 640–641
  - Lösung* 640
  - Mediator* 643
  - Memento* 643
  - Motivation* 640
  - MVC* 639, 921
  - Name* 640
  - Observer* 643
  - Problem* 640
  - Prototype* 642
  - Proxy* 642
  - Querverweise* 641
  - Singleton* 642
  - Singleton (als Beispiel)* 643
  - State* 643
  - Strategy* 643
  - Struktur* 641
  - Strukturmuster* 640
  - Template Method* 643
  - Verhaltensmuster* 640
  - Verwendungszweck* 640
  - Visitor* 643
  - Zusammenspiel* 641
- ENUM, SQL-Datentyp 676
- Environment → Umgebung
- EPOCH 288
- EPROM 124
- EPS, Dateiformat 824
  - Unterschiede zu PostScript* 824
- eq, Perl-Operator 479
- equals(), Java-Methode 462
- Ereignisanzeige, Windows 337
- Ereignisbehandlung 582
  - Java, AWT* 595
- Ereignisse, AWT 583
- eRuby (embedded Ruby) 928
- Erweiterte Partitionen 144
- Erweiterung (Dateiname) 302, 370
- Erzeugungsmuster 640
- Escape-Sequenz 1035
- Escape-Sequenz
  - C* 434
- Escape-Sequenzen 377
  - in RegExp* 532
- etc, UNIX-Verzeichnis 298
- Ethernet 192, 1035
  - 100BaseFL-Standard* 197
  - 100BaseTX-Standard* 197
  - 100BaseT-Standard* 196
  - 10Base2-Standard* 194
  - 10Base5-Standard* 194
  - 10BaseT-Standard* 196
  - Bridge* 196
  - Crosslink-Kabel* 196
  - CSMA/CD-Verfahren* 193
  - Entwicklung* 41
  - Hardware* 194
  - Hardware-Adresse* 192
  - Hub* 196
  - Koaxialkabel* 194
  - MAC-Adresse* 192
  - Switch* 196
  - Thicknet Coaxial* 194
  - Thinnet Coaxial* 194
  - Twisted-Pair-Kabel* 195
  - Vorläufer ALOHAnet* 193
- Ethernet-Kabeltypen 194
- Ettrich, Matthias 403
- Euklidische Geometrie 91
- EVA-Prinzip 108
- Event Handling 582
- Exabyte 79
- Exceptions
  - auslösen* 470
  - FileNotFoundException* 470
  - IOException* 470
  - Java* 470
- Exceptions, Java 461
  - IOException* 461
- exec(), C-Funktion 538–539
- executeQuery(), JDBC-Methode 699
- exit, UNIX-Befehl 362
- EXIT-SUCCESS, C-Konstante 434
- Exklusiv-Oder 64
- explode(), PHP-Funktion 898
- Exploits 1015

## Index

- Explorer, Ordneransichten, Windows 318
- Explorer, Windows-Dateimanager 316
- Exponentialschreibweise 81  
C 438
- Exponentielle Komplexität 95
- ext3, Linux-Dateisystem 378
- Extended Service Set (WLAN) 202
- extends, Java-Vererbung 466
- extends, PHP-Schlüsselwort 907
- Extensible Markup Language → XML
- Extreme Programming 631  
*Eigenschaften* 632  
*Programmieren in Paaren* 632  
*Test-first-Verfahren* 632, 649
- 
- ## F
- Facade, Entwurfsmuster 642
- Fachinformatiker 28  
*Anwendungsentwicklung* 28  
*Projektarbeit* 618  
*Prüfung* 30  
*Systemintegration* 28
- facsimileTelephoneNumber, LDAP-Attribut 756
- Factory Method, Entwurfsmuster 642
- Fakultät 520
- Falcon, MySQL-Tabellentyp 682
- Fallunterscheidungen  
*Bedingung* 443  
C 443  
*durch logische Operatoren, Perl* 480  
*in Shell-Skripten* 389  
*switch/case* 445  
*unless, Perl* 481
- Fallunterscheidungs-Operator 441
- Falsche Aussagen 58
- false, Java 462
- Farbaddition 54
- Farben  
*in HTML* 884
- Farbkanal, Bild 54
- Farblaserdrucker 161
- Farbsubtraktion 54
- Farbtiefe  
*der Grafikkarte* 156
- Farbtiefe, Bild 53
- Fast Ethernet 196
- FAT 1036
- FAT 12, Dateisystem 309
- FAT 16, Dateisystem 309
- FAT 32, Dateisystem 309
- FAT-Dateisysteme  
*FAT 12* 309  
*FAT 16* 309  
*FAT 32* 309
- FDDI 1036
- FDDI, Netzwerk 203
- Fedora, Linux-Distribution 351
- Feeds → Newsfeeds
- Fenster, programmieren 593
- Fensterbedienung  
*Mac OS X* 418  
*Windows* 312
- Festplatte, belegten Platz ermitteln (UNIX) 378
- Festplatten 141  
*Anschlüsse* 135  
*Cache* 146  
*CHS-Adressierung* 141  
*Defragmentierung* 142  
*EIDE* 135  
*Geschwindigkeit* 146  
*Konstante Winkelgeschwindigkeit* 146  
*LBA-Adressierung* 141  
*Mittlere Zugriffszeit* 146  
*Partitionierung* 142  
*Partitionstabelle* 142  
*RAID* 145  
*SCSI* 136
- Festplattengrößen 79
- Festwertspeicher → ROM
- fg, UNIX-Befehl 359
- fgets(), C-Funktion 454
- FHSS, WLAN-Technik 200
- FIFO 1036
- FIFO → First In, First Out
- FIFO → Queue
- File Type ID, HFS 423
- FileMaker, Datenbank 663
- FileNotFoundException, Java 470
- File-Server 186
- fillOval(), AWT-Methode 585
- fillRect(), AWT-Methode 585
- filter, iptables-Tabelle 1020
- find, Rails-Methode 926
- Finder  
*Mac OS X* 418, 420  
*Ordneransichten, Mac OS X* 420
- finger, UNIX-Dienstprogramm 355
- Firewall 1006  
*iptables* 1017, 1019  
*netfilter (Linux)* 1019
- Firewalls 1017
- FireWire 137, 1036
- First In, First Out 524
- Flachbettscanner 154
- Flags, des virtuellen Prozessors 102
- flash, Rails-Zwischenspeicher 932
- Flash-EPROM 124
- Fließkomma-Literale 438
- Fließkommazahlen 81  
*Exponentialschreibweise* 81  
*Exponentialschreibweise, C* 438
- Flip-Flop (Schaltung) 88
- float, C-Datentyp 437
- FLOAT, SQL-Datentyp 674
- floating point numbers → Fließkommazahlen
- Floating Point Unit → FPU
- FLOPS (CPU-Geschwindigkeit) 117
- FLUSH PRIVILEGES, MySQL-Anweisung 689
- FLUSH TABLES, MySQL-Anweisung 691
- Flussdiagramm 91
- Flusskontrolle 440
- Flyweight, Entwurfsmuster 642
- FM-Synthese (MIDI) 162
- font-family, CSS-Angabe 885
- font-size, CSS-Angabe 885
- font-style, CSS-Angabe 885
- font-weight, CSS-Angabe 886
- fopen(), C-Funktion 454
- for()-Schleife 447
- for-Befehl in Shell-Skripten 390
- foreach, Perl-Schleife 475
- fork(), C-Funktion 537
- fork(), UNIX-Systemaufruf 290
- form\_tag, Rails-Methode 932

- Formale Logik 57
  - Formatieren, Datenträger 379
  - Formulare
    - in Ruby on Rails* 932
  - Fortran 47
  - Fortran, Programmiersprache 47
  - FORWARD, iptables-Chain 1020
  - Foto-Multiplier (Trommelscanner) 155
  - FP-RAM 123
  - fprintf(), C-Funktion 454
  - FPU 113
  - Frame, AWT-Klasse 583, 593
  - Frames, Datenpakete 192
  - FreeBSD 277
  - Frege, Gottlob 57
  - Freie Software 281
  - Fremdschlüssel
    - in Ruby on Rails* 925
  - Fremdschlüssel, RDBMS 657
  - Frequency Hopping 200
  - Frequenz, Audio 54
  - Front Side Bus 116
  - FSB → Front Side Bus
  - fscanf(), C-Funktion 454
  - fsck, UNIX-Befehl 378
  - FTP 252, 1036
    - ASCII-/Binärmodus* 254
    - Befehle* 254
    - Clients* 253
    - HTML-Hyperlinks auf* 855
  - FULLTEXT, SQL-Schlüsselwort 676
  - function, JavaScript-Schlüsselwort 948
  - function, PHP-Schlüsselwort 900
  - Funk- Schnittstelle 138
  - Funktion 48, 1036
    - Rückgabewert, C* 434
  - Funktionale Programmiersprachen 51
  - Funktionen
    - Argumente* 448
    - aufrufen* 448
    - C* 447
    - Call by Reference* 449
    - Call by Value* 449
    - iterative* 520
    - Parameter* 448
    - rekursive* 520
    - Rückgabewert* 447
  - Funktionen (Forts.)
    - SQL* 678
  - Funktionsaufrufe, C 435
  - Fußgesteuerte Schleifen 446
  - fvwm2, Window-Manager 403
- ## G
- 
- Gamma, Erich 608, 639
  - GAN 1036
  - GAN, globales Netz 182
  - Ganze Zahlen 68
  - Garret, Jesse James 982
  - gate, Steuerungseingang des Transistors 83
  - gates (Gatter) → Logikschaltungen
  - Gates, Bill 40, 278
  - Gatter → Logikschaltungen
  - GCC, C-Compiler 431
    - starten* 432
  - gcc-Befehl 432
    - UNIX-Besonderheiten* 432
  - ge, Perl-Operator 479
  - Gerätedateien 297–298
  - Gerätetreiber 46, 273, 285
    - Blockgeräte* 286
    - für Drucker* 161
    - installieren, Windows* 335
    - Netzwerk, Windows* 341
    - Zeichengeräte* 286
  - GET, HTTP-Methode 264
    - zum HTML-Formularversand* 869
  - Get-Alias, PowerShell-CmdLet 324
  - getchar(), C-Funktion 453
  - Get-ChildItem, PowerShell-CmdLet 324
  - Get-Command, PowerShell-CmdLet 324
  - getDate(), JavaScript-Methode 955
  - getDay(), JavaScript-Methode 955
  - getElementById(), JavaScript-Methode 972
  - getElementsByTagName(), JavaScript-Methode 972
  - getFullYear(), JavaScript-Methode 955
  - gethostbyname(), Systemaufruf 547
  - getHours(), JavaScript-Methode 955
  - getMinutes(), JavaScript-Methode 955
  - getprotobyname(), Systemaufruf 547
  - gets(), C-Funktion 434, 453
  - getSeconds(), JavaScript-Methode 955
  - getservbyname(), Systemaufruf 548
  - Getter-Methoden, Java 613
  - getYear(), JavaScript-Methode 955
  - GGT (Größter gemeinsamer Teiler) 516
  - GhostScript 823
  - GID (Group-ID) 355
    - von Prozessen* 291
  - gidNumber, LDAP-Attribut 756
  - GIF, Bilddateiformat 829
  - Gigabit-Ethernet 197
  - Gigabyte 79
  - givenName, LDAP-Attribut 755
  - Gleichheit 65
  - Gleichheit, Operator 440
  - Gleichung 58
    - Lösung* 59
  - Gleichungen, lineare 59
  - Gleitkommazahlen → Fließkommazahlen
  - Global Area Network 182
  - global, PHP-Variablenmodifikation 901
  - Globale Variablen, C 437
  - GNOME
    - Gtk+-Bibliothek* 403
  - GNOME, Desktop 289, 403, 405
    - Desk Guide* 406
    - GNOME-Menü* 406
    - Kontrollzentrum* 407
    - Panel* 407
    - Verknüpfungen erstellen* 407
  - GNU Emacs, Texteditor 398
  - GNU General Public License 281
  - GNU Zip, Komprimierung 387
  - GNU-Projekt 281
  - Gosling, James 457
  - GPL 281

## Index

- Grafikkarte 156  
  AGP 157  
  Auflösung 156  
  Farbtiefe 156  
  Geschwindigkeit 157  
  mit mehreren Monitoren 157  
  PCI 157  
  RAMDAC 157
- Grafische Benutzeroberfläche  
  274, 279, 289, 403, 1036  
  Aqua, Mac OS X 417  
  Drag and Drop 286  
  Ereignisse 582  
  Fensteranwendungen 593  
  GNOME 289, 403, 405  
  GTK+, Programmierumgebung 405  
  JFC 582  
  KDE 289, 403–404  
  Menüs programmieren 593  
  Nachrichten 582  
  Programmierung 582  
  Qt, Programmierumgebung 404  
  Quartz, Grafikbibliothek 416  
  Terminalfenster 356  
  UNIX 401  
  Widgets 582  
  Window-Manager 289  
  Windows 310  
  Windows 95 311  
  Windows 98 311  
  Windows XP 311  
  X Window 289, 401
- GRANT, MySQL-Anweisung 686
- Graphics, AWT-Klasse 583
- Graphics2D, Java2D-Klasse 583
- Green Book (CDI) 148, 1036
- grep, UNIX-Befehl 375  
  Muster 376
- GridLayout 594
- Groß- und Kleinschreibung  
  ignorieren in RegExp 533  
  UNIX-Dateinamen 299
- größer als, Operator 65, 440
- größer oder gleich, Operator 440
- Großrechner 39
- Größter gemeinsamer Teiler 516
- groupadd, UNIX-Befehl 380
- Group-ID 355  
  von Prozessen 291
- Grüner Balken (Unit-Test) 648
- Gruppe  
  hinzufügen, Linux 380  
  wechseln (Datei) 374
- Gruppenrichtlinienobjekte, Windows 338
- gsub, Ruby-Methode 536
- gt, Perl-Operator 479
- GTK+, Grafikbibliothek 403, 405
- GUI → Grafische Benutzeroberfläche
- Gültigkeitsbereich  
  Variablen, C 437  
  Variablen, Java 462
- gunzip, UNIX-Befehl 835
- gzip, Dateikomprimierung 387
- gzip, UNIX-Befehl 835
- 
- ## H
- Halbaddierer (Schaltung) 86
- Halbleiter 38
- Halteproblem (Berechenbarkeit) 92
- Handshake 1036
- Handshake, Modem-Kommunikation 207
- Hard Links 299
- Hardware  
  Ausgabegeräte 139, 156  
  BIOS 111  
  Bus 112, 130  
  Bus Mastering 133  
  Chipsatz 112  
  Digitalkamera 155  
  DMA-Kanal 133  
  Drucker 159  
  Eingabegeräte 139, 152  
  Grafikkarte 156  
  I/O-Basisadresse 133  
  IRQ 132  
  konfigurieren, Windows 335, 337  
  Maus 154  
  Monitor 157  
  Netzwerke 191  
  Onboard-Peripherie 112  
  Peripherie 139  
  Plug and Play 134  
  Prozessor 111  
  RAM 122  
  RAM-Speicher 111
- Hardware (Forts.)  
  Ressourcen 132  
  ROM-Speicher 111  
  Scanner 154  
  schematischer Aufbau 107  
  Schmittstellen 112  
  Soundkarte 162  
  Steuerung durch Betriebssystem 273  
  Tastatur 153  
  Zentraleinheit 111
- Hardware Massenspeicher 139
- Hardware-Interrupt 120
- Harvard Mark I 37
- Harvard Mark II 37
- has\_and\_belongs\_to\_many, Rails-Relation 925
- has\_many, Rails-Relation 925
- Hashes  
  PHP 897  
  Schlüssel 476
- Hashes, Perl-Variablen 474
- Hauptplatine → Mainboard
- Hauptprogramm, main() als 50
- Hauptspeicher 109
- Hayes, Modem-Befehlssatz 206
- Hayes-Befehlssatz 1032
- HD DVD 152
- head, UNIX-Befehl 375
- Header-Datei 452
- Heinemeir Hansson, David 921
- Hello World 432
- Helm, Richard 639
- help, Windows-Befehl 288
- Helper (Rails) 930
- Herunterfahren, Betriebssystem 382
- Herunterfahren, Windows 316
- Hewlett-Packard, HP UX, Betriebssystem 277
- Hexadezimalsystem 74  
  in Dezimal umrechnen 77  
  in Dual umrechnen 77
- Hexadezimalzahlen, C 438
- Hex-Editor 46
- Hex-Editoren 824
- HFS+, Dateisystem 422
- HFS, Dateisystem 422  
  Data Fork 422  
  Resource Fork 423

- HIER-Dokument, in UNIX-Shells 367
- HIER-Dokumente  
*Perl* 479  
*Variablensubstitution* 480
- Hintergrund (Prozesse) 359
- History, UNIX-Shell 361
- History, Windows-Eingabeaufforderung 319
- Hoaxes 1014
- Hollerith, Hermann 38
- home, UNIX-Verzeichnis 298
- Homecomputer 42, 278
- homeDirectory, LDAP-Attribut 756
- Home-Verzeichnis 298, 302
- Host-zu-Host-Transport, DDN-Modell-Schicht 177
- Hot Spots 859
- Hot-Plugging 137
- HP UX, Betriebssystem 277
- HPGL (Druckersprache) 161
- hqx, Dateikomprimierung 423
- HR/DSSS, WLAN-Technik 200
- HTML  
*<a>-Tag* 852  
*<applet>-Tag* 877  
*<area>-Tag* 859  
*<b>-Tag* 848  
*<body>-Tag* 842  
*<br />-Tag* 843  
*<caption>-Tag* 861  
*<col>-Tag* 865  
*<colgroup>-Tag* 864  
*<dl>-Tag* 851  
*<dt>-Tag* 851  
*<embed>-Tag* 875  
*<form>-Tag* 869  
*<head>-Tag* 842  
*<html>-Tag* 842  
*<i>-Tag* 848  
*<img>-Tag* 857  
*<input>-Tag* 870  
*<li>-Tag* 849  
*<map>-Tag* 859  
*<meta>, Tag* 877  
*<meta>-Tag* 843  
*<object>-Tag* 876  
*<ol>-Tag* 849  
*<option>-Tag* 872  
*<p>-Tag* 846
- HTML (Forts.)  
*<pre>-Tag* 847  
*<script>-Tag* 940  
*<select>-Tag* 872  
*<style>-Tag* 882  
*<table>-Tag* 860  
*<tbody>-Tag* 864  
*<td>-Tag* 861  
*<textarea>-Tag* 872  
*<tfoot>-Tag* 864  
*<th>-Tag* 861  
*<thead>-Tag* 864  
*<title>-Tag* 842  
*<tr>-Tag* 860  
*<u>-Tag* 848  
*<ul>-Tag* 849  
*Absatzausrichtung* 846  
*Absätze* 846  
*Absendeknopf* 871  
*ActiveX-Objekte* 876  
*Anker* 856  
*Attribute* 841  
*Aufzählungen* 849  
*Aufzählungszeichen wählen* 849  
*Auswahlmenü* 872  
*Beschreibung für Suchmaschinen* 878  
*Bilder als Hyperlinks* 858  
*Bilder einbetten* 857  
*Blocksatz* 846  
*Body* 841  
*Button* 871  
*Checkboxen* 870  
*clientseitige Image Maps* 859  
*CSS* 880  
*Definitionsliste* 851  
*Dokumentkopf* 841  
*Dokumentkörper* 841  
*Dokumentstruktur* 841  
*Dokumenttitel* 842  
*Download-Hyperlinks* 854  
*E-Mail-Hyperlinks* 855  
*Entity-Referenzen* 844  
*Farben* 884  
*fett* 848  
*Formulardaten-Codierung* 870  
*Formulare* 869  
*Formulare, Absendeknopf* 871  
*Formulare, Auswahlmenü* 872  
*Formulare, Button* 871  
*Formulare, Checkboxen* 870
- HTML (Forts.)  
*Formulare, Hidden-Felder* 871  
*Formulare, Löschknopf* 871  
*Formulare, Passwortfelder* 871  
*Formulare, Radiobuttons* 870  
*Formulare, Reset-Button* 871  
*Formulare, Schaltfläche* 871  
*Formulare, Submit-Button* 871  
*Formulare, Textbereich* 872  
*Formulare, Textfelder* 871  
*Formulare, Versandmethode* 869  
*Formularelemente* 870  
*Formular-URL* 869  
*FTP-Hyperlinks* 855  
*Geeignete Titel* 842  
*Geschütztes Leerzeichen* 845  
*GET, Formularversandmethode* 869  
*Glossarliste* 851  
*h1 bis h6, Tags* 846  
*Head* 841  
*Hidden-Formularfelder* 871  
*Hyperlinks* 852  
*Hyperlinks ins Web* 854  
*Image Maps, client-seitige* 859  
*Image-Maps, serverseitige* 858  
*input-Tag* 870  
*Java-Applets einbetten* 877  
*Kommentare* 860  
*kursiv* 848  
*Layer* 888  
*Layout mit Tabellen* 866  
*Layout-Tags* 847  
*Links* 852  
*Listen* 849  
*Löschknopf* 871  
*Meta-Tags* 877  
*MIME-Types* 875  
*Multimedia einbetten* 875  
*nicht nummerierte Liste* 849  
*nummerierte Liste* 849  
*Nummerierungsart wählen* 850  
*Passwortfelder* 871  
*Pfadangaben in Links* 852  
*Pixel-GIF* 868  
*Plug-in-Formate einbetten* 875  
*POST, Formularversandmethode* 869  
*Radiobuttons* 870  
*Refresh* 878

