
Inhaltsverzeichnis

1	Grundlagen	24
1.1	Einführung	24
1.2	Generationen der Programmiersprachen	24
1.2.1	Erste Generation	25
1.2.2	Zweite Generation	25
1.2.3	Dritte Generation	25
1.2.4	Vierte Generation	26
1.2.5	Fünfte Generation	27
1.3	Klassifizierung von Programmiersprachen	27
1.3.1	Programmierparadigmen	27
1.3.2	Abstraktionsgrade	29
1.3.3	Ausführungsschemata von Programmiersprachen	29
1.4	Gebräuchliche Programmiersprachen	30
1.5	Programmierung	34
1.5.1	Prozedurale Programmierung	34
1.5.2	Objektorientierte Programmierung	35
1.5.3	Funktionale Programmierung	37
1.5.4	Logische Programmierung	37
1.6	Glossar Programmiersprachen	38
2	C	62
2.1	Übersicht	62
2.2	Organisation der Sprache	63
2.2.1	Modularisierung des Quelltextes	63
2.2.2	Namen und Literale	63
2.2.3	Präprozessor	64
2.2.4	Genereller Quelltextaufbau	68
2.3	Imperative Aspekte	68
2.3.1	Primitive Datentypen	68
2.3.2	Eigene und zusammengesetzte Datentypen	69
2.3.3	Speicherbereiche für Daten	74
2.3.4	Ausdrücke	76
2.3.5	Operatoren und ihre Reihenfolgen	76
2.3.6	Anweisungen	77
2.4	Funktionen	79
2.4.1	Header- und Implementierungsdatei	79
2.4.2	Variable Parameteranzahl	80
2.4.3	Programmstart	81
2.5	Zeiger	81
2.5.1	Zeiger auf Strukturen und Unions	82
2.5.2	Zeiger, Arrays und Zeigerarithmetik	83
2.5.3	Zeiger auf Funktionen	83
2.5.4	Dynamische Speicherverwaltung	84

2.6	Programmierrichtlinien	84
2.6.1	Namensschemata	84
2.6.2	Kommentierung	84
2.7	Einbettung der Sprache	84
2.7.1	Strings	85
2.7.2	Arrays und Speicher	85
2.7.3	Mathematik	86
3	C++	88
3.1	Übersicht	88
3.2	Organisation der Sprache	88
3.2.1	Modularisierung des Quelltextes	89
3.2.2	Kommentare, Namen und Literale	90
3.2.3	Präprozessor	90
3.2.4	Genereller Quelltextaufbau	91
3.3	Imperative Aspekte	92
3.3.1	Primitive Datentypen	92
3.3.2	