## Index

- HTML (Forts.)  
  *Reset-Button* 871  
  *robots.txt-Datei* 880  
  *Schaltfläche* 871  
  *Schlüsselwörter für Suchmaschinen* 879  
  *Seite neu laden* 878  
  *Seiteninterne Links* 856  
  *Server-Side-Image Maps* 858  
  *Sonderzeichen* 843  
  *Struktur-Tags* 847  
  *Style Sheets* 880  
  *style-Attribut* 883  
  *Submit-Button* 871  
  *Suchmaschinen-Informationen* 878  
  *Tabellen* 860  
  *Tabellen als Layoutmittel* 866  
  *Tabellen ausrichten* 862  
  *Tabellen, Gitternetzlinien steuern* 864  
  *Tabellen, Rahmenlinien steuern* 864  
  *Tabellen-Attribute* 862  
  *Tabellenbereiche* 864  
  *Tabellenbeschriftung* 861  
  *Tabellenbreite* 862  
  *Tabellenhöhe* 862  
  *Tabellenrahmen* 862  
  *Tabellen-Spaltenbreite* 864  
  *Tabellenzeilen* 860  
  *Tabellen-Zellabstand* 862  
  *Tabellenzellen* 861  
  *Tabellenzellen verbinden* 863  
  *Tabellenzellen-Attribute* 863  
  *Tabellenzellen-Ausrichtung* 863  
  *Tags* 841  
  *Tag-Verschachtelung* 848  
  *Textbereich* 872  
  *Textfelder* 871  
  *Textformatierung* 843  
  *Textmarken* 856  
  *Überschriften* 846  
  *unterstrichen* 848  
  *Verschachtelte Listen* 850  
  *vorformatierter Text* 847  
  *Web-Palette* 885  
  *XHTML* 840  
  *XHTML-Besonderheiten* 843  
  *Zeichenformatierung* 847  
  *Zeichensatz angeben* 843
- HTML (Forts.)  
  *Zeilenumbruch* 843
- HTML-Formulare  
  *per JavaScript modifizieren* 946  
  *Überprüfen per JavaScript* 949
- HTTP 263, 1036  
  *Cookies (PHP)* 912  
  *Header* 264  
  *Session-Tracking (PHP)* 911  
  *Sitzung* 263  
  *Statuscodes* 265
- HTTP-Befehle 264
- HTTP-Server 188
- Hub 196
- HUP, Signal 291
- Hybrid-CD 150, 1036
- Hyperlinks, HTML 852
- Hypertext 170, 1036
- 
- I**
- I/O → Ein-/Ausgabe  
I/O-Basisadresse 133, 1036  
I/O-Kanäle 367
- IANA 216, 1036
- IBM 42, 278  
  *AIX, Betriebssystem* 277  
  *DB2* 664  
  *OS/2, Betriebssystem* 279
- IBM PC 278
- IBM WebSphere-Server 573
- IBM-PC 42
- IC 37
- ICMP 1036
- ICMP-Protokoll 240
- IDE 608  
  *Borland* 608  
  *Eclipse* 608  
  *MS Visual Studio* 608  
  *UCSD-Pascal* 608
- IEEE 1036  
IEEE 1394 1036  
IEEE 1394 → FireWire  
IEEE 802.11 → Wireless LAN  
IEEE, Netzzugangsverfahren 802.x 191
- if(), C 443
- if(), Fallunterscheidung, nachgestellt, Perl 480
- if-Befehl in Shell-Skripten 389
- ifconfig, UNIX-Befehl 407
- Image, AWT-Klasse 589  
  *Bilddatei laden* 589
- Image, JavaScript-Klasse 959
- Imaginäre Zahlen 69
- IMAP 260, 1037
- Imperative Programmiersprachen 48
- Implementierung, Software-Engineering 627
- implode(), PHP-Funktion 898
- import, Java-Anweisung 460
- include(), PHP-Funktion 908
- Index, im RDBMS 657
- INDEX, SQL-Schlüsselwort 676
- indexOf(), Java-Methode 463
- inet\_aton, Netzwerkprogrammierung 547
- inetOrgPerson, LDAP-Objekt-klasse 754
- Informatik 26  
  *Angewandte* 26  
  *Bioinformatik* 32  
  *Medieninformatik* 32  
  *medizinische* 32  
  *Praktische* 26  
  *Studium* 31  
  *Technische* 26  
  *Theoretische* 26  
  *Wirtschaftsinformatik* 32
- Informatikkaufleute 29
- Informationen  
  *analoge* 52  
  *digitale* 52
- Infrarot-Anschluss 138
- init-Prozess 290
- Inner Join, SQL 679  
  *durch WHERE ausdrücken* 680
- INNER JOIN, SQL-Klausel 680
- InnoDB, MySQL-Tabellentyp 682
- InnoDB-Tabellen (MySQL) 663
- inode 299
- INPUT, iptables-Chain 1019
- Input/Output → Ein-/Ausgabe
- InputStreamReader, Java-Klasse 460
- INSERT, SQL-Abfrage 680
- Instanzen 458  
  *erzeugen, Java* 462
- Instruction Table → Befehls-tabelle

- int
  - C-Datentyp* 437
  - Funktionsdatentyp, C* 433
- INT, Signal 382
- INT, SQL-Datentyp 674
- Integer 1037
- Integer, Java-Klasse 518
- Integer-Literale 438
- Integrationsmanagement 621
- Integrierter Schaltkreis 37
- Intel 40, 113
- Intel-Assembler 121
- Interfaces
  - Java* 468
  - Runnable, Java* 469
  - Serializable, Java* 469
- Internet
  - Anwendungsprotokolle* 250
  - Dateiübertragung* 252
  - Denial of Service-Angriff* 344
  - Dokumentation* 169
  - Geschichte* 167
  - Offizielle Einführung* 169
  - RFC* 169
  - Standards* 169
  - Transportprotokolle* 240
  - Vorläufer ARPANet* 167
  - Zugang per Modem* 205
  - Zugang über DSL* 209
  - Zugang über ISDN* 207
- Internet Explorer
  - ActiveX* 876
- Internet, DDN-Modell-Schicht 176
- Internet-Schichtenmodell 175
  - Anwendungsschicht* 177
  - Host-zu-Host-Transportschicht* 177
  - Internetschicht* 176
  - Netzzugangsschicht* 176
- Interpreter 40, 47, 471
- Interpreter, Entwurfsmuster 643
- Inter-Prozess-Kommunikation 292
  - System V IPC* 292
  - über Pipes* 292
  - über Signale* 292
- Interrupt Request → IRQ
- Interrupts, Hardware- 120
- Intrusion Detection Systems 1018
  - Network-IDS* 1018
  - Snort* 1018
- IO::Socket 549, 551–552
- IOException, Java 461, 470
- IP-Adressen
  - Broadcast-Adresse* 215
  - CIDR-Adressierung* 218
  - CIDR-Berechnungen* 220
  - für Sockets* 547
  - IPv6* 226
  - Klassen* 214
  - Konzept* 213
  - link local* 217
  - Loopback* 217
  - Netzwerkadresse* 215
  - private* 217
  - spezielle* 216
  - Subnet Mask* 218
  - Subnetting* 219
  - Supernetting* 219
  - Teilnetzmaske* 218
  - VLSM* 223
  - Zuweisen, Linux* 408
  - Zuweisen, Mac OS X* 424
  - Zuweisen, Windows* 342
- IPC → Inter-Prozess-Kommunikation
- IP-Masquerading 239
- IP-Protokoll
  - Adressverteilung* 216
  - Datagramme* 224
  - Default Gateway* 230
  - Header* 224
  - IPv6* 226
  - Maximum Transmission Unit (MTU)* 226
  - Multicasting* 216
  - Network Address Translation (NAT)* 238
  - Paketfragmentierung* 226
  - Router* 230
  - Routing* 230
  - Routing-Protokolle* 234
  - spezielle Adressen* 216
  - TTL* 234
- IP-Routing
  - Beispiele* 231
  - Routing-Tabelle* 233
- iptables 1017, 1019
  - Beispiele* 1022
  - Chains* 1019
  - Kommandozeilenoptionen* 1021
  - Regeln* 1020
  - Tabellen* 1020
- IPv4-Adressverteilung 216
- IPv6 226
  - Adressaufbau* 227
  - Datagramm-Header* 228
  - Migration (Umstieg)* 229
  - Motivation* 226
  - Tunnelung* 230
- IPX/SPX, Netzwerkprotokoll 269
- irb
  - console, Rails-Variante* 926
- IrDA 138, 1037
- IRQ 132, 1037
  - reserviert* 133
- IS A-Beziehungen, OOP 636
- is\_array(), PHP-Funktion 904
- is\_float(), PHP-Funktion 904
- is\_int(), PHP-Funktion 904
- is\_null(), PHP-Funktion 904
- is\_string(), PHP-Funktion 904
- ISA 135, 1037
- ISDN 207, 1037
  - Anschließen* 208
  - Kanalbündelung* 208
  - Kanäle* 207
- ISO 1037
- ISO 9660 (CD-ROM-Format) 150
- ISO-8859-Zeichensätze 817
- Itanium, Prozessor 116
- IT-Ausbildung 27
  - Fachinformatiker* 28
  - Informatikkaufleute* 29
  - IT-Systemelektroniker* 29
  - IT-Systemkaufleute* 29
  - Projektarbeit* 618
  - Prüfung* 30
  - Studienfächer* 31
- Iteration 520
- Iterator, Entwurfsmuster 643
- ITS, Betriebssystem 275
- IT-Sicherheit 1001
- IT-Systemelektroniker 29
- IT-Systemkaufleute 29

## Index

### J

J2EE 457, 572  
    *Enterprise Java Beans* 573  
J2ME 457  
J2SE 457  
Jacobson, Ivar 633  
Java 1037  
    *Installation* 457  
    *Virtuelle Maschine* 457  
Java Application-Server 573  
Java Foundation Classes 582  
Java Servlets 578  
    *Beispiel* 578  
Java Virtual Machine (JVM) 457  
java, Programm 460  
Java, Programmiersprache 51, 457, 544  
    *ActionListener* 595  
    *Animation, AWT* 589  
    *Applets* 457, 583, 606  
    *Ausnahmen* 461, 470  
    *Ausnahmen auslösen* 470  
    AWT 582  
    *Biddatei laden* 589  
    *Binäre Suche* 523  
    *boolean-Datentyp* 462  
    *BorderLayout* 594  
    *BufferedReader* 460, 470  
    *Button* 593  
    *byte-Datentyp* 462  
    *Callback-Methoden* 543  
    *Canvas* 583  
    *catch()* 461  
    *charAt()-Methode* 463  
    *class-Deklaration* 460  
    CLASSPATH 458  
    *Color* 586  
    *compareTo()-Methode* 463  
    *Connection, JDBC-Klasse* 698  
    *Dateien verarbeiten* 469  
    *Datenbankverbindung herstellen* 698  
    *Datentypen* 461  
    *drawLine()-Methode* 585  
    *drawOval()-Methode* 585  
    *drawPolygon()-Methode* 585  
    *drawRect()-Methode* 585  
    *drawString()-Methode* 588  
    *Eclipse-IDE* 609  
    *Eigenschaften* 458

Java, Programmiersprache (Forts.)  
    *Ein- und Ausgabe* 460  
    *Einsatzgebiete* 457  
    *Enterprise Edition* 572  
    *equals()-Methode* 462  
    *Ereignisbehandlung* 595  
    *Exceptions* 461, 470  
    *executeQuery()-Methode* 699  
    *extends* 466  
    *false* 462  
    *fehlende Zeiger* 463  
    *FileNotFoundException* 470  
    *fillOval()-Methode* 585  
    *fillRect()-Methode* 585  
    *Frame* 583, 593  
    *Getter-/Setter-Methoden* 613  
    *Graphics2D, Klasse* 583  
    *Graphics-Klasse* 583  
    *GridLayout* 594  
    *Gültigkeitsbereich für Variablen* 462  
    *Image-Klasse* 589  
    *import-Anweisung* 460  
    *indexOf()-Methode* 463  
    *init(), Applets* 606  
    *InputStreamReader* 460  
    *Instanzen* 458  
    *Instanzen erzeugen* 462  
    *Integer-Klasse* 518  
    *Interfaces* 468  
    *IOException* 461, 470  
    *Java* 519  
    *java.awt.\** 582  
    *java.awt.event.\** 583  
    *java.lang.\** 463  
    *java.sql.\** 698  
    *java.util.\** 470  
    *Java2D* 583  
    *Javadoc* 611  
    *javax.swing.\** 583  
    *javax.swing.event.\** 583  
    *javax.swing.table.\** 583  
    *JDBC-Datenbankschnittstelle* 697  
    *JDBC-ODBC-Bridge* 697  
    *JDBC-Treiber laden* 698  
    JFC 582  
    *JFrame* 583  
    *Kapselung* 458  
    *Klassen* 458  
    *Kommentare* 463

Java, Programmiersprache (Forts.)  
    *Kompilieren* 459  
    *Konstruktor* 465  
    *Label* 593  
    *lastIndexOf()-Methode* 463  
    *LayoutManager* 594  
    *length()-Methode* 463  
    *Lineare Suche* 523  
    *Listen* 526  
    *Listener* 595  
    *main()-Methode* 460, 584  
    *Math-Klasse* 463  
    *Menu* 593  
    *MenuBar* 593  
    *MenuItem* 593  
    *Methoden* 458  
    *MouseListener* 595  
    *MouseMotionListener* 596  
    *new* 462  
    *Object-Klasse* 463  
    *Objekte* 458  
    *Objektorientierung* 458, 463  
    *org.xml.sax.\** 800  
    *Package java.io.\** 460  
    *paint()-Methode* 583  
    *Panel* 583, 593  
    *parseInt()-Methode* 518  
    *print()* 461  
    *println()* 461  
    *private* 464  
    *Programme starten* 459  
    *protected* 466  
    *public* 460, 465  
    *readLine()-Methode* 461  
    *repaint()-Methode* 592  
    *ResultSet, JDBC-Klasse* 699  
    *RMI (Remote Method Invocation)* 573  
    *run()-Methode* 542  
    *Runnable-Interface* 469, 542  
    SAX 800  
    SDK 457  
    *SDK, Enterprise Edition (J2EE)* 457  
    *SDK, Micro Edition (J2ME)* 457  
    *SDK, Standard Edition (J2SE)* 457  
    *Serializable-Interface* 469  
    *Servlets* 578  
    *setColor()-Methode* 586  
    *setLayout()* 594

- Java, Programmiersprache (Forts.)
- show()*-Methode 584
  - start()*-Methode 542
  - Statement*, JDBC-Klasse 699
  - static* 460
  - String* 461
  - String-Methoden* 462
  - Strings* vergleichen 462
  - String-Verkettung* 461–462
  - substring()*-Methode 463
  - super* 467
  - Swing* 583
  - TextArea* 593
  - TextField* 593
  - this* 465
  - Thread*, Klasse 541
  - Threads* 469, 541
  - throws*-Klausel 470
  - true* 462
  - try* 461
  - try/catch*-Blöcke 461
  - Überladen 465
  - Umwandlung von SQL-Datentypen 700
  - Unterschiede zu C 461
  - Unterstützung durch Mac OS X 417
  - update()*-Methode 592
  - Verblindeklaration 462
  - Vererbung 466
  - WindowListener* 596
- java.applet.\*, Package 583
- java.awt.\*, Package 582
- java.awt.event.\*, Package 583
- java.awt.Graphics, Klasse 583
- java.io.\*, Package 460
- java.lang.\*, Package 463
- java.sql.\*, Package 698
- java.util.\*, Package 470
- Java2 Enterprise Edition 572
- Java2 SDK 457
- Enterprise Edition (J2EE)* 457
  - Micro Edition (J2ME)* 457
  - Standard Edition (J2SE)* 457
- Java2D-API 583
- Java-Applets 583, 606
- in HTML einbetten* 877
  - init()*-Methode 606
- Javadoc 611
- JavaScript 940, 1037
- +*, Operator 943
- JavaScript (Forts.)
- Ajax* 981
  - angepasste Browserfenster öffnen* 966
  - Ausgabe ins Dokument* 941
  - automatisierte Hyperlinks* 966
  - Bilder austauschen* 957
  - Bilder vorausladen* 959
  - Bildschirmgröße* 969
  - Browserweiche* 964
  - charAt()*, *String*-Methode 950
  - CSS-Formate ändern*, DOM 977
  - Date*, Klasse 954
  - document.forms*, Formulare 947
  - document.images* 957
  - document.write()*, Methode 941
  - Dokumenthierarchie ändern*, DOM 979
  - DOM*, Objektmodell 971
  - DOM-Baumstruktur anzeigen* 974
  - DOM-Knoteneigenschaften* 972
  - DOM-Textknoten* 973
  - Event Handler* 945
  - Externe Datei einbinden* 941
  - Fenstereigenschaften* 966
  - Fenster-Methoden* 970
  - Formular, Fokus setzen* 951
  - Formulare überprüfen* 949
  - Formularzugriff* 946
  - function*, Schlüsselwort 948
  - Funktionen* 948
  - Geschichte* 940
  - getDate()*, Methode 955
  - getDay()*, Methode 955
  - getElementById()*, *DOM*-Methode 972
  - getElementsByName()*, *DOM*-Methode 972
  - getFullYear()*, Methode 955
  - getHours()*, Methode 955
  - getMinutes()*, Methode 955
  - getSeconds()*, Methode 955
  - getYear()*, Methode 955
  - history*-Objekt 966
  - Image*, Klasse 959
  - in der Browser-History blättern* 966
  - in HTML einbetten* 940
  - indexOf()*, *String*-Methode 950
  - JSON* 988, 997
- JavaScript (Forts.)
- lastIndexOf()*, *String*-Methode 950
  - length*, *String*-Eigenschaft 949
  - location*-Objekt 966
  - navigator*-Objekt 964
  - onreadystatechange*, *Ajax*-Eigenschaft 984
  - open()*, *Ajax*-Methode 983
  - open()*, *window*-Methode 966
  - parseFloat()*, Methode 944
  - parseInt()*, Methode 944
  - prompt()*-Methode 942
  - readyState*, *Ajax*-Eigenschaft 984
  - Reguläre Ausdrücke* 951
  - responseText*, *Ajax*-Eigenschaft 985
  - responseXML*, *Ajax*-Eigenschaft 989, 991
  - Rollover*, *praxistaugliche Lösung* 959
  - Rollover-Effekt* 958
  - screen*-Objekt 969
  - send()*, *Ajax*-Methode 984
  - setTimeout()*, Methode 956
  - String*, Klasse 949
  - String-Vergleiche* 944
  - String-Verkettung* 943
  - substring*, *String*-Methode 950
  - Timeout* 956
  - Vollbildfenster* 969
- JavaServer Pages 579
- javax.ejb.\*, Java-Package 574
- javax.swing.\*, Package 583
- javax.swing.event.\*, Package 583
- javax.swing.table.\*, Package 583
- Jaz-Laufwerk 147
- JBoss, Java Application-Server 573
- JDBC 1037
- JDBC, Java-Datenbankschnittstelle 697
- JDBC-ODBC-Bridge 697
- JFC 1037
- JFC (Java Foundation Classes) 582
- JFrame, Swing-Klasse 583
- Jobs, Steve 41, 278
- Johnson, Ralph 639
- join, Perl-Anweisung 476

## Index

- Join-Abhängigkeiten, RDBMS 662
  - Joins, SQL 679
  - Joliet 1037
  - Joliet (CD-Format) 150
  - Joliet, Unterstützung auf dem Mac 150
  - Joomla!, CMS 935
  - Journaling-Dateisysteme 378
  - Joy, Bill 395, 457
  - JPEG, Bilddateiformat 829
  - JSON, Ajax-Datenaustauschformat 988, 997
  - json\_encode, PHP-Funktion 997
  - JSP 579
  - JUnit 647
    - Beispiele 647
    - Grafische Oberfläche 648
- ## K
- 
- Kabelanschluss als Internetzugang 210
  - Kapselung 50, 458
  - Kartensteckplätze
    - AGP 134
    - ISA 135
    - PCI 134
    - PCMCIA 135
  - KDE 1037
    - Qt-Bibliothek 403
  - KDE, Desktop 289, 403–404
    - Kontrollzentrum 405
    - Panel 404
    - Verknüpfungen erstellen 405
  - Kernel 283
    - mach-Mikrokern 283, 416
    - Mikrokern 283
    - monolithischer 283
    - selbst kompilieren, Linux 387
    - Systemaufrufe 284, 287
    - Task Scheduler 284
  - Kernelmodus 284, 290
  - Kernighan, Brian 48, 276, 431
  - Kettenmails 1014
  - keys, Perl 477
  - KI → Künstliche Intelligenz
  - Kildall, Gary 278
  - kill(), UNIX-Systemaufruf 291
  - KILL, Signal 291, 382
  - kill, UNIX-Befehl 382
  - Kilobyte 79
  - Kindklasse 466
  - Klammern
    - in RegExp 533
    - Operatoren-Rangfolge verändern 442
  - Klassen 50, 458
    - abgeleitete 466
    - Elternklasse 466
    - Instanzen erzeugen, Java 462
  - Klassendiagramme (UML) 635
  - Klassentests → Unit-Tests
  - Kleinbildscanner 155
  - Kleincomputer 39
  - kleiner als, Operator 65, 440
  - kleiner oder gleich, Operator 440
  - Knuth, Donald E. 822
  - Koaxialkabel 1037
  - KOffice, KDE-Office-Paket 404
  - Kommandozeile 1037
  - Kommandozeile → Konsole
  - Kommandozeilenparameter, C 449
  - Kommentare
    - C 435
    - in HTML 860
    - Java 463
    - Perl 478
    - PHP 900
  - Kommunikation zwischen Prozessen 292
  - Kommunikationsmanagement 621
  - Kommunikationssteuerung, OSI-Schicht 175
  - Komplexe Zahlen 69
  - Komplexität 1037
    - exponentielle 95
    - logarithmische 95
    - O-Notation 94
    - polynomielle 95
    - quadratische 95
  - Komplexität von Algorithmen 93
  - Komplexitätsklasse 93
  - Komponentenmodelle 573
  - Komprimierung 826
    - BZip2 387
    - gzip 387
    - hqx 423
    - verlustbehaftete 826
    - verlustfreie 826
  - Konfigurationsdateien
    - .bashrc (UNIX) 358
    - /etc/profile (UNIX) 358
  - Konjunktion, logische 61
  - Konqueror, KDE-Browser 404
  - Konsole 274, 288
    - Ausgabe, C 453
    - Ausgabeumleitung 367
    - Befehle, Windows 322
    - Eingabe, C 453
    - Eingabeaufforderung 357
    - Eingabeaufforderung, Windows 322
    - Eingabeumleitung 367
    - Eingabevervollständigung 361
    - Pipes 368
    - Praktische Verwendung (UNIX) 353
    - Windows 319
  - Konsole, Starten, Windows 321
  - Konstante lineare Geschwindigkeit 148
  - Konstante Winkelgeschwindigkeit 146
  - Konstanten, symbolische 456
  - Konstruktor, Java 465
    - überladen 465
  - Kontextmenü, Mac OS X 421
  - Kontrollstrukturen 442
    - C 435
    - Fallunterscheidung 443
    - foreach, Perl 475
    - in der PowerShell 328
    - Schleifen 446
    - unless, Perl 481
  - Kooperatives Multitasking 285
  - Kopfgesteuerte Schleifen 446
  - Kopieren
    - Dateien, Mac OS X 421
    - Dateien, UNIX 371
    - Dateien, Windows 317
    - Dateien, Windows-Konsole 322
  - Koprozessor 113
  - Korn Shell (ksh) 358
  - Kostenmanagement 621
  - Kritischer Pfad, Netzplan 623
  - Kryptoanalyse 1023
  - Kryptografie 1023
    - asymmetrische Verschlüsselung 1024
    - Cäsar-Code 1023

- Kryptografie (Forts.)  
*Digitale Signatur* 1024  
*Einwegverschlüsselung* 1024  
*Grundbegriffe* 1023  
*Message-Digest* 1024  
*ROT13* 1023  
*SSH* 1025  
*SSL/TLS* 1025  
*symmetrische Verschlüsselung* 1024
- ksh (Korn Shell) 358  
Kugelpopfdrucker 160  
Künstliche Intelligenz 45
- L**
- 
- l, LDAP-Attribut 756  
Label, AWT-Klasse 593  
Lamarr, Hedy 200  
LAMP-System 894, 1037  
LAN 182, 1038  
*Technische Lösungen* 183  
*Wireless* 198  
Laserdrucker 160  
*Farbe* 161  
Last In, First Out 524  
Lastenheft 624  
lastIndexOf(), Java-Methode 463  
LaTeX 1038  
LaTeX, Satzsprache 821  
*Beispieldokument* 822  
Laufwerke  
*Anschlüsse* 135  
*magnetische* 140  
*magneto-optische* 140  
*optische* 140  
Laufwerksbuchstaben 300  
Laufzeit 47  
Laufzeitbibliothek, C 452  
Lauschende Sockets 550  
LayoutManager  
*BorderLayout* 594  
*GridLayout* 594  
LayoutManager, AWT 594  
LBA (Festplattenadressierung) 141  
LCD 1038  
LCD-Monitor 158  
*Funktionsprinzip* 158  
*Nachteile* 159
- LCD-Monitor (Forts.)  
*TFT* 158  
*Vorteile* 158  
LDAP 753, 1038  
*Active Directory* 753  
*Benutzerkonten abbilden in* 754  
*cn-Attribut* 755  
*dc-Knoten* 754  
*dn, Attribut* 755  
*facsimileTelephoneNumber, Attribut* 756  
*gidNumber, Attribut* 756  
*givenName, Attribut* 755  
*Grundlagen* 754  
*homeDirectory, Attribut* 756  
*inetOrgPerson, Objektklasse* 754  
*l, Attribut* 756  
*loginShell, Attribut* 756  
*mail, Attribut* 756  
*o, Attribut* 756  
*objectClass* 754  
*objectClass, Attribut* 755  
*OpenLDAP* 753  
*Organisationseinheiten* 754  
*ou-Knoten* 754  
*people, Organisationseinheit* 754  
*person, Objektklasse* 754  
*posixAccount, Objektklasse* 754  
*Schema* 754  
*sn, Attribut* 755  
*telephoneNumber, Attribut* 756  
*uid, Attribut* 756  
*uidNumber, Attribut* 756  
*userPassword, Attribut* 756  
*verschiedene Server für* 753  
*Wurzel* 754
- le, Perl-Operator 479  
Lead-In-Area (CD) 149  
Lead-Out-Area (CD) 149  
LED-Drucker 161, 1038  
Leere Menge 70  
left, CSS-Angabe 889  
Leibniz, Gottfried Wilhelm 35  
Leichtgewichtige Entwicklungsprozesse 631  
length(), Java-Methode 463  
Lerdorf, Rasmus 893  
less, UNIX-Befehl 375  
letter-spacing, CSS-Angabe 886  
Level-1-Cache 114
- Level-2-Cache 114  
LF, UNIX-Zeilenumbruch 813  
Lichtfarben, RGB 54  
Lichtwellenleiter 45  
LIFO 1038  
LIFO → Last In, First Out  
Lightweight Directory Access Protocol → LDAP  
LIKE, SQL-Klausel 678  
Lineare Algebra 91  
Lineare Geschwindigkeit, konstante 148  
Lineare Gleichungen 59  
Lineare Komplexität 94  
Lineare Suche 93, 1038  
*Java* 523  
line-height, CSS-Angabe 886  
Linux 277, 280  
*\$0, Systemvariable* 357  
*.bashrc, Konfigurationsdatei* 358  
*/etc/passwd-Datei* 355  
*/etc/profile, Konfigurationsdatei* 358  
*/etc/shadow, Datei* 356  
*alias-Befehl* 393  
*als Server einrichten* 409  
*apt, Paketmanager* 387  
*Arbeitsverzeichnis anzeigen* 372  
*auf NFS-Freigaben zugreifen* 410  
*auf Windows-Server zugreifen* 413  
*bash* 358  
*Befehle regelmäßig ausführen* 393  
*Benutzerrechte* 300  
*Bourne-Shell* 358  
*bunzip2-Befehl* 836  
*bzip2-Befehl* 835  
*BZip2-Komprimierung* 387  
*cat-Befehl* 374  
*cd-Befehl* 372  
*chgrp-Befehl* 374  
*Child-Prozess* 291  
*chmod-Befehl* 373  
*chown-Befehl* 373  
*cp-Befehl* 371  
*Cronjobs* 393  
*C-Shell* 358  
*CUPS, Drucksystem* 411

## Index

### Linux (Forts.)

*Daemon* 382  
*date-Befehl* 379  
*Dateibefehle* 370  
*Dateibesitzer wechseln* 373  
*Dateien kopieren* 371  
*Dateien löschen* 371  
*Dateien umbenennen* 371  
*Dateien verschieben* 371  
*Datei-Endungen* 370  
*Dateigruppe wechseln* 374  
*Dateinamen* 299, 370  
*Dateinamen-Platzhalter* 370  
*Dateisysteme* 297  
*Datum und Uhrzeit ändern* 379  
*Datum und Uhrzeit formatieren* 379  
*Debian-Distribution* 352  
*diff-Befehl* 377  
*Distributionen* 281  
*du-Befehl* 378  
*echo-Befehl* 374  
*Emacs, Texteditor* 398  
*Escape-Sequenzen* 377  
*exit-Befehl* 362  
*fg-Befehl* 359  
*finger, Dienstprogramm* 355  
*fork(), Systemaufruf* 290  
*fsck-Befehl* 378  
*Gerätedateien* 297–298  
*GNOME* 289, 403, 405  
*GNU-Projekt* 281  
*Grafische Benutzeroberfläche* 401  
*grep-Befehl* 375  
*groupadd-Befehl* 380  
*Group-ID* 291, 355  
*gunzip-Befehl* 835  
*gzip-Befehl* 835  
*gzip-Komprimierung* 387  
*Hard Links* 299  
*Hardware-Plattformen* 280  
*head-Befehl* 375  
*HIER-Dokument* 367  
*Home-Verzeichnis* 298  
*ifconfig-Befehl* 407  
*init-Prozess* 290  
*inodes* 299  
*Installation von Software* 386  
*IP-Adresse zuweisen* 408  
*KDE* 289, 403–404

### Linux (Forts.)

*Kernel kompilieren* 387  
*Kernel-Module laden* 387  
*kill(), Systemaufruf* 291  
*kill-Befehl* 382  
*Korn Shell* 358  
*less-Befehl* 375  
*Links (Dateisystem)* 299  
*logger-Befehl* 384  
*Login* 353  
*ls-Befehl* 372  
*Macintosh im Netz bedienen* 414  
*mail-Befehl* 394  
*make-Befehl* 387  
*man-Befehl* 288  
*Mandriva-Distribution* 352  
*mkdir-Befehl* 373  
*mkfs-Befehl* 379  
*modprobe-Befehl* 387  
*more-Befehl* 375  
*mount-Befehl* 377  
*mv-Befehl* 371  
*MySQL-Installation* 667  
*netatalk-Server* 414  
*netfilter, Kernel-Firewall* 1019  
*Netzwerkkonfiguration* 407  
*Neustart* 382  
*NFS* 410  
*Pager* 375  
*Paketmanager* 386  
*Parent-Prozess* 291  
*passwd-Befehl* 381  
*Passwort ändern* 381  
*patch-Befehl* 377  
*PATH, Umgebungsvariable* 360  
*pause(), Systemaufruf* 291  
*Pfadangaben* 298  
*Pipes* 368  
*Programme automatisch starten* 384  
*Prozessmodell* 290  
*Prozessverwaltung* 381  
*ps-Befehl* 292, 381  
*pstree-Befehl* 382  
*pwd-Befehl* 372  
*RedHat-Distribution* 351  
*Reguläre Ausdrücke suchen* 375  
*rm-Befehl* 371  
*rmdir-Befehl* 373  
*root, Benutzer* 292, 355  
*route-Befehl* 408

### Linux (Forts.)

*rpm, Paketmanager* 386  
*Runlevel* 385  
*Samba-Server* 412  
*SaX, openSUSE-X-Konfigurationsprogramm* 402  
*Shell* 288, 353  
*Shell-Ausgabeumleitung* 367  
*Shell-Eingabeumleitung* 367  
*Shell-Eingabevollständigung* 361  
*Shell-History* 361  
*Shell-Skripte* 389  
*shutdown-Befehl* 382  
*Software installieren* 386  
*Stand-Alone Shell* 358  
*Standardrouter einrichten* 408  
*startx-Befehl* 402  
*su-Befehl* 362  
*SUSE-Distribution* 351  
*Swap-Partition* 295  
*Symbolic Links* 299  
*Syslog* 382–383  
*System herunterfahren* 382  
*System V Init* 384  
*Systemprogramme* 280, 369  
*tail-Befehl* 375  
*tar-Befehl* 387, 833  
*tar-Dateien* 387  
*Textbefehle* 374  
*Textdatei anzeigen* 374  
*Textdateien vergleichen* 377  
*Texteditoren* 395  
*Ubuntu-Distribution* 352  
*Umgebung* 358  
*Umgebungsvariablen setzen* 360  
*umount-Befehl* 378  
*unalias-Befehl* 393  
*unzip-Befehl* 836  
*useradd-Befehl* 380  
*userdel-Befehl* 380  
*User-ID* 291, 355  
*Verwaltungsbefehle* 377  
*Verzeichnis anlegen* 373  
*Verzeichnis löschen* 373  
*Verzeichnis wechseln* 372  
*Verzeichnisbaum* 297  
*Verzeichnisbefehle* 370  
*Verzeichnisinhalt anzeigen* 372  
*vi, Editor* 395  
*virtuelles Terminal* 356

- Linux (Forts.)
    - wc-Befehl* 377
    - Window-Manager* 289
    - Wörter zählen* 377
    - X Window* 289, 401
    - Xconfigurator, X-Konfigurationsprogramm* 402
    - zip-Befehl* 836
    - Zugriffsrechte* 300
  - Lisp, Programmiersprache 51, 401
  - Listen
    - C 524
    - Java 526
    - Perl 475
    - qw//, Perl* 478
  - listen(), Perl-Funktion 550
  - Listener, AWT-Ereignisbehandlung 595
  - Literal 1038
  - Literale 438
    - Fließkommazahlen* 438
    - Integer* 438
    - Perl* 478
    - Strings* 438
    - Zeichen* 438
  - Little-Endian-Architektur 1033, 1038
  - Little-Endian-Plattformen 825
  - LOAD DATA INFILE, MySQL-Anweisung 692
  - Local Area Network 182
  - LocalTalk 267
  - localtime(), C-Funktion 455
  - Lochkarte 38
  - Lochkarten 274
  - Logarithmische Komplexität 95
  - Logdateien
    - MySQL 693
  - logger, UNIX-Befehl 384
  - Logical Link Control (LLC) 174
  - Logik
    - AND 61
    - Aussageformen* 59
    - Aussagen* 58
    - Aussagenlogik* 57
    - Definition* 57
    - Disjunktion* 62
    - Formale Logik* 57
    - Gleichung* 58
    - Konjunktion* 61
  - Logik (Forts.)
    - Lügende Kreter* 93
    - Mathematische Aussagen* 58
    - Operatoren, C* 439
    - OR 61
    - Prädikatenlogik* 51, 57
    - Schlussfolgerung* 60
    - Term* 58
    - Umkehrschluss* 60
    - Ungleichung* 58
    - Verknüpfungen* 60
    - Wahre und falsche Aussagen* 58
    - Widerspruch* 93
    - XOR 64
  - Logikschaltkreis 82
  - Logikschaltungen
    - Addierwerk* 87
    - AND/OR-Aufbau durch Transistoren* 85
    - Flip-Flop* 88
    - Gattersymbole* 85
    - Halbaddierer* 86
    - mit einfachen Mitteln nachbauen* 82
    - Multiplexer* 86
    - NAND-Schaltung* 84
    - NOR-Schaltung* 84
    - NOT-Schaltung* 84
    - RS-Flip-Flop* 88
    - Speicherzelle* 89
    - Volladdierer* 87
    - XOR-Schaltung* 86
  - Login 1038
  - Login (UNIX-Anmeldung) 353
  - loginShell, LDAP-Attribut 756
  - Logische Laufwerke → Logische Partitionen
  - Logische Operatoren
    - C 439
    - Short-Circuit-Logik* 480
    - Vergleich mit Bit-Operatoren* 62
  - Logische Partitionen 144
  - Logische Programmiersprachen 51
  - Logische Schlussfolgerung 60
  - Logische Verknüpfungen 60
    - in Programmiersprachen* 66
  - Logisches Nicht, Operator 439
  - Logisches Oder, Operator 439
  - Logisches Und, Operator 439
  - Logo, Programmiersprache 51
  - Logos 57
  - Lokale Variablen, C 437
  - Lokales Netz, Entwicklung 41
  - Lokalisierung
    - Mac OS X 422
  - long, C-Datentyp 437
  - LONGBLOB, SQL-Datentyp 675
  - LONGTEXT, SQL-Datentyp 675
  - Loopback, IP-Protokoll 217
  - Löschabfrage 672
  - Löschen
    - Dateien, UNIX* 371
    - Dateien, Windows-Konsole* 322
    - Verzeichnis, Windows* 322
  - Lösung einer Gleichung oder Ungleichung 59
  - Lösungsmenge 59
  - Lovelace, Ada 36
  - ls, UNIX-Befehl 372
  - LS-120-Laufwerk 147
  - lt, Perl-Operator 479
  - LVALUE 440
    - Liste als* 475
- ## M
- 
- m, n-Relation, RDBMS 658
  - m//, Perl-Operator 530, 534
  - Mac 43
  - Mac OS 9, MacPerl 471
  - Mac OS X 277, 415
    - Alias* 421
    - als Server einrichten* 425
    - Anwendungsmenü* 419
    - APIs* 416
    - Apple-Menü* 418
    - Aqua* 277, 417
    - Aqua, praktische Anwendung* 417
    - Aqua-Fenster* 418
    - Benutzerverwaltung* 422
    - Carbon* 417
    - Classic-Umgebung* 417
    - Cocoa* 416
    - Darwin* 416
    - Data Fork* 422
    - Dateien kopieren* 421
    - Dateien umbenennen* 421
    - Dateien verschieben* 421
    - Dateisysteme* 422
    - Desktop* 418

## Index

- Mac OS X (Forts.)
  - Dock* 418–419
  - DSL einrichten* 424
  - File Sharing* 425
  - Finder* 418, 420
  - Geschichte* 415
  - HFS+-Dateisystem* 422
  - HFS-Dateisystem* 422
  - hqx-Format* 423
  - IP-Adresse zuweisen* 424
  - Java* 417
  - Konfiguration* 421
  - Kontextmenü* 421
  - Landeseinstellungen* 422
  - Menüleiste* 417
  - Monitore, Systemeinstellungen* 422
  - Netzwerkkonfiguration* 424
  - OpenGL, 3D-Grafikbibliothek* 416
  - Ordneransichten* 420
  - Papierkorb* 420
  - Quartz, Grafikbibliothek* 416
  - QuickTime* 416
  - Resource Fork* 423
  - Server nutzen* 426
  - Startvolume einstellen* 421
  - Systemeinstellungen* 421
  - Terminal starten* 417
  - UNIX-Komponenten* 417
  - Windows File Sharing* 425
- Mac OS X
  - Architektur* 416
- Mac9, Perl-Unterstützung 471
- MAC-Adresse 192, 1038
- mach-Mikrokern 283
- Macintosh 43, 279
  - BIOS* 129
  - Firmware-Update* 130
  - Leonardo, ISDN-Leonardo, Mac-  
ISDN-DFÜ* 208
  - Mac OS X* 415
  - Parameter-RAM* 129
  - Serversysteme* 425
- MacPerl 471
- Magnetbänder 141
- Magnetbandspeicher 40
- Magnetische Datenträger 140
  - Magnetbänder* 141
  - Magnetscheiben* 140
- Magneto-optische Datenträger 140
  - mail, LDAP-Attribut* 756
  - mail, UNIX-Befehl* 394
  - Mail-Server* 187
  - mailto, HTML-Hyperlink* 855
  - main(), AWT-Methode* 584
  - main(), C-Funktion* 50, 433, 447
    - Kommandozeilenparameter* 449
  - main(), Java-Methode* 460
  - Mainboard* 111
    - Chipsatz* 112
    - Kartensteckplätze* 134
    - Onboard-Peripherie* 112
  - Mainframe* 39
  - make, UNIX-Befehl* 387
  - Makefile* 387
  - Makroviren* 1004
  - malloc(), C-Funktion* 526
  - MAN* 1038
  - MAN, Stadtnetz* 182
  - man, UNIX-Befehl* 288
  - Mandriva, Linux-Distribution* 352
  - Manetscheiben* 140
  - mangle, iptables-Tabelle* 1020
  - Man-in-the-Middle-Angriff* 1016
  - Manuelle Datenverarbeitung* 25
  - Marconi, Guglielmo* 198
  - margin, CSS-Angabe* 887
  - Mark II, Röhrenrechner* 37
  - Mark I, Röhrenrechner* 37
  - Maschinenbefehle* 121
  - Maschinensprache* 45
  - Massenspeicher* 139
    - CD-ROM* 147
    - Diskettenlaufwerk* 146
    - DVD* 151
    - Festplatten* 141
    - Jaz-Laufwerk* 147
    - LS-120* 147
    - magnetische* 140
    - magneto-optische* 140
    - optische* 140
    - Übersicht* 140
    - Wechseldatenträger* 146
    - Zip-Laufwerk* 147
  - Master Boot Record* 142
  - Master-Nameserver* 249
  - Matching, Reguläre Ausdrücke* 534
- Math, Java-Klasse 463
- Mathematische Aussagen 58
- Mathematische Funktion, Undefinedheitsstelle 92
- Mathematische Variablen 59
- Mathematischer Term 58
- Matrixdrucker 159
- Maus 154
- MAX, SQL-Funktion 679
- Maximum Transmission Unit (MTU) 226
- MBR → *Master Boot Record*
- md → *mkdir*, Windows-Befehl
- MDI, Windows-Anwendungen 313
- Mechanische Datenverarbeitung 25
- Media Access Control (MAC) 174
- Median, QuickSort 521
- Mediator, Entwurfsmuster 643
- MediaWiki, Wikipedia-Software 936
- Medieninformatik 32
- MEDIUMBLOB, SQL-Datentyp 675
- MEDIUMINT, SQL-Datentyp 674
- MEDIUMTEXT, SQL-Datentyp 675
- Medizinische Informatik 32
- Megabyte 79
- Megapixel (Digitalkamera) 155
- Mehrfachvererbung 468
- mem.h, C-Bibliothek 526
- Memento, Entwurfsmuster 643
- Memory Management Unit → *MMU*
- Mena, Federico 403
- Mengen, diskrete 52
- Mengenoperationen 67
  - Differenzmenge* 70
  - Echte Teilmenge* 68
  - Element* 67
  - Leere Menge* 70
  - Obermenge* 68
  - Schnittmenge* 70
  - Teilmenge* 67
  - Vereinigungsmenge* 70
  - Verknüpfungen* 70
- Menu, AWT-Klasse 593
- MenuBar, AWT-Klasse 593
- MenuItem, AWT-Klasse 593

- Message-Digest 1024
- Message-Passing 582
- METAFONT, TeX-Zeichensätze 822
- Methoden 458
  - Callbacks 543
  - überladen 465
- Metropolitan Area Network 182
- Microsoft 40, 278
  - ASP.NET 934
  - BizTalk Server 349
  - Exchange Server 349
  - Management Console (MMC) 338
  - MS-DOS 278
  - SQL Server 349
  - SQL-Server 664
  - Systems Management Server 349
  - Visual Studio 608
  - Windows 279, 305
  - Windows 2000 280
  - Windows 95 279
  - Windows 98 280
  - Windows Me 280
  - Windows NT 279
  - Windows XP 280
  - Windows, Versionsübersicht 306
- MIDI 162, 1038
  - FM-Synthese 162
  - Wavetable-Synthese 163
- Migration (Ruby on Rails) 924
- Mikrokern 283
  - mach 283, 416
- Mikroprozessor 40, 111
- Mikroprozessor → Prozessor
- MilNet 169
- MIME 1038
  - E-Mail-Nachricht 257
- MIME-Multipart-Nachrichten 258
- MIME-Nachrichtenheader 257
- MIME-Type
  - HTML-Plug-in-Formate 875
  - XML-Dokumente 771
- MIME-Types 257
  - im HTTP-Header 264
- MIN, SQL-Funktion 679
- Minicomputer 40
- Minix, Lehrbetriebssystem 280
- MIPS (CPU-Geschwindigkeit) 117
- MIPS, Prozessor 119
- Mirroring (RAID) 145
- MITS 40
- Mixed-Mode-CD 148
- mkdir, UNIX-Befehl 373
- mkfs, UNIX-Befehl 322
- mkfs, UNIX-Befehl 379
- MMC (Microsoft Management Console) 338
- MMU 115, 294, 1038
  - Seitentabelle 295
- MMX (CPU-Befehlserweiterung) 118
- Mnemonics 46, 1032
- Model (Ruby on Rails) 923
- Model, View, Controller → MVC-Entwurfsmuster
- Modem 205
  - Anschließen 206
  - AT-Befehlssatz 206
  - Handshake 207
  - Hayes-Befehlssatz 206
  - Pulswahl 206
  - Tonwahl 206
- modprobe, Linux-Befehl 387
- Modularisierung von Programmen 48
- Modularisierung, Programme 48, 448
- Modularität, von UNIX 276
- Modulo, Operator 439
- Molenaar, Bram 395
- Mongrel, Webserver 933
- Monitor 157
  - Bildwiederholrate 158
  - Konfiguration, Mac OS X 422
  - LCD 158
  - mehrere verwenden 157
  - Röhrenmonitor 157
  - Zeilenfrequenz 158
- Monolithischer Kernel 283
- more, UNIX-Befehl 375
- MosTek 43
- MosTek 6502, Prozessor 116
- Motherboard → Mainboard
- Motorola 68000, Prozessorfamilie 116
- mount, UNIX-Befehl 377
- Mounten 1038
  - Dateisysteme 299, 377
  - NFS-Freigaben 410
- MouseListener 595
- MouseMotionListener 596
- MovableType, Blog-Tool 936
- MOV-Befehl
  - beim virtuellen Prozessor 101
  - x86-Assembler 121
- move, Windows-Befehl 322
- MP3, Audio-Dateiformat 831, 1038
- MP4, Audio-Dateiformat 832
- MPEG 1038
  - Video-Dateiformat 833
- MS-DOS 278
  - Kommandozeile 319
- MS-DOS-Anwendungen, unter Win32 307
- MTU 226, 1038
- Multi Document Interface → MDI, Windows-Anwendungen
- Multicasting, IP-Protokoll 216
- MULTICS 276
- Multimedia-Datenbank 655
- Multipart-E-Mails 258
- Multiplexer (Schaltung) 86
- Multiplikation, Operator 439
- Multiplikator
  - bei Rambus-RAM 123
  - der Taktfrequenz 116
- Multisession-CD 149
- Multitasking 285, 289
  - kooperatives 285
  - präemptives 285
  - Unterstützung durch CPU 114
- mv, UNIX-Befehl 371
- MVC-Entwurfsmuster 639, 921
- MX-Record, BIND-Nameserver 753
- my, Perl-Variablendeklaration 474
- my.cnf, MySQL-Konfigurationsdatei 692
- MyISAM, MySQL-Tabellentyp 682
- MyISAM-Tabellen (MySQL) 663
- MySQL 667
  - Authentifizierung 684
  - Backups 689
  - Backups automatisieren 691
  - Benutzerrechte 687
  - Benutzerverwaltung 684

## Index

- MySQL (Forts.)  
  *CREATE USER, Anweisung* 685  
  *Datentypen* 674  
  *DROP USER, Anweisung* 689  
  *Export in Textdateien* 692  
  *FLUSH PRIVILEGES, Anweisung* 689  
  *FLUSH TABLES, Anweisung* 691  
  *GRANT, Anweisung* 686  
  *Import aus Textdateien* 692  
  *Import/Export* 689  
  *InnoDB-Tabellen* 663  
  *Installation (UNIX)* 667  
  *Installation (Windows)* 669  
  *JDBC-Anbindung* 697  
  *Konfiguration (Windows)* 669  
  *Konfigurationsdateien* 692  
  *LOAD DATA INFILE, Anweisung* 692  
  *Logdateien* 693  
  **lesen** 694  
  *my.cnf* 692  
  *MyISAM-Tabellen* 663  
  *MySQL Administrator* 668  
  *MySQL Query Browser* 668  
  *mysqldadmin, Hilfsprogramm* 683  
  *mysqlbinlog, Hilfsprogramm* 694  
  *mysqldump, Hilfsprogramm* 689  
  *mysqli, PHP-Schnittstelle* 919  
  *mysqli-Kommandozeilen-Client* 671  
  *PHP-Zugriff auf* 914  
  *Replikation* 695  
  *REVOKE, Anweisung* 688  
  *SET PASSWORD, Anweisung* 686  
  *Sicherheitshinweise (UNIX)* 668  
  *Sicherheitshinweise (Windows)* 670  
  *Tabellentypen* 682  
  *Testdatenbank* 914  
  *Transaktionen* 663, 682  
MySQL Connector/J, JDBC-Schnittstelle 697  
MySQL Grant Tables 668  
mysqldadmin, Hilfsprogramm 683  
mysqlbinlog, Hilfsprogramm 694  
mysqli-Client  
  *nicht-interaktiver Betrieb* 691  
mysqldump, Hilfsprogramm 689  
mysqli-Schnittstelle 919
- ## N
- Nachrichtenübermittlung 582  
Nadeldrucker 159  
Namensräume, XML 792  
Nameserver  
  *BIND* 747  
  *Master* 249  
  *Slave* 249  
Nassi-Shneiderman-Struktogramm 91  
NAT 1038  
NAT, IP-Protokoll 238  
  *IP-Masquerading* 239  
nat, iptables-Tabelle 1020  
Natürliche Zahlen 68  
navigator, JavaScript-Objekt 964  
ne, Perl-Operator 479  
Nebenläufigkeit 537  
Nessus 1018  
netatalk  
  *afpd* 414  
  *atalkd* 414  
  *Konfiguration* 414  
  *papd* 414  
  *Verzeichnisse freigeben* 414  
netatalk, Serverdienst 414  
NetBEUI 1038  
NetBEUI, Netzwerkprotokoll 270  
NetBIOS, Windows-Netzwerk 270  
netfilter, Linux-Kernel-Firewall 1019  
Netscape-Palette 885  
netstat, TCP/IP-Dienstprogramm 344  
  *Routing-Tabellen anzeigen* 233  
Network Address Translation (NAT) 238  
Network File System → NFS  
Netzplan 622  
  *kritischer Pfad* 623  
Netzwerkclient, Definition 185  
Netzwerke 165  
  *Anwendungsgebiete* 166  
  *ARCNet* 204  
  *ATM* 203  
  *Client-Server* 184  
  *Denial of Service-Angriff* 344  
  *drahtlose* 198  
  *Drucker freigeben, Windows* 347  
  *Ethernet* 192  
  *FDDI* 203  
  *Funktionsebenen* 172  
  *GAN (Global Area Network)* 182  
  *Geschichte* 167  
  *Hardware* 165, 191  
  *IEEE 802-Standard* 191  
  *Klassifizierung* 181  
  *Konfiguration, Linux* 407  
  *Konfiguration, Mac OS X* 424  
  *Konfiguration, Windows* 341  
  *LAN (Local Area Network)* 182  
  *Logical Link Control* 174  
  *lokales, Entwicklung* 41  
  *MAN (Metropolitan Area Network)* 182  
  *Media Access Control* 174  
  *Netzwerkprogrammierung* 545  
  *OSI-Referenzmodell* 172  
  *OSI-Schicht* 174  
  *Peer-to-Peer* 185  
  *Protokolle* 166, 212  
  *Reichweite* 182  
  *Schichtenmodelle* 172  
  *TCP/IP-Protokolle* 212  
  *Token Ring* 197  
  *Topologie* 183  
  *Verkabelung* 165  
  *WAN (Wide Area Network)* 182  
  *Zentralisierungsgrad* 184  
  *Zugang per Modem* 205  
  *Zugang über DSL* 209  
  *Zugang über ISDN* 207  
Netzwerkhardware 191  
  *Ethernet* 194  
  *ISDN-Adapter* 208  
  *Modem* 205  
Netzwerkprogrammierung 545  
  *Berkeley Socket API* 545  
  *IO::Socket* 551  
  *sockaddr\_in* 547  
  *Sockets* 545  
Netzwerkprotokolle 166  
  *AppleTalk* 266  
  *IPX/SPX* 269  
  *NetBEUI* 270

- Netzwerkserver, Definition 185  
 Netzwerktopologie 183  
   *Baum* 184  
   *Bus* 183  
   *logische* 184  
   *physikalische* 184  
   *Ring* 183  
   *Stern* 183  
 Netzzugang, DDN-Modell-Schicht 176  
 Netzzugangsverfahren  
   CSMA/CA 201  
   CSMA/CD 193  
   IEEE 802-Standards 191  
   Token-Passing 197  
 Neumann, John von 37, 109  
 Neuronale Netze 45  
 Neustart, Betriebssystem 382  
 Neustart, Windows 316  
 new, Java-Operator 462  
 Newsfeeds 936  
 Newsgroups 261  
   *Hierarchie* 262  
   NNTP-Protokoll 261  
 NeXT 415  
 NFS 410, 1038  
   */etc/exports, Konfigurationsdatei* 410  
   *auf andere Server zugreifen* 410  
   *Starten* 410  
   *Verzeichnisse freigeben* 410  
 NNTP 261, 1038  
   *Header* 261  
 Normalformen, RDBMS 661  
 Normalisierung 1039  
 Normalisierung, RDBMS 661  
 NOR-Schaltung 84  
 NOT-Schaltung 84  
 Novell NetWare 172  
 NSFNet 169  
 nslookup, TCP/IP-Dienstprogramm 345  
 NS-Record (DNS) 752  
 NTBA (ISDN-Endgerät) 208  
 NTFS, Dateisystem 310  
 Nucleus, Blog-Tool und CMS 936  
 Null, im Stellenwertsystem 35  
 NULL, leerer C-Zeiger 455  
 NULL, SQL-Feldoption 676  
 Null-terminierte Strings 451  
 Nyquist-Theorem 1039  
 Nyquist-Theorem → Shannon-Theorem
- O**
- 
- o, LDAP-Attribut 756  
 Obermenge 68  
 Object Database Management Group → ODMG  
 Object Definition Language 665  
 Object Management Group 633  
 Object Management Group (OMG) 633  
 Object Query Language → OQL  
 Object, Java-Klasse 463  
 objectClass (LDAP) 754  
 objectClass, LDAP-Attribut 755  
 Objective-C, Programmiersprache 416  
 Objekte 50, 458  
 Objektorientierte Analyse 624  
 Objektorientierte Datenbank 655, 664  
   *Abfragen* 667  
   ODL 665  
   OQL 667  
 Objektorientierung 50, 458, 1039  
   *Eigenschaften* 458  
   *Elternklasse* 466  
   *Ereignisse* 582  
   *im Software-Engineering* 617  
   *Instanzen* 458  
   *Instanzen erzeugen, Java* 462  
   *Interfaces, Java* 468  
   *IS A-Beziehungen* 636  
   *Java* 463  
   *Kapselung* 50, 458  
   *Kindklasse* 466  
   *Klassen* 50, 458  
   *Konstruktor* 465  
   *Methoden* 458  
   *Nachrichten* 582  
   *Objekte* 458  
   *Perl* 551  
   *PHP* 904  
   *Überladen* 465  
   *Vererbung* 50, 466  
 Observer, Entwurfsmuster 643  
 ODBC 1039  
 ODBC, Microsoft-Datenbank-schnittstelle 697  
 Oder-Schaltung → OR-Schaltung  
 Oder-Verknüpfung → OR-Verknüpfung  
 ODL 665  
 ODMG 665  
 OFDM, WLAN-Technik 200  
 OGG Vorbis, Audio-Dateiformat 832  
 Oktalsystem 74  
   *in Dual umrechnen* 77  
 Oktalzahlen, C 438  
 OMG → Object Management Group  
 Onboard-Hardware 112  
 Online-Dienste 170  
 Online-Durchsuchung 1008  
 O-Notation 1039  
 O-Notation (Komplexität) 94  
 onreadystatechange, Ajax-Eigenschaft 984  
 OODBMS 665  
 OOP → Objektorientierung  
 Open Database Connectivity → ODBC, Microsoft-Datenbank  
 open(), Ajax-Methode 983  
 open, Perl-Anweisung 482  
 opendir, Perl-Anweisung 483  
 OpenGL, 3D-Grafikbibliothek 416  
 OpenSSH 251  
 openSUSE, Linux-Distribution 351  
 Operatoren 440  
   - 439  
   ! 439  
   != 440  
   !~ 534  
   \$, *RegExp* 533  
   % 439  
   &&, C 439  
   &, C 439  
   \* 439  
   \*, *RegExp* 531  
   + 439  
   +, *Java-String-Verkettung* 461–462  
   +, *RegExp* 531  
   -, C 452  
   ., *Perl* 479

## Index

- Operatoren (Forts.)  
/ 439  
= 440  
== 440  
=~ 534  
=~, Perl 483  
? 441  
?, RegExp 531  
^, C 440  
^, RegExp 531, 533  
` 539  
| 439  
|, RegExp 533  
|| 439  
~ 440  
Addition 439  
arithmetische 439  
Backticks 539  
Bit-Komplement 440  
Bit-Verschiebung, links 440  
Bit-Verschiebung, rechts 440  
bitweise 439  
Bitweise Exklusiv Oder 440  
Bitweise Oder 439  
Bitweise Und 439  
C 439  
Division 439  
eq, Perl 479  
Fallunterscheidung 441  
ge, Perl 479  
Gleichheit 440  
größer als 440  
größer oder gleich 440  
gt, Perl 479  
in der PowerShell 325  
kleiner als 440  
kleiner oder gleich 440  
le, Perl 479  
logische 439  
Logisches Nicht 439  
Logisches Oder 439  
Logisches Und 439  
lt, Perl 479  
m// 534  
Mengen 67  
Modulo 439  
Multiplikation 439  
ne, Perl 479  
Perl 477  
PHP 896  
Post-Dekrement 441
- Operatoren (Forts.)  
Post-Inkrement 441  
Prä-Dekrement 441  
Prä-Inkrement 441  
Rangfolge 442  
Rangfolge durch Klammern  
ändern 442  
s// 534  
Stringvergleiche, Perl 479  
String-Verkettung, Java 461–462  
Subtraktion 439  
tr// 535  
Ungleichheit 440  
Vergleiche 65, 440  
Wertzuweisung 440  
opt, UNIX-Verzeichnis 298  
Optionen, bei Systemprogrammen 359  
Optische Datenträger 140  
OQL 667  
Oracle, Datenbank 664  
Orange Book 1039  
Orange Book (CD-R, CD-RW) 148  
ORDER BY, SQL-Klausel 679  
Ordnungsdaten 654  
org.w3c.dom.\*, Java-Package 808  
org.xml.sax.\*, Java-Package 800  
Organisationseinheiten (LDAP) 754  
OR-Schaltung 85  
Aufbau mit Transistoren 85  
mit einfachen Mitteln nachbauen 83  
OR-Verknüpfung 61  
OS/2, Betriebssystem 279  
OSI-Referenzmodell 172, 1039  
Anwendungsschicht 175  
Bitübertragungsschicht 173  
Darstellungsschicht 175  
Kommunikationssteuerungsschicht 175  
Netzwerkschicht 174  
Präsentationsschicht 175  
Sicherheitsschicht 174  
Sitzungsschicht 175  
Transportschicht 174  
Vergleich mit der Praxis 175  
Vergleich zum Internet-Schichtenmodell 176  
OSPF 1039
- OSPF, Routing-Protokoll 236  
ou-Knoten (LDAP) 754  
OUTPUT, iptables-Chain 1020  
Outsourcing 171  
Overclocking → Übertakten
- 
- ## P
- Paare, Extreme Programming 632  
packet switching 166  
padding, CSS-Angabe 887  
Page Fault (Speicher) 295  
Page File (Auslagerungsdatei) 295  
Pager (seitenweise anzeigen) 375  
Paging (Speicher) 115, 295  
paint(), AWT-Methode 583  
Paketfilter  
iptables 1019  
netfilter 1019  
Paketmanager, Linux 386  
Paketvermittlung 166  
Panel, AWT-Klasse 583, 593  
papaya CMS 935  
papd, Netatalk-Dienst 414  
Papierkorb, Mac OS X 420  
Parallele Datenübertragung 130  
Parallelport 138  
Parameter  
bei Systemprogrammen 359  
von Perl-Subroutinen 481  
Parameter-RAM (Mac) 129  
Parameter-RAM, löschen 129  
PARC, Xerox-Forschungszentrum 278  
Parent-Prozess 291, 1039  
Parity Bit → Prüfbit  
parseFloat(), JavaScript-Methode 944  
parseInt(), Java-Methode 518  
parseInt(), JavaScript-Methode 944  
Partitionen  
erweiterte 144  
logische 144  
primäre 142  
Partitionierung 1039  
praktische Durchführung 144  
Partitionierung (Festplatten) 142  
Partitionstabelle 142  
Partitionstypen 143  
Pascal, Blaise 35

- Pascal, Programmiersprache 48  
 passwd, UNIX-Befehl 381  
 Passwort, ändern, UNIX 381  
 Passwörter 1019  
   *Brute-Force-Attacke* 356  
   *crack (Knackprogramm)* 355  
   *Erzeugungstipps* 356  
 patch, UNIX-Befehl 377  
 Patches, Sicherheit 1006  
 path, Umgebungsvariable  
   Windows 321  
 PATH, Umgebungsvariable (UNIX)  
   360  
 pause(), UNIX-Systemaufruf 291  
 PC 42  
   *Aufbau* 109  
   *Geschichte* 278  
   *Zentraleinheit* 111  
 PC-Card → PCMCIA-Anschluss  
 PCDATA, Text in XML 777  
 PCI 134, 1039  
 PCMCIA-Anschluss 135  
 PC-Netzwerke, Entwicklung 172  
 PDA (Personal Digital Assistant)  
   44  
 PDF 1039  
 PDP, Kleincomputerserie von DEC  
   276  
 PDP, Kleinrechnerserie von DEC  
   39, 49  
 Pentium, Prozessorfamilie 113,  
   118  
 people, LDAP-Organisationsein-  
   heit 754  
 Peripherie  
   *Ausgabegeräte* 156  
   *Digitalkamera* 155  
   *Drucker* 159  
   *Eingabegeräte* 152  
   *Einsteckkarten* 134  
   *Grafikkarte* 156  
   *Maus* 154  
   *Monitor* 157  
   *Scanner* 154  
   *Tastatur* 153  
   *Übersicht* 139  
 Perl 1039  
 Perl, Programmiersprache 471  
   !~-Operator 534  
   \$ 473–474  
   \$/-Variable 814  
 Perl, Programmiersprache (Forts.)  
   \$\_ 476  
   \$-Operator, RegExp 533  
   % 474  
   &&-Operator 480  
   \*-Operator, RegExp 531  
   +-Operator, RegExp 531  
   .-Operator 479  
   //-Operator 534  
   =~ 483  
   =~-Operator 534  
   ?-Operator 531  
   @ 474  
   @\_ 481  
   ^Operator, RegExp 531, 533  
   || zur Fallunterscheidung 480  
   |-Operator 480  
   |-Operator, RegExp 533  
   accept()-Funktion 551  
   ActivePerl-Distribution 471  
   alternative Zeichen in RegExp  
     531  
   Anführungszeichen 478  
   Array-Elementanzahl 519  
   Arrays 474  
   Arrays zusammenfassen 476  
   Ausdrücke 477  
   Ausgabe 473  
   Autoflush-Modus 549  
   Backticks 539  
   bind()-Funktion 550  
   chomp()-Funktion 473  
   close 482  
   closedir 483  
   connect()-Funktion 549  
   Dateien bearbeiten 482  
   Dateihandles 482  
   defined 481  
   Dereferenzierung 477  
   die 480  
   Eingabe 473  
   elsif 480  
   eq-Operator 479  
   Ersetzen durch RegExp 534  
   foreach-Schleife 475  
   ge-Operator 479  
   gethostbyname()-Funktion 547  
   getprotobyname()-Funktion 547  
   getservbyname()-Funktion 548  
   gt-Operator 479  
   Hashes 474  
 Perl, Programmiersprache (Forts.)  
   HIER-Dokumente 479  
   if, nachgestelltes 480  
   IO 549, 551  
   join 476  
   keys 477  
   Kommentare 478  
   le-Operator 479  
   Listen 475  
   listen()-Funktion 550  
   Listenfunktionen 475  
   Literale 478  
   lt-Operator 479  
   m/-Operator 530  
   MacPerl 471  
   my 474  
   ne-Operator 479  
   Objektorientierung 551  
   open 482  
   opendir 483  
   -Operator 473  
   Operatoren 477  
   Parameter von Subroutinen 481  
   pipe() 540  
   Pipes verwenden 540  
   pop 475  
   print-Anweisung 473  
   Programmabbruch 480  
   Programme starten 472  
   push 475  
   q// 478  
   qq// 478  
   Queue 524  
   qw// 478  
   readdir 483  
   Referenzen 477  
   Reguläre Ausdrücke 483, 529  
   s///-Operator 484, 530, 534  
   Shebang 472  
   shift 475  
   Skalare 473–474  
   Socket-Modul 546  
   split 476  
   Stack 524  
   strenge Wertüberprüfung 474  
   strict 474  
   Strings zerlegen 476  
   String-Vergleiche 479  
   String-Verkettung 479  
   sub 481  
   Subroutinen 481

## Index

- Perl, Programmiersprache (Forts.)
  - tr///-Operator* 535
  - unless* 481
  - unshift* 475
  - use-Direktive* 474
  - Variablen 473
  - Variablenexistenz prüfen 481
  - Variablensubstitution 473
  - Verzeichnisse verarbeiten 483
  - w, Modifikator* 472
  - Warnungen aktivieren 472
  - Zeichengruppen in *RegExp* 531
  - Zeilenumbbruch 814
  - Zeilenumbüche entfernen 473
- Perl-Anweisung 480
- Permutationen 94
- person, LDAP-Objektklasse 754
- Personal Computer → PC
- Personal Digital Assistant (PDA) 44
- Petabyte 79
- Pfad
  - absoluter* 298, 301
  - in HTML-Hyperlinks* 852
  - relativer* 298, 301
  - unter UNIX* 298
  - unter Windows* 301
- PGP 1025
- PGP 1039
- Phishing 1011
- Phishing 1039
- PHP 893, 1039
  - \$, Variablenbeginn* 895
  - Ajax-Antwort durch* 986
  - Anführungszeichen* 897
  - Arrays 896
  - Arrays als Hashes* 897
  - Arrays, mehrdimensionale* 899
  - Bezeichner 895
  - Call by Reference* 902
  - Cookies 912
  - Datei-Uploads* 910
  - Datenbankzugriff* 914
  - Datentyp testen* 904
  - Dokumentaufbau* 894
  - each()-Funktion* 897
  - explode()-Funktion* 898
  - extends, Schlüsselwort* 907
  - function, Schlüsselwort* 900
  - Funktionen 900
  - Funktionsparameter* 901
- PHP (Forts.)
  - Funktionswertrückgabe* 902
  - global* 901
  - Globale Variablen* 901
  - implode()-Funktion* 898
  - include()-Funktion* 908
  - Include-Dateien* 908
  - is\_array()-Funktion* 904
  - is\_float()-Funktion* 904
  - is\_int()-Funktion* 904
  - is\_null()-Funktion* 904
  - is\_string()-Funktion* 904
  - json\_encode(), Funktion* 997
  - Kommentare 900
  - Konstruktor 906
  - Mehrdimensionale Arrays* 899
  - mysqli-Schnittstelle* 919
  - Objektorientierung 904
  - Operatoren 896
  - php.ini, Konfigurationsdatei* 738
  - preg\_match()-Funktion* 899
  - preg\_replace()-Funktion* 899
  - Referenzen 902
  - Reguläre Ausdrücke* 899
  - return-Anweisung* 902
  - Sessions 911
  - sizeof()-Funktion* 897
  - Sprachgrundlagen* 895
  - Strings zerlegen* 898
  - Variablen 895
  - Vererbung 907
  - Webspezifische Funktionen* 909
- php.ini, Konfigurationsdatei* 738
- PICT, Bilddateiformat 831
- PID (Prozess-ID) 290
- ping, TCP/IP-Dienstprogramm 343
  - Ergebnisse auswerten* 343
- Pipe 1039
- pipe()*, Perl-Funktion 540
- Pipeline (CPU-Warteschlange) 118
- Pipes
  - Anwendung* 539
  - in Programmiersprachen* 537
  - in UNIX-Shells* 368
  - Perl 540
  - Windows 321
  - zur Inter-Prozess-Kommunikation* 292
- Pixelgrafik 53
- PK-Zip-Dateien → ZIP-Dateien
- Planung, Software-Engineering 620
- Platzhalter
  - in UNIX-Dateinamen* 370
  - Windows-Dateinamen* 321
- Plug & Play 134, 1039
- Pluralisierung (Rails) 923
- PNG, Bilddateiformat 830
- Point 1040
- Polymorphe Viren 1004
- Polynomielle Komplexität 95
- pop, Perl-Anweisung 475
- POP3 259, 1039
  - Befehle* 260
  - Sitzung* 259
- Ports, TCP 243
- Ports, UDP 245
- position, CSS-Angabe 889
- POSIX 1039
- posixAccount, LDAP-Objektklasse 754
- POSIX-Standard 277
- POST 125
- POST (BIOS-Selbsttest) 125
- POST, HTTP-Befehl 264
- POST, HTTP-Methode
  - zum HTML-Formularversand* 869
- Post-Dekrement 441
- PostgreSQL 664
- Post-Inkrement 441
- POSTROUTING, iptables-chain 1020
- PostScript 161, 823, 1039
  - EPS* 824
  - PPD* 823
- Power Management 127
- Power-On Self Test → POST
- PowerPC, Prozessor 116
- PowerShell 323
  - Benutzereingaben* 329
  - CmdLets* 324
  - Datentypen* 329
  - Fallentscheidungen* 328
  - Get-Alias, CmdLet* 324
  - Get-ChildItem, CmdLet* 324
  - Get-Command, CmdLet* 324
  - Kontrollstrukturen* 328
  - Operatoren* 325

- PowerShell (Forts.)  
*Read-Host, CmdLet* 329  
*Schleifen* 330  
*Skriptdateien* 332  
*Variablen* 327  
*Write-Host, CmdLet* 328
- PPD, Druckerbeschreibungsdatei 823
- PPP 1040
- PPP-Protokoll, DFÜ 205
- Prä-Dekrement 441
- Prädikatenlogik 51, 57, 1040
- Präemptives Multitasking 285
- Prä-Inkrement 441
- Praktische Informatik 26
- Präprozessor, C 456  
*#define* 456  
*#endif* 456  
*#ifdef* 456  
*#include* 433, 456
- Präsentation, OSI-Schicht 175
- preg\_match()*, PHP-Funktion 899
- preg\_replace()*, PHP-Funktion 899
- PREROUTING, iptables-Chain 1019
- Primäre Partitionen 142
- Primärschlüssel 657, 1040
- Primärschlüssel, einrichten, SQL 676
- PRIMARY KEY, SQL-Feldoption 676
- print()*, Java-Methode 461
- print*, Perl-Anweisung 473
- printf()*, C-Funktion 434, 453  
*Formatangaben* 434
- println()*, Java-Methode 461
- private*, Java-Kapselung 464
- Problemorientierte Programmiersprachen 47
- Programmablaufplan 91
- Programmgesteuerter Rechenautomat 33
- Programmiersprachen 45  
*Ada* 36  
*Algorithmen* 515  
*Anweisungsblock* 443  
*Assembler* 46  
*BASIC* 48  
*Baum, Datenstruktur* 527  
*Binäre Suche* 523
- Programmiersprachen (Forts.)  
*C* 48, 276, 430  
*C#* 51  
*C++* 50  
*Cobol* 47  
*Compiler* 47, 121  
*Datenstrukturen* 515, 524  
*deklarative* 51  
*Fortran* 47  
*Funktion* 48  
*funktionale* 51  
*GUI-Programmierung* 582  
*Imperative* 48  
*Interpreter* 47, 471  
*Iteration* 520  
*Java* 51, 457  
*Kontrollstrukturen, C* 442  
*Lineare Suche* 522  
*Lisp* 51, 401  
*logische* 51  
*Logo* 51  
*Maschinensprache* 45  
*mit Datenbanken arbeiten* 697  
*Modularisierung* 48, 448  
*Objective-C* 416  
*objektorientierte* 50  
*Objektorientierung* 458  
*Pascal* 48  
*Perl* 471  
*PHP* 893  
*Pipes* 537  
*problemorientierte* 47  
*Prolog* 51  
*Prozedur* 48  
*Prozedurale* 48  
*Rekursion* 520  
*Skriptsprachen* 47, 471  
*SmallTalk* 50  
*Sortier-Algorithmen* 518  
*Strukturierung* 48, 448  
*Such-Algorithmen* 522  
*Systemaufrufe* 122, 537  
*Threads* 541  
*Turing-Vollständigkeit* 99  
*Unicode-Unterstützung* 819
- Programmstrukturierung 48
- Projektmanagement 620  
*Netzplan* 622
- Prolog, Programmiersprache 51
- PROM 124
- Prompt 1040
- prompt()*, JavaScript-Methode 942
- Prompt, UNIX 357
- protected*, Java-Kapselung 466
- Prototype, Ajax-Bibliothek 999
- Prototype, Entwurfsmuster 642
- Proxy, Entwurfsmuster 642
- Prozedur 48
- Prozedurale Programmiersprachen 48
- Prozessanalyse 624
- Prozesse 1040  
*als Baum anzeigen* 382  
*Benutzermodus* 290  
*Child-Prozess* 291  
*Deadlock* 293  
*Definition* 290  
*duplizierte* 537  
*im Benutzermodus* 284  
*im Kernelmodus* 284  
*in den Hintergrund stellen* 359  
*init* 290  
*Kernelmodus* 290  
*Kommunikation* 292  
*Management durch Betriebssystem* 273  
*Multitasking* 289  
*Parent-Prozess* 291  
*Prozess-ID (PID)* 290  
*Race Condition* 293  
*Signalverarbeitung* 291  
*Threads als Alternative* 293  
*unter UNIX* 290  
*Unterstützung durch CPU* 114, 120  
*Verwaltung durch Betriebssysteme* 289  
*Verwaltung, UNIX* 381  
*Windows* 292
- Prozess-ID 290
- Prozessmanagement 273
- Prozessor 37, 111, 115  
*3D Now!* 118  
*Adressbus-Wortbreite* 116  
*Alpha* 116, 119  
*als Bauteil* 112  
*ALU* 113  
*AMD* 113  
*Arbeitsweise* 119  
*Architektur* 118  
*Athlon* 113, 118

## Index

- Prozessor (Forts.)  
  *Aufbau* 112  
  *Bedingter Sprung* 120  
  *Befehlstabelle* 114  
  *Befehlszeiger* 113  
  *Bestandteile* 113  
  *Bus* 114  
  *Cache* 114  
  *CISC* 118  
  *Datenbus-Wortbreite* 116  
  *der Grafikkarte* 156  
  *Effizienz* 117  
  *FLOPS* 117  
  *Intel* 113  
  *Itanium* 116  
  *Maschinenbefehle* 121  
  *MIPS* 117, 119  
  *MMX* 118  
  *MosTek 6502* 116  
  *Motorola 68000-Familie* 116  
  *Pentium-Familie* 113, 118  
  *Pipeline* 118  
  *PowerPC* 116  
  *Prozesse* 120  
  *Register* 113  
  *Registerwortbreite* 115  
  *RISC* 118  
  *Sprungbefehle* 119  
  *Stack* 120  
  *Steuerbus-Wortbreite* 116  
  *Steuerwerk* 113  
  *Sun SPARC* 119  
  *Taktfrequenz* 116  
  *Übertakten* 117  
  *Unbedingter Sprung* 119  
  *Unterprogramm-Aufrufe* 120  
  *Wortbreite* 115  
  *Wortbreiten-Vergleich* 116  
  *Z80* 116  
Prozessor, virtueller 99  
Prozessorarchitektur 118  
Prozessverwaltung 289  
Prüfbit 131  
Prüfung  
  *IT-Berufe* 30  
ps, UNIX-Befehl 292, 381  
PS/2-Anschluss 138  
PSD, Bilddateiformat 827  
Pseudocode, zur Algorithmandarstellung 92  
pstree, UNIX-Befehl 382  
PTR-Record (DNS) 751  
public, Java-Kapselung 460, 465  
Public-Key-Verschlüsselung 1024  
Pulswahlverfahren 206  
Punkt 1040  
push, Perl-Anweisung 475  
puts(), C-Funktion 433, 453  
pwd, UNIX-Befehl 372
- ## Q
- q//, Perl-Quoting 478  
QBit 45  
qq//, Perl-Quoting 478  
Qt, Grafikbibliothek 403–404  
Quadratische Komplexität 95  
Qualitätsmanagement 621  
Quantencomputer 45  
Quantifizierer (RegExp) 531  
Quartz, Grafikbibliothek 416  
Queue 1040  
Queue, Datenstruktur 524  
  *Perl* 524  
QUEUE, iptables-Regel 1020  
QuickEdit-Modus, Windows-Eingabeaufforderung 320  
QuickSort 1040  
QuickSort, Algorithmus 521  
  *Funktionsprinzip* 521  
  *Median* 521  
QuickTime, Multimedia-Technologie 416  
QuickTime, Video-Dateiformat 832  
qw//, Perl-Quoting 478
- ## R
- Race Condition 293, 1040  
radio\_button, Rails-Methode 932  
RAID 145, 1040  
  *Advanced Data Guarding* 145  
  *Levels* 145  
  *Mirroring* 145  
  *Stripe Set* 145  
  *Stripe Set mit Parity* 145  
RAID 0 145  
RAID 01 146  
RAID 1 145  
RAID 10 146  
RAID 5 145  
RAID 6 145  
Rails 1040  
Rails → Ruby on Rails  
rails, Generator 922  
RAILS\_ENV, Umgebungsvariable 923  
rake, Ruby-Automatisierung 925  
RAM 111, 122  
  *als Bauteil* 122  
  *Auslagerungsdatei* 295  
  *Bedeutung in der Speicherhierarchie* 115  
  *CMOS* 126  
  *DDR-RAM* 123  
  *der Grafikkarte* 156  
  *DIMM-Module* 123  
  *Dynamic* 122  
  *EDO* 123  
  *einbauen* 123  
  *empfohlene Menge* 124  
  *FP* 123  
  *Paging* 295  
  *Rambus* 123  
  *RD-RAM* 123  
  *RIMM-Module* 123  
  *SD-RAM* 123  
  *Segmentierung* 294  
  *Seitenfehler* 295  
  *SIMM-Module* 123  
  *Static* 122  
  *Verwaltung durch das Betriebssystem* 294  
  *virtuelle Adressierung* 294  
Rambus-RAM 123  
RAMDAC (Grafikkarte) 157  
Random Access Memory → RAM  
Rangfolge, durch Klammern ändern 442  
Rangfolge, Operatoren 442  
Rational Unified Process 630  
Rationale Zahlen 69  
rd → rmdir, Windows-Befehl  
RDBMS 657, 1040  
RD-RAM 123  
readdir, Perl-Anweisung 483  
Read-Host, PowerShell-CmdLet 329  
readLine(), Java-Methode 461  
Read-only Memory → ROM  
readyState, Ajax-Eigenschaft 984  
REAL, SQL-Datentyp 674

- Rechenautomat 33
- Rechenbefehle, des virtuellen Prozessors 101
- Rechendaten 654
- Rechenmaschinen, mechanische 35
- Rechentafel 35
- Rechenwerk 109
- Record (Datensatz) 656
- Red Book (Audio-CD) 148
- Red, Green, Refactor (TDD) 649
- redirect\_to, Rails-Methode 929
- Reelle Zahlen 69
- Refactoring 613
- Referenzen, auflösen, Perl 477
- Referenzen, Perl 477
- regedit, Windows-Dienstprogramm 339
- RegExp → Reguläre Ausdrücke
- Regionale Einstellungen, Mac OS X 422
- Register 1040
  - der CPU 113
  - des virtuellen Prozessors 99
  - Wortbreite 115
- Register (Schaltung) 89
- Registermaschine 99, 1040
- Registrierdatenbank → Registry, Windows
- Registry, Windows 339
- Reguläre Ausdrücke 529, 1040
  - !~, Operator 534
  - =~, Perl-Operator 534
  - Alternative Textteile 533
  - Alternative Zeichen 531
  - beliebig viele Zeichen 531
  - ein oder mehr Zeichen 531
  - ersetzen 534
  - Escape-Sequenzen 532
  - grep 375
  - Groß- und Kleinschreibung ignorieren 533
  - in PHP 899
  - JavaScript 951
  - Klammern 533
  - Leerzeichen 532
  - m/, Operator 534
  - m/-Operator 530
  - Matching 534
  - mehrere Treffer finden 533
  - Modifikatoren 533
- Reguläre Ausdrücke (Forts.)
  - Muster 530
  - optionale Zeichen 531
  - Perl 483, 529
  - Quantifizierer 531
  - Ruby 536
  - s///-Operator 530, 534
  - Sonderzeichen 532
  - Teilausdrücke 533
  - Whitespace 532
  - Wortgrenzen 532
  - Wortzeichen 532
  - Zeichen ausschließen 531
  - Zeichenanzahl 532
  - Zeichengruppen 531
  - Zeilenanfang 533
  - Zeilenende 533
  - Ziffern 532
- reiserfs, Linux-Dateisystem 378
- REJECT, iptables-Regel 1020
- Rekursion 520, 1040
- Relais 36
- Relationale Datenbak, Join-Abhängigkeiten 662
- Relationale Datenbank 654, 657, 1040
  - 1:1-Relation 658
  - 1:n-Relation 658
  - Access 663
  - Änderungsabfrage 672
  - Arten 663
  - atomare Information 661
  - Auswahlabfrage 660, 672
  - Boyce-Codd-Normalform 661
  - Desktop-Datenbank 663
  - Einfügeabfrage 672
  - FileMaker 663
  - freie Server 664
  - Fremdschlüssel 657
  - Grenzen 664
  - Index 657
  - Java-Programmierung 697
  - JDBC 697
  - kommerzielle Server 664
  - Konsistenz 657
  - Löschabfrage 672
  - m:n-Relation 658
  - MySQL 667
  - Normalformen 661
  - Normalisierung 661
  - ODBC 697
- Relationale Datenbank (Forts.)
  - PostgreSQL 664
  - Primärschlüssel 657
  - Primärschlüssel einrichten, SQL 676
  - Programmierung 697
  - Relationen 657
  - Schlüssel 657
  - SQL 661
  - Tabelle erzeugen, SQL 673
  - Tabelle löschen, SQL 673
- Relationale Datenbanken, SQL 671
- Relationen
  - im RDBMS 657
- Relativer Pfad 298, 301
- Remote Method Invocation → RMI, Java
- rename, Windows-Befehl 322
- repaint(), AWT-Methode 592
- Replikation 168
  - MySQL 695
- Request For Comments → RFC
- Resource Fork, HFS 423
  - Creator ID 423
  - File Type ID 423
- responseText, Ajax-Eigenschaft 985
- responseXML, Ajax-Eigenschaft 989, 991
- Ressourcen
  - Hardware- 132
  - Plug and Play 134
  - Zuweisung 134
- Ressourcenmanagement 621
- ResultSet, JDBC-Klasse 699
- return, C-Anweisung 434, 447
- RETURN, iptables-Regel 1020
- REVOKE, MySQL-Anweisung 688
- RFC 169, 1040
  - 1034 und 1035, DNS 247
  - 1300 169
  - 1723, RIP-2 236
  - 1738, URL 263
  - 1918, private IP-Adressen 217
  - 2045 bis 2049, MIME 257
  - 2060, IMAP 261
  - 2131 und 2123, DHCP 238
  - 2178, OSPF 236
  - 2324, HTCP 169

## Index

- RFC (Forts.)  
2460, IPv6 226  
2616, HTTP 263  
2821, SMTP (Neufassung) 255  
2822, Textnachricht 257  
3330, Spezial-IP-Adressen 217  
768, UDP 240  
791, IP-Protokoll 214  
793, TCP 240  
821, SMTP 255  
822, Textnachricht 257  
854, Telnet 251  
959, FTP 252  
977, NNTP 261
- RGB-Farbe 54
- Rhapsody (Mac OS X) 415
- RIMM-Module (RAM) 123
- Ring-Topologie, Netzwerk 183
- RIP 1040
- RIP, Routing-Protokoll 235
- RISC 1040
- RISC-Prozessor 118  
Beispiele 119
- Risikomanagement 622
- Ritchie, Dennis 48, 276, 431
- rm, UNIX-Befehl 371
- rmdir, UNIX-Befehl 373
- rmdir, Windows-Befehl 322
- RMI (Remote Method Invocation) 1040
- RMI, Java 573
- robots.txt, Suchmaschinen-Info 880
- Röhrenmonitor 157
- Röhrenrechner 36
- Rollback (Transaktionen) 663
- ROLLBACK, SQL-Anweisung 683
- Rollover-Effekt  
JavaScript 958  
praktische JavaScript-Lösung 959
- ROM 111, 124  
Bauarten 124  
Bedeutung 112  
bei 8-Bit-Homecomputern 111  
BIOS 124
- Römische Zahlen 73
- root, Benutzer 292, 355  
Home-Verzeichnis 298  
temporär arbeiten als 362
- root, UNIX-Verzeichnis 298
- root, unter Mac OS X 422
- Rootkits 1015
- ROT13 1023
- Roter Balken (Unit-Test) 648
- Routing-Tabelle 233
- Round-Robin-DNS 752
- route, UNIX-Befehl 408
- Router 1040
- Router, IP-Protokoll 230
- Routing  
Autonomes System 234  
DE-CIX 235  
IP-Protokoll 230
- Routing-Protokolle 234  
BGP 237  
OSPF 236  
RIP 235
- Routing-Tabellen, anzeigen 233
- rpm, Linux-Paketmanager 386
- RS-232 138
- RS-Flip-Flop 88
- RSS, Newsfeed-Format 936
- Ruby 1040  
gsub, Methode 536  
rake, Automatisierungstool 925  
Reguläre Ausdrücke 536  
Singleton, Modul 646  
sub, Methode 536
- Ruby on Rails 920, 1040  
Action Mailer 921  
Action Pack 921  
Action Web Service 921  
Active Record 921  
Active Support 921  
Anwendung erzeugen 922  
Anwendung testen 927  
belongs\_to, Relation 925  
Capistrano, Deployment-Tool 933  
console 926  
Controller 928  
Convention over Configuration 921  
create, Methode 926  
create\_table, Methode 924  
database.yml, Konfigurationsdatei 922  
Datensätze ändern 926  
Deployment 933  
Designprinzipien 921  
Don't Repeat Yourself 921
- Ruby on Rails (Forts.)  
eRuby 928  
find, Methode 926  
flash, Zwischenspeicher 932  
form\_tag, Methode 932  
Formulare 932  
Fremdschlüssel 925  
Generatoren 922  
has\_and\_belongs\_to\_many, Relation 925  
has\_many, Relation 925  
Helper 930  
Installation 922  
Interaktive Konsole 926  
Komponenten 921  
Migration 924  
Model erzeugen 923  
Mongrel, Webserver 933  
Pluralisierung 923  
Praxisbeispiel 922  
radio\_button, Methode 932  
rails, Generator 922  
RAILS\_ENV, Umgebungsvariable 923  
rake db:migrate 925  
redirect\_to, Methode 929  
Relationen 925  
Scaffolding 927  
Startseite 929  
submit\_tag, Methode 933  
Verzeichnisstruktur 922  
Views 928  
WEBrick 927
- Ruby on Rails → Rails
- Rumbaugh, James 633
- run(), Java-Methode 542
- Runlevel 385
- Runnable, Interface 542
- Runnable, Java-Interface 469

## S

- s//, Perl-Operator 484, 530, 534
- Samba 412, 1040  
als Client für Windows-Server 413  
Drucker freigeben 413  
globale Parameter 413  
Konfiguration 412  
smb.conf, Konfigurationsdatei 412

- Samba (Forts.)
  - starten* 412
  - Verzeichnisse freigeben* 413
  - Windows-Freigabeart* 413
- Sampling 1041
- Sampling, Audio 54
- Sampling-Rate 54
- Sampling-Tiefe, Audio 54
- sash (Stand-Alone Shell) 358
- Satellit, DSL-Verbindung 210
- SAX 1041
  - Beispielprogramm* 807
  - ContentHandler, Interface* 801
  - ContentHandler-Callbacks* 802
  - ContentHandler-Methoden* 802
  - Dokument parsen* 801
  - DTDHandler, Interface* 801
  - EntityResolver, Interface* 801
  - ErrorHandler, Interface* 801
  - Event-Handler* 801
  - InputSource, Klasse* 801
  - Parser-Instanz erzeugen* 800
  - XMLReader-Interface* 800
- SAX (Simple API for XML) 800
- SaX (X-Server-Konfigurationsprogramm) 402
- sbin, UNIX-Verzeichnis 298
- Scaffolding (Rails) 927
- scanf(), C-Funktion 445, 453
- Scanner 154
  - Auflicht* 154
  - Diascanner* 155
  - Durchlicht* 154
  - Flachbettscanner* 154
  - Kleinbildscanner* 155
  - Trommelscanner* 155
- Schaltalgebra 26, 82
- Schaltkreisvermittlung 166
- Schaltung, Register 89
- Schema (LDAP) 754
- Schichtenmodelle 172, 1041
  - Alltagsbeispiel* 178
  - Mail-Beispiel* 180
  - OSI-Referenzmodell* 172
  - Praxis* 178
  - TCP/IP* 175
- Schleifen
  - C* 446
  - do/while()* 446
  - Endlosschleifen* 522
  - for()* 447
- Schleifen (Forts.)
  - foreach, Perl* 475
  - fußgesteuerte* 446
  - in der PowerShell* 330
  - in Shell-Skripten* 390
  - kopfgesteuerte* 446
  - mit break abbrechen* 522
- Schleifenrumpf 446
- Schleifenzähler, Variable 446
- Schlüssel
  - eines Hashes* 476
  - in RDBMS* 657
- Schlussfolgerung, logische 60
- Schlussfolgerung, Umkehrschluss 60
- Schnittmenge 70
- Schnittstellen, Hardware 112
- Schnittstellen, Softwareentwurf 627
- Schreib-Lese-Kopf, der Turing-Maschine 96
- Schriftarten, im Drucker 161
- screen, JavaScript-Objekt 969
- script.aculo.us, Ajax-Bibliothek 999
- SCSI 136, 1041
  - anschießen* 136
  - ID* 137
  - Terminator* 136
- SCSI-ID 137
- SDI, Windows-Anwendungen 313
- SD-RAM 123
- SDSL 209
- sed 471, 529
- Sedezimalsystem → Hexadezimalsystem
- Segmentierung (Speicher) 294
- Seiten (Speicher) 295
- Seitenfehler (Speicher) 295
- Seitentabelle (Speicher) 295
- SELECT, SQL-Abfrage 677
- Semikolon, Abschluss von Anweisungen 433
- send(), Ajax-Methode 984
- Sequenzdiagramme (UML) 638
- Serialisierung 469
- Serializable, Java-Interface 469
- Serielle Datenübertragung 130
  - Bedeutung* 131
  - Kontrollbit* 130
- Serielle Datenübertragung (Forts.)
  - Leitungskonventionen* 131
  - Prüfbit* 131
  - Startbit* 130
  - Stoppbit* 131
- Server, Netzwerk 185
- Server, Windows-Betriebssysteme 306
- Serverdienste
  - Anwendungsserver* 189
  - Application Server* 189
  - Dateiserver* 186
  - Druckserver* 187
  - Einrichten, Linux* 409
  - Einrichten, Mac OS X* 425
  - Einrichten, UNIX* 409
  - Einrichten, Windows* 346
  - File-Server* 186
  - HTTP-Server* 188
  - Mail-Server* 187
  - netatalk* 414
  - Print-Server* 187
  - Samba* 412
  - Übersicht* 185
  - Websserver* 188
- Server-Gefahren 1014
- Serversysteme, Macintosh 425
- Servlets, Java 578
- Session-Hijacking 1016
- Sessions (PHP) 911
- SET PASSWORD, MySQL-Anweisung 686
- SET, SQL-Befehl 681
- SET, SQL-Datentyp 676
- setColor(), AWT-Methode 586
- setLayout(), AWT-Methode 594
- Setter-Methoden, Java 613
- setTimeout(), JavaScript-Methode 956
- SGML 1041
  - HTML-DTD* 840
  - XML als moderne Version* 769
- sh (Bourne-Shell) 358
- Shannon-Theorem 1041
- Share Level Security, Samba 413
- Share Level Security, Windows-Freigabeart 346
- Shebang
  - bei Shell-Skripten* 389
  - Perl* 472

## Index

- Shell 288, 1041
  - /etc/profile, Konfigurationsdatei* 358
  - Ausgabeumleitung* 367
  - bash* 358
  - Befehle, Windows* 322
  - Bourne-Shell* 358
  - Cmd.exe, WinNT* 319
  - COMMAND.COM, MS-DOS* 319
  - C-Shell* 358
  - Eingabeaufforderung* 357
  - Eingabeumleitung* 367
  - Eingabevollständigkeit* 361
  - ermitteln, welche läuft* 357
  - Escape-Sequenzen* 377
  - HIER-Dokument* 367
  - History* 361
  - Korn Shell* 358
  - Pipes* 368
  - Prozesse in den Hintergrund stellen* 359
  - Shell-Skripte* 389
  - Stand-Alone Shell* 358
  - Umgebung* 358
  - Umgebungsvariablen* 358
  - UNIX* 288
  - unter Mac OS X* 417
  - Windows* 319
- shell-Befehle als root ausführen 362
- Shell-Skripten 389
  - Beispiele* 391
  - case-Befehl* 390
  - Fallunterscheidungen* 389
  - for-Befehl* 390
  - if-Befehl* 389
  - Schleifen* 390
  - Shebang* 389
  - Variablen* 391
  - while-Befehl* 391
- shift, Perl-Anweisung 475
- short, C-Datentyp 437
- Short-Circuit-Logik 480
- show(), AWT-Methode 584
- shutdown, UNIX-Befehl 382
- Sicherheit 1001
  - Admin-Rechte* 1007
  - AdWare* 1010
  - Backdoors* 1008
  - Backups* 1006
- Sicherheit (Forts.)
  - CGI u. Webanwendungen* 1016
  - chroot-Umgebung* 1018
  - Crackerangriffe* 1015
  - Crackertools* 1018
  - Cross-Site-Scripting (XSS)* 1016
  - Dialer* 1009
  - Exploits* 1015
  - Firewall* 1006
  - Firewalls* 1017
  - Hoaxes* 1014
  - Intrusion Detection Systems* 1018
  - keine absolute* 1001
  - Kettenmails* 1014
  - Kryptografie* 1023
  - Man-in-the-Middle-Angriff* 1016
  - Menschliches Versagen* 1019
  - MySQL (UNIX)* 668
  - MySQL (Windows)* 670
  - Passwörter* 1019
  - Patches installieren* 1006
  - PC-Gefahren* 1002
  - Phishing* 1011
  - Rootkits* 1015
  - Server-Gefahren* 1014
  - Session-Hijacking* 1016
  - Social Engineering* 1019
  - Spam* 1012
  - SpyWare* 1010
  - SQL-Injection* 1016
  - Viren* 1002
  - Würmer* 1004
- Sicherung, OSI-Schicht 174
- SIGALRM, Signal 291
- SIGHUP, Signal 291
- SIGINT, Signal 382
- SIGKILL, Signal 291, 382
- Signal 1041
- Signale
  - an Prozesse senden* 382
  - SIGALRM* 291
  - SIGHUP* 291
  - SIGINT* 382
  - SIGKILL* 291, 382
  - SIGTERM* 291, 382
  - Verarbeitung durch Prozesse* 291
  - zur Inter-Prozess-Kommunikation* 292
- Signatur, digitale 1024
- signed, C-Datentyp 437
- SIGTERM, Signal 291, 382
- Silicon Valley 38
- Silizium 38
- SIMM-Module (RAM) 123
- Simulation, eines Prozessors 99
- Sinclair ZX Spectrum 278
- Sinclair ZX81 42, 278
- Sinclair, Clive 42
- Single Document Interface → SDI, Windows-Anwendungen
- Singleton, Entwurfsmuster 642–643
  - Implementierung (Java)* 645
  - Implementierung (Ruby)* 645
  - Ruby-Modul* 646
- Sinuskurve, zeichnen, AWT 586
- Sitzung, OSI-Schicht 175
- sizeof(), PHP-Funktion 897
- Skalare, Perl-Variablen 473–474
- Skriptsprachen 47, 471
- Slash, Blog-Tool 936
- Slave-Nameserver 249
- Slots, Prozessoren 113
- SMALLINT, SQL-Datentyp 674
- Smalltalk 618
- SmallTalk, Programmiersprache 50
- smbclient, Samba-Dienst 413
- SMB-Protokoll, Windows-Netzwerk 270
- SMTP 255
  - Befehle* 256
  - Sitzung* 256
- sn, LDAP-Attribut 755
- Snort 1018
- SOAP 1041
  - Clients* 581
- SOAP, Web Services 580
- SOA-Record (DNS) 751
- Social Engineering 1019
- sockaddr\_in, Netzwerkprogrammierung 547
- Socket, für Prozessoren 113
- Socket, Perl-Modul 546
- Sockets 545, 1041
  - accept(), Funktion* 551
  - Adressen* 547
  - bind(), Funktion* 550
  - connect(), Funktion* 549

- Sockets (Forts.)  
*Datagramme senden und empfangen* 548  
 Domain 546  
 erzeugen 546  
 IO 551  
 IP-Adresse 547  
 lauschende 550  
 listen(), Funktion 550  
 Protokoll 547  
 TCP 549  
 TCP-Client 549  
 TCP-Port 547  
 TCP-Server 550  
 Typ 546  
 UDP 548  
 Verbindung aufnehmen 551
- Software installieren, Windows 335
- Software-Engineering 617  
 Agile Entwicklungsprozesse 631  
 Analyse 623  
 CASE-Tools 634  
 Code-Review 628  
 Dokumentation 628  
 Entwicklungsprozesse 629  
 Entwicklungszyklus 619  
 Entwurfsmuster 639  
 Entwurfsphase 626  
 Extreme Programming 631  
 Implementierungsphase 627  
 in der IT-Ausbildung 618  
 Lastenheft 624  
 mit Eclipse 608  
 Objektorientierte Analyse 624  
 Objektorientierung 617  
 Pflichtenheft 625  
 Planungsphase 620  
 Projektmanagement 620  
 Projektphasen 619  
 Refactoring 613  
 Schnittstellen 627  
 Spiralmodell 619  
 Strukturierte Analyse 624  
 Test-first-Verfahren 632, 649  
 Testphase 628  
 UML 633  
 Unified Process 630  
 Unit-Tests 628, 646  
 Wasserfallmodell 619
- Softwareentwicklung → Software-Engineering  
 Softwarekrise 617  
 Softwaretechnik → Software-Engineering  
 Solaris 1041  
 Solaris, Betriebssystem 277  
 Sortier-Algorithmen 518  
   BubbleSort 518  
   QuickSort 521  
 Soundkarte 162  
   Anschlüsse 162  
   Audio-CDs abspielen 162  
   MIDI 162  
   SP-DIF-Anschlüsse 162  
 Source, Stromeingang des Transistors 83  
 Spam 1012  
 SpamAssassin 1013  
 SP-DIF-Anschlüsse 162  
 Special Files → Gerätedateien  
 Spectrum, Homecomputer 278  
 Speicher  
   Management durch Betriebssystem 273  
   RAM 111  
   ROM 111  
   virtueller 115  
 Speicher reservieren, C 526  
 Speicheradressierung 78  
 Speichermanagement 273  
 Speicherseiten 295  
 Speicherverwaltung 294  
   x86-Systeme 295  
 Speicherzelle (Schaltung) 89  
 Spiralmodell 619  
 split, Perl-Anweisung 476  
 Sprache (Umgangssprache), zur Algorithmen­darstellung 91  
 Sprungbefehle  
   bedingte 120  
   beim virtuellen Prozessor 103  
   der CPU 119  
   unbedingte 119  
 Sprungvorhersage 115  
 Sprungvorhersage → Prozessor 115  
 SpyWare 1010  
 SQL 661, 671, 1041  
   Aggregatfunktionen 678  
   Änderungsabfrage 672, 681
- SQL (Forts.)  
 Arithmetische Operatoren 678  
 AS-Klausel 678  
 Auswahlabfrage 672, 677  
 AUTO\_INCREMENT, Feldoption 676  
 BIGINT, Datentyp 674  
 BINARY, Feldoption 676  
 BLOB, Datentyp 675  
 CHAR, Datentyp 675  
 COMMIT, Anweisung 683  
 COUNT-Funktion 679  
 CREATE DATABASE, Befehl 672  
 CREATE TABLE, Befehl 673  
 DATE, Datentyp 675  
 Datentypen 674  
 Datentypen in Java 700  
 Datentypen, Aufzählung 676  
 Datentypen, Binärobjekte 675  
 Datentypen, Datum und Uhrzeit 675  
 Datentypen, Fließkomma 674  
 Datentypen, ganzzahlige 674  
 Datentypen, Text 675  
 DATETIME, Datentyp 675  
 DECIMAL, Datentyp 674  
 DEFAULT, Feldoption 676  
 DELETE-Abfrage 681  
 DOUBLE, Datentyp 674  
 DROP DATABASE, Befehl 673  
 DROP TABLE, Befehl 673  
 Einfügeabfrage 672, 680  
 ENUM, Datentyp 676  
 Felder mit Nullen füllen 676  
 Feldoptionen 676  
 Feldwert einmalig machen 676  
 FLOAT, Datentyp 674  
 Funktionen 678  
 Indizes erstellen 676  
 Inner Join 679  
 Inner Join durch WHERE ausdrücken 680  
 INNER JOIN-Klausel 680  
 INSERT-Abfrage 680  
 INT, Datentyp 674  
 Joins 679  
 LIKE-Klausel 678  
 LONGBLOB, Datentyp 675  
 LONGTEXT, Datentyp 675  
 Löschafrage 672, 681  
 MAX-Funktion 679

## Index

- SQL (Forts.)
  - MEDIUMBLOB*, Datentyp 675
  - MEDIUMTEXT*, Datentyp 675
  - MIN*-Funktion 679
  - Mustervergleich* 678
  - NULL*, Feldoption 676
  - ORDER BY*-Klausel 679
  - Primärschlüssel einrichten* 676
  - PRIMARY KEY*, Feldoption 676
  - REAL*, Datentyp 674
  - ROLLBACK*, Anweisung 683
  - SELECT*-Abfrage 677
  - SET*, Datentyp 676
  - SET*-Befehl 681
  - SMALLINT*, Datentyp 674
  - Sortieren* 679
  - Standardwert angeben* 676
  - START TRANSACTION*, Anweisung 682
  - SUM*-Funktion 679
  - TEXT*, Datentyp 675
  - TIME*, Datentyp 675
  - TIMESTAMP*, Datentyp 675
  - TINYBLOB*, Datentyp 675
  - TINYINT*, Datentyp 674
  - TINYTEXT*, Datentyp 675
  - Transaktion beginnen* 682
  - UNSIGNED*, Feldoption 676
  - UPDATE*-Abfrage 681
  - VARCHAR*, Datentyp 675
  - Vergleichsoperatoren* 678
  - Volltextindex* 676
  - WHERE*-Klausel 677
  - YEAR*, Datentyp 675
  - ZEROFILL*, Feldoption 676
- SQL-Injection 1016
- SQL-Server, Microsoft 664
- SRAM 122
- SSH 1025, 1041
- SSH (Secure Shell) 251
  - OpenSSH* 251
- SSL 1025
- Stack
  - der CPU* 114, 120
  - des virtuellen Prozessors* 103
  - Java-Klasse* 470
- Stack Pointer → Stack-Zeiger
- Stack, Datenstruktur 524
  - Perl* 524
- Stack-Overflow 120, 1041
- Stack-Zeiger 114, 120
- Stallman, Richard 281
- Stammdaten 653
- Stand-Alone Shell (sash) 358
- Standardausgabe (stdout) 367
- Standardausgabe, Java 461
- Standardeingabe (stdin) 367
- Standardfehlerausgabe 544
- Standard-Fehlerausgabe (stderr) 367
- Stapelverarbeitung 275
- START TRANSACTION, SQL-Anweisung 682
- Startbit 130
- Startvolumen, Mac OS X-Einstellung 421
- startx, X Window starten 402
- State, Entwurfsmuster 643
- Statement, JDBC-Klasse 699
- Static RAM → SRAM
- static, Java-Methoden 460
- static, Variableneigenschaft, C 437
- Statische Variablen 437
- stddef.h, C-Bibliothek 437
- stderr 544
- stderr, Standard-Fehlerausgabe 367
- stdin, Standardeingabe 367
- stdio.h, C-Bibliothek 453
- stdio.h, C-Header-Datei 433
- stdlib.h, C-Header-Datei 433
- stdout, Standardausgabe 367
- Stealth-Viren 1004
- Stellenwertsystem 35, 72
  - Basis* 73
- Stern-Topologie, Netzwerk 183
- Steuerbus 114
  - Wortbreite* 116
- Steuerwerk 109
  - der CPU* 113
- Stoppbit 131
- Strategy, Entwurfsmuster 643
- strcat(), C-Funktion 455
- strcmp(), C-Funktion 454
- strcpy(), C-Funktion 455
- Stream-Sockets 546
- strftime(), C-Funktion 455
- strict, Perl 474
- String 433, 1041
  - aus Datei lesen*, C 454
  - Eingabe*, C 453
- String, Java-Datentyp 461
- string.h, C-Bibliothek 454
- String-Literale 438
- Strings
  - aufteilen*, Java 463
  - aus Arrays*, Perl 476
  - Darstellung in C* 451
  - Darstellung*, C 433
  - einlesen*, C 434
  - einzelne Zeichen lesen*, Java 463
  - Funktionen in C* 454
  - in GUI schreiben* 588
  - Java* 461
  - Kopieren*, C 455
  - Länge ermitteln*, Java 463
  - Null-terminierte* 451
  - Position ermitteln*, Java 463
  - vergleichen*, C 454
  - vergleichen*, JavaScript 944
  - vergleichen*, Perl 479
  - Verkettung*, C 455
  - Verkettung*, Java 461–462
  - Verkettung*, Perl 479
  - zerlegen*, Perl 476
  - zerlegen*, PHP 898
- Strings, Operationen in Java 462
- String-Verkettung, Perl 479
- Stripe Set (RAID) 145
- Stripe Set mit Parity (RAID) 145
- Stroustrup, Bjarne 50
- struct, C 451
- Structural Patterns → Strukturmuster
- Structured Query Language → SQL
- Strukturen, C 451
- Strukturierte Analyse 624
- Strukturierung von Programmen 48
- Strukturierung, Programme 48, 448
- Strukturmuster 640
- Studiengänge, Informatik 31
- Style Sheets → CSS
- su, UNIX-Befehl 362
- sub, Perl-Subroutine 481
- sub, Ruby-Methode 536
- Subdomain (DNS) 247
- submit\_tag, Rails-Methode 933
- Subnet Mask 1041
- Subnet Mask, IP-Adresse 218

- Subnetting 1041
  - Subnetting, IP-Netze teilen 219
  - Subroutinen
    - Parameter* 481
    - Wertrückgabe* 482
  - Subroutinen, Perl 481
  - SubSeven, Backdoor 1008
  - substring(), Java-Methode 463
  - Subtraktion, Operator 439
  - Such-Algorithmen 522
    - Binäre Suche* 523
    - Lineare Suche* 522
  - Suchmaschinen
    - Anmeldung bei* 880
    - HTML aufbereiten für* 878
    - robots.txt-Datei* 880
  - SUM, SQL-Funktion 679
  - Sun Microsystems
    - Java* 457
    - Solaris, Betriebssystem* 277
  - Sun SPARC, Prozessor 119
  - super, Java 467
  - Supernetting 1041
    - IP-Netze zusammenfassen* 219
  - Superuser 292, 355
  - SUSE, Linux-Distribution 351
  - SVG 1041
  - SVG (Scalable Vector Graphics) 770
  - Swap-Partition 295
  - Swapping (Speicher) 115
  - Swing, Java 583
    - Ereignisbehandlung* 583
    - Tabellen* 583
  - Switch 196
  - switch/case-Fallunterscheidung 445
    - default-Wert* 446
  - Symbolic Links 299
  - Symbolische Konstante 456, 1041
  - Symmetrische Verschlüsselung 1024
  - sys/types.h, C-Bibliothek 538
  - Syslog 382–383, 1041
  - System V 1041
  - System V Init 384
  - System V IPC 292
  - System V, UNIX 276
  - System.err, Java 544
  - System.out, Java 461
  - Systemanalyse 623
  - Systemaufrufe 122, 284, 287, 1041
    - CreateProcess()* 292
    - fork()* 290
    - kill()* 291
    - pause()* 291
    - programmieren* 537
    - UNIX* 287
    - Win32 API* 288
  - Systembefehle
    - UNIX* 369
    - UNIX-Dateimanipulation* 370
    - UNIX-Systemverwaltung* 377
    - UNIX-Textmanipulation* 374
  - Systemkonfiguration
    - Mac OS X* 421
    - Windows* 334
  - Systemprogramme 286
    - Linux* 280
    - Optionen* 359
    - Parameter* 359
    - UNIX* 369
    - UNIX-Dateimanipulation* 370
    - UNIX-Systemverwaltung* 377
    - UNIX-Textmanipulation* 374
  - Systemsteuerung, Windows 334
  - Systemvariablen 1042
    - PATH (UNIX)* 360
    - setzen (UNIX)* 360
  - Systray, Windows 316
- ## T
- 
- Tabelle
    - erzeugen, SQL* 673
    - löschen, SQL* 673
  - Tabellen (iptables) 1020
  - Tabellen, HTML 860
  - Tags, HTML 841
  - Tags, XML 771
  - tail, UNIX-Befehl 375
  - Taktfrequenz
    - der CPU* 116
    - der Mainboard* 116
    - Multiplikator* 116
    - praktische Bedeutung* 117
  - Tanenbaum, Andrew 283
  - tar, UNIX-Befehl 387, 833
  - tar-Datei 387
  - Task Scheduler 284, 1042
  - Taskleiste, Windows 314
    - Systray* 316
  - Tastatur 153
    - Zeichensatzeinstellung* 819
  - Tastenkürzel, Windows-Eingabeaufforderung 319
  - TCP 241, 1042
    - Drei-Wege-Handshake* 242
    - Funktionsweise* 241
    - im Vergleich zu UDP* 240
    - Paket-Header* 241
    - Ports* 243
    - Urgent Data* 244
    - Verbindungsaufbau* 242
  - TCP/IP 212, 1042
    - Adressierung* 213
    - Anwendungsprotokolle* 250
    - ARP, Netzzugang* 213
    - DHCP* 238
    - Dienstprogramme* 343
    - DNS* 246
    - Domain Name System* 246
    - FTP, Anwendungsprotokoll* 252
    - HTTP, Anwendungsprotokoll* 263
    - ICMP-Protokoll* 240
    - IMAP, Anwendungsprotokoll* 260
    - Loopback-Interface* 217
    - Nameserver* 246
    - Netzzugang* 212
    - NNTP, Anwendungsprotokoll* 261
    - POP3, Anwendungsprotokoll* 259
    - Routing* 230
    - Routing-Protokolle* 234
    - SMTP, Anwendungsprotokoll* 255
    - Telnet, Anwendungsprotokoll* 251
    - Transportprotokolle* 240
  - TCP/IP-Dienstprogramme 343
    - netstat* 344
    - nslookup* 345
    - ping* 343
    - tracert* 344
  - TCP/IP-Schichtenmodell 175
  - TCP-Client-Socket, mit IO 552
  - TCP-Client-Sockets 549
  - TCP-Header 241

## Index

- TCP-Port, für Sockets 547
- TCP-Ports 243
  - Well-Known Ports* 243
- TCP-Server-Socket, mit IO 552
- TCP-Server-Sockets 550
- tcsh (erweiterte C-Shell) 358
- TDD → Test-driven Development
- Technische Informatik 26
- teilerfremd 516
- Teilmenge 67
- Teilnetzmaske, IP-Adresse 218
- Telefongespräch als Schichtenmodell 178
- Telefonleitung
  - Pulswahl* 206
- Telefonverbindung 166
- telephoneNumber, LDAP-Attribut 756
- Telnet 251, 1042
- Template Method, Entwurfsmuster 643
- Terabyte 79
- Term 58
- TERM, Signal 291, 382
- Terminal 39, 275, 1042
  - unter grafischer Oberfläche* 356
  - virtuelles* 356
- Terminator 1042
  - SCSI 136
- Test, Software-Engineering 628
  - Code-Review* 628
  - Lasttests* 628
  - Unit-Tests* 628, 646
- Test-driven Development 649, 1042
- Test-first-Verfahren 632, 649
- Testgetriebene Entwicklung → Test-driven Development
- TeX, Textsatzsystem 822
- Text
  - ausgeben (UNIX)* 374
  - Datei anzeigen, Windows-Konsole* 323
  - Dateien vergleichen* 377
  - Dateinhalt anzeigen (UNIX)* 374
  - Editoren, UNIX* 395
  - Emacs, Editor* 398
  - Manipulationsbefehle, UNIX* 374
- Text (Forts.)
  - vi, Editor* 395
  - Wörter zählen, UNIX* 377
- TEXT, SQL-Datentyp 675
- text-align, CSS-Angabe 886
- TextArea, AWT-Klasse 593
- Textdatei
  - anzeigen, Windows-Konsole* 323
  - Inhalt anzeigen (UNIX)* 374
- Textdateiformate
  - YAML 922
- Textdateiformate, Vorteile 769
- text-decoration, CSS-Angabe 886
- Texteditor
  - Emacs* 398
  - für XML verwenden* 770
  - unter UNIX* 395
  - vi* 395
  - vim* 395
- TextField, AWT-Klasse 593
- text-indent, CSS-Angabe 886
- TFT 1042
- TFT-Monitor 158
- Theoretische Informatik 26
- Thermosublimationsdrucker 161
- Thermotransferdrucker 161
- Thicknet Coaxial, Ethernet 194
- Thinnet Coaxial, Ethernet 194
- this, Java 465
- Thompson, Ken 276
- Thread, Java-Klasse 541
- Threads 283, 293, 1042
  - in Programmiersprachen* 541
  - Java* 469, 541
  - run()* 542
  - Runnable-Interface* 542
- throws-Klausel, Java 470
- TIFF, Bilddateiformat 828
- time(), C-Funktion 455
- TIME, SQL-Datentyp 675
- time.t, C-Bibliothek 455
- time\_t, C-Datentyp 455
- Timesharing 275, 1042
- TIMESTAMP, SQL-Datentyp 675
- Tintenstrahldrucker 160
  - Bubble-Technik* 160
  - Piezo-Technik* 160
- TINYBLOB, SQL-Datentyp 675
- TINYINT, SQL-Datentyp 674
- TINYTEXT, SQL-Datentyp 675
- Titel, HTML-Dokument 842
- TLS → SSL
- Token-Passing 1042
- Token-Passing, Netzzugangsverfahren 197
- Token-Ring-Netzwerk 197, 1042
- Tomlinson, Ray 169
- Tonkanäle, Audio 54
- Tonwahlverfahren 206
- Top Level Domain 1042
- Top Level Domain, Länder 248
- top, CSS-Angabe 889
- Top-Level-Domain (DNS) 247
- Top-Level-Domain, Generic 248
- Topologie (Netzwerk) 183
- Torvalds, Linus 277, 280
- tr///, Perl-Operator 535
- tracert, TCP/IP-Dienstprogramm 344
- Track-at-once 149, 1042
- Transaktionen (Datenbanken) 682
  - Commit* 683
  - in MySQL* 682
  - Rollback* 683
- Transaktionen (RDBMS) 663
  - in MySQL* 663
- Transistor 37–38, 83
  - TFT* 158
- Transistorrechner 37
- Transport, OSI-Schicht 174
- Treiber → Gerätetreiber
- Triode 38
- tripwire 1018
- Trojaner → Trojanische Pferde
- Trojanische Pferde 1007
- Trolltech 404
- Trommelscanner 155
  - Foto-Multiplier* 155
- true, Java 462
- TrueType 1042
- try, Java 461
- try/catch-Blöcke, Java 461
- TTL 1042
- TTL, IP-Datagramm 234
- Turing, Alan 95, 1043
- Turing-Maschine 96, 1043
  - Band* 96
  - Beispiele* 97–98
  - einfaches Beispiel* 97
  - komplexeres Beispiel* 98
  - Schreib-Lese-Kopf* 96

- Turing-Maschine (Forts.)  
*Zeichenvorrat* 96  
*Zustände* 96
- Turing-Test 95, 1043
- Turing-Vollständigkeit 99, 1043
- Tweening 1043
- Twisted-Pair-Kabel 195  
*Kategorien* 195
- type, Windows-Befehl 323
- Typecasting, C 438
- Typenraddrucker 160
- TYPO3, CMS 934
- U**
- 
- Überladen 465
- Übertakten 117
- Ubuntu Linux 352
- UCSD-Pascal 608
- UDDI 1043
- UDDI, Web Services 580
- UDF 152
- UDP 244, 1043  
*Anwendungsbeispiel* 244  
*im Vergleich zu TCP* 240  
*Paketheader* 244  
*Ports* 245
- UDP-Header 244
- UDP-Ports 245
- UID (User-ID) 355  
*von Prozessen* 291
- uid, LDAP-Attribut 756
- uidNumber, LDAP-Attribut 756
- Umbenennen  
*Dateien, Mac OS X* 421  
*Dateien, UNIX* 371
- Umfangsmanagement 621
- Umgebung  
*UNIX* 358
- Umgebungsvariablen 358  
*CLASSPATH* 458  
*PATH (UNIX)* 360  
*RAILS\_ENV* 923  
*setzen (UNIX)* 360  
*Setzen unter Windows* 321
- Umkehrschluss 60
- UML 92, 633, 1043  
*Akteure* 635  
*Aktivitätsdiagramme* 638  
*Anwendungsfalldiagramme* 635  
*ArgoUML, Tool* 634
- UML (Forts.)  
*Diagrammtypen* 633  
*Klassendiagramme* 635  
*Praktischer Einsatz* 634  
*Sequenzdiagramme* 638  
*Version 2.0* 633
- umount, UNIX-Befehl 378
- Umrechnung  
*Dezimal nach Dual* 75  
*Dezimal nach Hexadezimal* 76  
*Dual nach Dezimal* 76  
*Dual nach Hexadezimal* 77  
*Dual nach Oktal* 77  
*Hexadezimal nach Dezimal* 77  
*Hexadezimal nach Dual* 77  
*Oktal nach Dual* 77  
*Zahlensysteme* 75
- unalias, UNIX-Befehl 393
- Unbedingter Sprung 119
- Undefiniiertheitsstelle (Funktion)  
 92
- Und-Schaltung → AND-Schaltung
- Und-Verknüpfung → AND-Verknüpfung
- Ungleichheit 65
- Ungleichheit, Operator 440
- Ungleichung 58  
*Lösung* 59
- Unicode 818, 1043  
*BMP-Teilmenge* 818  
*Tabelle wichtiger Teilzeichensätze* 818  
*Unterstützung durch Software* 819  
*UTF-8-Codierung* 818
- Unicode, Zeichensatz 55
- Unified Modeling Language → UML
- Unified Process 630  
*Aktivitäten* 630  
*Anwendungsfälle* 630  
*Artefakte* 630  
*Phasen* 631  
*Rollen* 630  
*Vorgehen* 630
- UNIQUE, SQL-Schlüsselwort 676
- Unit-Tests 628, 646  
*Grüner Balken* 648  
*JUnit-Framework* 647  
*Motivation* 647
- Unit-Tests (Forts.)  
*Roter Balken* 648
- Universal Disk Format → UDF
- University of California, Berkeley 276, 395, 545
- UNIX 49, 276  
*\$0, Systemvariable* 357  
*.bashrc, Konfigurationsdatei* 358  
*/etc/passwd-Datei* 355  
*/etc/profile, Konfigurationsdatei* 358  
*/etc/shadow, Datei* 356  
*alias-Befehl* 393  
*als Server einrichten* 409  
*Arbeitsverzeichnis anzeigen* 372  
*auf NFS-Freigaben zugreifen* 410  
*auf Windows-Server zugreifen* 413  
*bash* 358  
*Befehle regelmäßig ausführen* 393  
*Benutzerrechte* 300  
*Berkeley System Distribution (BSD)* 276  
*Bourne-Shell* 358  
*BSD* 276  
*BSD-Startskripten* 386  
*bunzip2-Befehl* 836  
*bzip2-Befehl* 835  
*BZip2-Komprimierung* 387  
*cat-Befehl* 374  
*cd-Befehl* 372  
*chgrp-Befehl* 374  
*Child-Prozess* 291  
*chmod-Befehl* 373  
*chown-Befehl* 373  
*cp-Befehl* 371  
*Cronjobs* 393  
*C-Shell* 358  
*CUPS, Drucksystem* 411  
*Daemon* 382  
*Darwin* 277  
*date-Befehl* 379  
*Dateibefehle* 370  
*Dateibesitzer wechseln* 373  
*Dateien kopieren* 371  
*Dateien löschen* 371  
*Dateien umbenennen* 371  
*Dateien verschieben* 371

## Index

### UNIX (Forts.)

*Datei-Endungen* 370  
*Dateigruppe wechseln* 374  
*Dateinamen* 299, 370  
*Dateinamen-Platzhalter* 370  
*Dateisysteme* 297  
*Datum und Uhrzeit ändern* 379  
*Datum und Uhrzeit formatieren* 379  
*diff-Befehl* 377  
*du-Befehl* 378  
*echo-Befehl* 374  
*Emacs, Texteditor* 398  
*Escape-Sequenzen* 377  
*exit-Befehl* 362  
*fg-Befehl* 359  
*finger, Dienstprogramm* 355  
*fork(), Systemaufruf* 290  
*fsck-Befehl* 378  
*Gerätedateien* 297–298  
*GNOME* 289, 403, 405  
*Grafische Benutzeroberfläche* 401  
*grep-Befehl* 375  
*groupadd-Befehl* 380  
*Group-ID* 291, 355  
*gunzip-Befehl* 835  
*gzip-Befehl* 835  
*gzip-Komprimierung* 387  
*Hard Links* 299  
*head-Befehl* 375  
*HIER-Dokument* 367  
*Home-Verzeichnis* 298  
*HP UX* 277  
*IBM AIX* 277  
*ifconfig-Befehl* 407  
*init-Prozess* 290  
*inodes* 299  
*Installation von Software* 387  
*IP-Adresse zuweisen* 408  
*KDE* 289, 403–404  
*kill(), Systemaufruf* 291  
*kill-Befehl* 382  
*Korn Shell* 358  
*less-Befehl* 375  
*Links (Dateisystem)* 299  
*Linux* 277, 280  
*logger-Befehl* 384  
*Login* 353  
*ls-Befehl* 372  
*Mac OS X* 277, 417

### UNIX (Forts.)

*Macintosh im Netz bedienen* 414  
*mail-Befehl* 394  
*make-Befehl* 387  
*man-Befehl* 288  
*Minix* 280  
*mkdir-Befehl* 373  
*mkfs-Befehl* 379  
*Modularität* 276  
*more-Befehl* 375  
*mount-Befehl* 377  
*mv-Befehl* 371  
*MySQL-Installation* 667  
*netatalk-Server* 414  
*Neustart* 382  
*NFS* 410  
*Pager* 375  
*Parent-Prozess* 291  
*passwd-Befehl* 381  
*Passwort ändern* 381  
*patch-Befehl* 377  
*PATH, Umgebungsvariable* 360  
*pause(), Systemaufruf* 291  
*Pfadangaben* 298  
*Pipes* 368  
*POSIX-Standard* 277  
*Programme automatisch starten* 384  
*Prozessmodell* 290  
*Prozessverwaltung* 381  
*ps-Befehl* 292, 381  
*pstree-Befehl* 382  
*pwd-Befehl* 372  
*Reguläre Ausdrücke suchen* 375  
*rm-Befehl* 371  
*rmdir-Befehl* 373  
*root, Benutzer* 292, 355  
*route-Befehl* 408  
*Runlevel* 385  
*Samba-Server* 412  
*Shell* 288, 353  
*Shell-Ausgabeumleitung* 367  
*Shell-Eingabeumleitung* 367  
*Shell-Eingabevollständigung* 361  
*Shell-History* 361  
*Shell-Skripte* 389  
*shutdown-Befehl* 382  
*Software installieren* 387  
*Stand-Alone Shell* 358  
*startx-Befehl* 402

### UNIX (Forts.)

*su-Befehl* 362  
*Sun Solaris* 277  
*Swap-Partition* 295  
*Symbolic Links* 299  
*Syslog* 382–383  
*System herunterfahren* 382  
*System V* 276  
*Systemaufrufe* 287  
*Systemprogramme* 369  
*tail-Befehl* 375  
*tar-Befehl* 387, 833  
*tar-Dateien* 387  
*Textbefehle* 374  
*Textdatei anzeigen* 374  
*Textdateien vergleichen* 377  
*Texteditoren* 395  
*Umgebung* 358  
*Umgebungsvariablen setzen* 360  
*umount-Befehl* 378  
*unalias-Befehl* 393  
*unzip-Befehl* 836  
*useradd-Befehl* 380  
*userdel-Befehl* 380  
*User-ID* 291, 355  
*Verwaltungsbefehle* 377  
*Verzeichnis anlegen* 373  
*Verzeichnis löschen* 373  
*Verzeichnis wechseln* 372  
*Verzeichnisbaum* 297  
*Verzeichnisbefehle* 370  
*Verzeichnisinhalt anzeigen* 372  
*vi, Editor* 395  
*virtuelles Terminal* 356  
*wc-Befehl* 377  
*Window-Manager* 289  
*Wörter zählen* 377  
*X Window* 289, 401  
*zip-Befehl* 836  
*Zugriffsrechte* 300  
UNIX System V Init 384  
UNIX-Benutzerkonten in LDAP abbilden 754  
unless, Perl-Fallunterscheidung 481  
unshift, Perl-Anweisung 475  
unsigned, C-Datentyp 437  
UNSIGNED, SQL-Feldoption 676  
Unterprogramm, Aufruf durch CPU 120  
unzip, UNIX-Befehl 836

- update(), AWT-Methode 592
  - UPDATE, SQL-Abfrage 681
  - URL 1043
    - Schema 265
  - URL, Web-Adresse 263
  - USB 137, 1043
  - Use Cases → Anwendungsfälle
  - use, Perl-Direktive 474
  - Usenet 261, 1043
  - User Level Security, Samba 413
  - User Level Security, Windows-Freigabeart 346
  - useradd, UNIX-Befehl 380
  - userdel, UNIX-Befehl 380
  - User-ID 355
    - von Prozessen 291
  - userPassword, LDAP-Attribut 756
  - Users, Mac OS X-Verzeichnis 298
  - usr, UNIX-Verzeichnis 298
  - UTF-8 1043
  - UTF-8, Unicode-Codierung 818
- V**
- 
- V.24-Schnittstelle 138
  - var, UNIX-Verzeichnis 298
  - VARCHAR, SQL-Datentyp 675
  - Variablen
    - Arrays, Perl 474
    - automatische (lokale) 437
    - C 436
    - Datentypen, C 436
    - Deklaration, C 433, 435–436
    - Deklaration, Java 462
    - deklarieren, Perl 474
    - Existenz testen, Perl 481
    - globale 437
    - Gültigkeitsbereich, C 437
    - Hashes, Perl 474
    - in der PowerShell 327
    - in Programmiersprachen 66
    - in Shell-Skripten 391
    - lokale 437
    - mathematische 59
    - Perl 473
    - PHP 895
    - Schleifenzähler 446
    - Skalare, Perl 473–474
    - statische 437
    - Substitution, Perl 473
  - Variablen (Forts.)
    - Substitution, Shell-Skripten 391
    - Typecasting 438
  - Variablensubstitution
    - in Shell-Skripten 391
  - Variablensubstitution
    - in Perl-HIER-Dokumenten 480
    - Perl 473
  - VAX, Minicomputer-Serie von DEC 40
  - Vektorgrafik 53
  - Vektorrechnung 91
  - Vereinigungsmenge 70
  - Vererbung 50, 466, 1043
    - Interfaces, Java 468
  - Vergleichsoperationen 65
    - beim virtuellen Prozessor 102
    - Umkehrung 65
  - Vergleichsoperatoren 440
    - SQL 678
  - Verhaltensmuster 640
  - Verknüpfungen, logische 60
  - Verschieben
    - Dateien, Mac OS X 421
    - Dateien, UNIX 371
    - Dateien, Windows 317
  - Verschlüsselung → Kryptografie
  - Verteilte Anwendung 572
    - Aufgaben 572
    - Frameworks im Vergleich 572
    - J2EE-Plattform 572
  - Verteilte Anwendungen 190
  - vertical-align, CSS 886
  - Verzeichnis
    - anlegen, UNIX 373
    - anlegen, Windows 322
    - Arbeitsverzeichnis anzeigen, UNIX 372
    - Inhalt anzeigen, UNIX 372
    - löschen, UNIX 373
    - löschen, Windows 322
    - wechseln, UNIX 372
    - wechseln, Windows 322
  - Verzeichnisdienst 1043
    - Active Directory 347
  - Verzeichnisdienste
    - LDAP 753
  - Verzeichnisse, verarbeiten, Perl 483
  - vi, Texteditor 395
    - Befehlsmodus 395
  - vi, Texteditor (Forts.)
    - Dateibefehle 398
    - Editiermodus 395
    - kopieren 397
    - Löschen 397
    - Navigation 396
    - Suchfunktionen 397
  - Video
    - Dateiformate 832
  - Video-Dateiformate 832
    - AVI 832
    - MPEG 833
    - QuickTime 832
  - Video-DVDs
    - auf dem PC abspielen 151
    - Region-Code 151
  - View (Ruby on Rails) 928
  - Vim, Texteditor 395
  - Viren 47, 1002
    - Antivirenprogramme 1005
    - Aufbau 1003
    - Bootsektor-Viren 1003
    - Dateiviren 1004
    - Makroviren 1004
    - polymorphe 1004
    - Schutzmaßnahmen 1005
    - Stealth-Viren 1004
  - Virtual Private Network → VPN
  - Virtuelle Maschine, Java 457
  - Virtueller Prozessor 99
    - Arbeitsspeicher 100
    - Aufbau 99
    - Befehle 101
    - Beispielprogramme 104
    - Flags 102
    - Rechenbefehle 101
    - Register 99
    - Sprungbefehle 103
    - Stack 103
    - Vergleichsoperationen 102
  - Virtueller Speicher 115, 294
  - Visitor, Entwurfsmuster 643
  - Vista → Windows Vista
  - Visual Studio, IDE 608
  - Vlissides, John 639
  - VLSM 1044
  - VLSM, variable IP-Teilnetze 223
  - VMS, Betriebssystem 279
  - void, Datentyp
    - Zeiger auf 526
  - void, Funktionsdatentyp, C 447

## Index

- Volladdierer (Schaltung) 87  
Vollbildfenster, JavaScript 969  
Volltextdatenbank 655  
von Neumann, John 37, 109  
Von-Neumann-Rechner 99, 109, 1044  
VPN 1025
- ## W
- w, Perl-Modifikator 472  
W3C, DOM 807  
Wahre Aussagen 58  
Wahrheitstabelle 62  
    *NAND-Verknüpfung* 84  
    *XOR-Verknüpfung* 64  
Wall, Larry 471  
WAMP-System 894  
WAN 182, 1044  
    *Technische Lösungen* 182  
Warnock, John 41  
Warnungen in Perl aktivieren 472  
Wasserfallmodell 619  
WAV, Audio-Dateiformat 831  
Wavetable-Synthese (MIDI) 163  
wc, UNIX-Befehl 377  
wchar\_t, C-Datentyp 437  
Wearable Computers 44  
Web 2.0 939  
Web Services 579, 1044  
    *Beschreibungsdatei* 580  
    *Clients* 581  
    *Definition* 579  
    *registrieren* 580  
    *SOAP* 580  
    *Sprachen* 580  
    *UDDI* 580  
    *WSDL* 580  
Webanwendungen  
    *Ajax* 981  
    *CGI* 933  
    *Grundprinzip* 893  
    *Sicherheitsprobleme* 1016  
Web-Browser  
    *KDE Konqueror* 404  
WebLogic, JavaApplication Server 573  
Weblogs 935  
    *Blogging-Communities* 936  
    *Software für* 936  
WEBrick, Ruby-Webserver 927  
Web-Safe Colors 885  
Webserver 188  
    *Mongrel* 933  
    *Programmierung* 893  
Websichere Farben 885  
Websites  
    *robots.txt-Datei* 880  
Wechseldatenträger 146  
    *CD-ROM* 147  
    *Diskettenlaufwerk* 146  
    *DVD* 151  
    *Jaz-Laufwerk* 147  
    *LS-120* 147  
    *Zip-Laufwerk* 147  
Weizenbaum, Joseph 95  
Well-Known Ports, Tabelle einiger wichtiger 245  
wertdiskret 54  
Wertetabelle  
    *NOR-Verknüpfung* 84  
Wertrückgabe, Perl-Subroutinen 482  
Wertzuweisung, C 435  
Wertzuweisung, Operator 440  
What You See Is What You Get → *WYSIWYG*  
WHERE-Klausel, SQL 677  
while()-Schleife 446  
while-Befehl in Shell-Skripten 391  
White Book 1044  
White Book (Video-CD) 148  
Whitespace  
    *in C-Programmen* 435  
    *in RegExp* 532  
Wide Area Network 182  
Widerspruch, logischer 93  
Widgets 582  
Wikipedia 936  
Wikis 936  
Wildcards  
    *in UNIX-Dateinamen* 370  
    *Windows-Dateinamen* 321  
Win32 1044  
    *Umgang mit 16-Bit-Anwendungen* 307  
Win32 API 288, 1044  
WindowListener, AWT-Klasse 596  
Window-Manager 403  
    *CDE* 403  
    *fwm2* 403  
Windows 279, 305  
    *16-Bit-Versionen* 305  
    2000 280, 306  
    2000 Server 306  
    95 279, 305  
    98 280, 305  
    *Active Directory* 347  
    *als Server einrichten* 346  
    *Anzeige-Konfiguration* 335  
    *Apache-Installation* 720  
    *attrib-Befehl* 322  
    *auf Netzwerkrechner zugreifen* 347  
    *Ausgabeumleitung* 321  
    *Batch-Datei* 323  
    *Benutzer-An- und Abmeldung* 316  
    *Benutzerverwaltung* 335  
    *cd-Befehl* 322  
    *copy-Befehl* 322  
    *CreateProcess(), Systemaufruf* 292  
    *CygWin* 431  
    *Dateiattribute* 302  
    *Dateiattribute ändern* 322  
    *Dateien kopieren* 317  
    *Dateien löschen, Konsole* 322  
    *Dateien umbenennen* 322  
    *Dateien verschieben* 317, 322  
    *Dateinamen* 302  
    *Dateinamen-Platzhalter* 321  
    *Dateisysteme* 300, 309  
    *Datenträgerverwaltung* 338  
    *del-Befehl* 322  
    *Desktop-Konfiguration* 335  
    *Dienste* 338  
    *dir-Befehl* 322  
    *Drucker freigeben* 347  
    *durch UNIX-Server bedienen* 412  
    *Eingabeaufforderung* 322  
    *Eingabeumleitung* 321  
    *Entwicklung* 305  
    *Ereignisanzeige* 337  
    *Explorer* 316  
    *Fenster* 312  
    *Freigabeart* 346  
    *Gruppenrichtlinienobjekte* 338

- Windows (Forts.)  
*Hardware-Konfiguration* 335, 337  
*help-Befehl* 288  
*herunterfahren* 316  
*Home-Verzeichnis* 302  
*IP-Adresse zuweisen* 342  
*Konsole* 319  
*Konsole starten* 321  
*Konsolenbefehle* 322  
*Laufwerksbuchstaben* 300  
*MDI-Anwendung* 313  
*Me* 280, 305  
*Microsoft Management Console (MMC)* 338  
*mkdir-Befehl* 322  
*MMC (Microsoft Management Console)* 338  
*move-Befehl* 322  
*MySQL-Installation* 669  
*Netzwerkdrucker nutzen* 347  
*Netzwerkkonfiguration* 341  
*Netzwerktreiber* 341  
*Netzwerkverbindungen einrichten* 341  
*neu starten* 316  
*NT* 279, 306  
*NT Server* 306  
*NT-Familie* 306  
*Oberfläche* 310  
*Ordner freigeben* 346  
*Ordner-Ansichten* 318  
*path, Umgebungsvariable* 321  
*Pfadangaben* 301  
*Pipes* 321  
*Prompt* 322  
*Prozessmodell* 292  
*regedit, Dienstprogramm* 339  
*Registry* 339  
*rename-Befehl* 322  
*rmdir-Befehl* 322  
*SDI-Anwendungen* 313  
*Server 2003* 306  
*Serverpakete* 349  
*Serversysteme* 347  
*Server-Versionen* 306  
*Share Level Security* 346  
*Software installieren* 335  
*Startmenü/Startmenü, Windows* 314  
*Systemkonfiguration* 334
- Windows (Forts.)  
*Systemsteuerung* 334  
*Systray* 316  
*Taskleiste* 314  
*Textdatei anzeigen, Konsole* 323  
*type-Befehl* 323  
*Umgebungsvariablen setzen* 321  
*User Level Security* 346  
*Versionsübersicht* 306  
*Verzeichnis anlegen* 322  
*Verzeichnis löschen* 322  
*Verzeichnis wechseln* 322  
*Verzeichnisinhalt, Konsole* 322  
*Win32 API* 288
- Windows 2000 280, 306  
*Server* 306  
*Shell* 319
- Windows 2000 Server 347
- Windows 95 279, 305  
*Oberfläche* 311
- Windows 98 280, 305  
*Oberfläche* 311
- Windows Me 280, 305
- Windows NT 279, 306  
*Server* 306  
*Shell* 319
- Windows PowerShell → PowerShell
- Windows Server 2003 306, 347  
*als DHCP-Server* 348  
*als DNS-Server* 348  
*als Webserver* 348  
*Server-Rollen* 348  
*Varianten* 348
- Windows Vista 280
- Windows XP 280, 306  
*Oberfläche* 311  
*Shell* 319
- Windows-Eingabeaufforderung 319  
*Eingabevollständigung* 319  
*History* 319  
*QuickEdit-Modus* 320  
*Tastenkürzel* 319
- WinGate, Backdoor 1008
- Winkelgeschwindigkeit, konstante 146
- WinTel-PC 42
- Wireless LAN 198–199  
*Access Point* 201  
*Basic Service Set (BSS)* 201
- Wireless LAN (Forts.)  
*CSMA/CA-Verfahren* 201  
*Extended Service Set (ESS)* 202  
*Frequency Hopping* 200  
*Frequenzbereiche* 200  
*Hardware* 201  
*Sicherheit* 202  
*Übertragungstechnik* 200
- Wirth, Niklaus 48
- Wirtschaftsinformatik 32
- WLAN → Wireless LAN
- Wohlgeformtheit 1044
- Wohlgeformtheit, XML-Dokumente 779
- Word, 16 Bit 79
- WordPress, Blog-Tool 936
- World Wide Web 263  
*Geschichte* 170  
*HTTP-Protokoll* 263  
*Komponenten* 170  
*URL* 263
- Worst Case (Komplexität) 93
- Wortbreite 40, 78, 1044  
*der CPU* 115  
*verschiedener CPUs* 116
- Wörter zählen (Textdateien) 377
- Wozniak, Steve 41, 278
- Write-Host, PowerShell-CmdLet 328
- WSDL 1044
- WSDL, Web Services 580
- Würmer 1004  
*Schutzmaßnahmen* 1005
- WYSIWYG 41, 1044

---

**X**

- X Window 289, 401, 1044  
*Konfiguration* 402  
*SaX, openSUSE-X-Konfigurationsprogramm* 402  
*starten* 402  
*Xconfigurator, Fedora-Konfigurationsprogramm* 402
- X11R7 402
- x86-Assembler, Beispiele 121
- Xalan, XSLT-Prozessor 797
- XAMPP 703
- XConfigurator (X-Server-Konfigurationsprogramm) 402
- Xerces, XML-Parser 800

## Index

- XEROX PARC 41  
Xerox PARC 278  
XFree86 402  
XHTML 840, 1044  
    *Besonderheiten* 843  
XML 769, 1044  
    *Ajax-Datenaustausch durch*  
        988, 990–991  
    *Attributdeklaration (DTD)* 788  
    *Attribute* 771, 775  
    *Attribute, falsche Verwendung*  
        776  
    *Attribute, Verwendung* 776  
    *Beispieldokument* 771  
    *CDATA-Blöcke* 777–778  
    *CSS verwenden* 796  
    *Dateiendung* 771  
    *Datenbank* 655  
    *DocBook* 770  
    *DOCTYPE-Angabe* 781  
    *Dokument parsen, SAX* 801  
    *Dokumenteingabe* 770  
    *Dokument-Fragmente* 780  
    *Dokumentstruktur* 771  
    *DTD* 781  
    *DTD definieren* 782  
    *DTD, Attribute deklarieren* 788  
    *DTD, Elemente deklarieren* 784  
    *DTD, Entities deklarieren* 791  
    *EJB-Deployment-Deskriptor* 576  
    *Elementdeklaration (DTD)* 784  
    *Elemente (Tags)* 773  
    *Entitäten deklarieren (DTD)* 791  
    *Entity-Referenzen* 777, 791  
    *Entity-Referenzen vermeiden*  
        778  
    *Entity-Referenzen, numerisch*  
        777  
    *Hierarchie* 773  
    *in Datenbank schreiben, Beispiel-*  
        *programm* 807  
    *leere Tags* 775  
    *leere Tags, Kurzfassung* 775  
    *mehrere Namensräume* 792  
    *MIME-Types* 771  
    *Namensräume* 792  
    *Namensräume, mehrere im Doku-*  
        *ment* 792  
    *Parser* 799  
    *Parser, Xerces* 800  
    *PCDATA* 777  
XML (Forts.)  
    *Processing Instructions (PI)* 773  
    *Programmierung* 799  
    *Programmierung, SAX* 800  
    *PUBLIC-ID* 781  
    *SAX* 800  
    *Schema* 793  
    *Schema-Beispiel* 793  
    *SGML-Erbe* 769  
    *Sonderzeichen* 776  
    *spezielle Editoren* 770  
    *Standalone-Dokument* 773  
    *Standard-Namensraum* 792  
    *Steueranweisungen* 773  
    *SVG* 770  
    *SYSTEM-ID* 781  
    *Tag-Namen* 773  
    *Tags* 771  
    *Tag-Verschachtelung* 773  
    *Universalität* 770  
    *Verschachtelte Tags* 773  
    *verschachtelte Tags* 780  
    *Verschachtelungsfehler* 780  
    *Web Services* 579  
    *wichtige Dokumentformate* 770  
    *Wohlgeformtheit* 779  
    *Wurzelement* 779  
    *Xerces, Parser* 800  
    *XHTML* 770  
    *xmlns-Angabe* 793  
    *xml-Steueranweisung* 772  
    *XPath* 1045  
    *XSL* 796  
    *XSL-FO* 796  
    *XSLT* 796  
    *Zeichensätze* 773  
XML Schema 793, 1045  
    *Attribute deklarieren* 795  
    *Elemente deklarieren* 794  
    *Verschachtelte Elemente* 794  
XML-Datenbank 655  
XML-Editoren 770  
XMLHttpRequest 982  
XMLReader, Java-Interface 800  
XOR-Schaltung 86  
XOR-Verknüpfung 64  
XP → Extreme Programming  
XPath 1045  
X-Server 289, 401  
    *Konfiguration* 402  
    *über das Netzwerk betreiben* 252  
XSL 796  
XSL Formatting Objects 796  
XSL-FO 796, 1045  
XSLT 796, 1045  
    *Prozessor* 797  
    *Wichtige Elemente* 799  
    *Wurzelement* 799  
    *Xalan, Prozessor* 797  
    *xsl:for-each, Tag* 799  
    *xsl:template, Tag* 799  
    *xsl:value-of, Tag* 799  
XSLT-Prozessor 797  
XSS → Cross-Site-Scripting  

---

**Y**

---

YAML, Dateiformat 922  
YEAR, SQL-Datentyp 675  
Yellow Book (CD-ROM) 148  

---

**Z**

---

Z3, erster funktionierender Com-  
puter 36  
Z80, Prozessor 116  
Zahlenmengen  
    *Ganze Zahlen* 68  
    *Imaginäre Zahlen* 69  
    *Komplexe Zahlen* 69  
    *Natürliche Zahlen* 68  
    *Rationale Zahlen* 69  
    *Reelle Zahlen* 69  
Zahlensysteme 72  
    *Dezimalsystem* 73  
    *Dualsystem* 73  
    *Hexadezimalsystem* 74  
    *Oktalsystem* 74  
    *Römische Zahlen* 73  
    *Schreibweise* 77  
    *Umrechnung* 75  
Zeichen  
    *alternative in RegExp* 531  
    *aus Strings lesen, Java* 463  
    *ausschließen in RegExp* 531  
    *beliebig viele, in RegExp* 531  
    *Darstellung, C* 437  
    *eines oder mehr, in RegExp* 531  
    *Eingabe, C* 453  
    *genaue Anzahl, in RegExp* 532  
    *Gruppen in RegExp* 531  
    *optionale, in RegExp* 531

- Zeichen (Forts.)
  - Wortbestandteile in RegExp* 532
- Zeichengeräte (char devices) 286
- Zeichenkette → String
- Zeichen-Literale 438
- Zeichensatz, in HTML angeben
  - 843
- Zeichensätze 55, 814
  - ANSI 816
  - ASCII 55
  - Codepage 817
  - Eingabe chinesischer Zeichen* 821
  - Unicode 55, 818
- Zeichenvorrat
  - der Turing-Maschine* 96
- Zeiger 449, 1045
  - auf beliebigen Datentyp* 526
- Zeiger (Forts.)
  - auf nichts (NULL)* 455
  - fehlende in Java* 463
  - für Call by Reference* 449
- Zeilenfrequenz (Monitor) 158
- Zeilenumbruch
  - auf verschiedenen Plattformen* 812
  - HTML 843
  - konvertieren* 813
  - Perl-Programme* 814
- Zeilenumbrüche entfernen, Perl 473
- zeitdiskret 54
- Zeitmanagement 621
  - Netzplan* 622
- Zentraleinheit 111
  - alte Bedeutung* 112
- ZEROFILL, SQL-Feldoption 676
- Ziffern in RegExp 532
- z-index, CSS-Angabe 889
- zip, UNIX-Befehl 836
- Zip-Dateien 833
- Zip-Laufwerk 147
- Zugriffsrechte 300
  - ändern, UNIX* 373
- Zuordnungseinheiten (Dateisystem) 296
- Zuse, Konrad 36
- Zustände, der Turing-Maschine 96
- Zweierkomplement 80
- ZX Spectrum 278
- ZX81 42, 278

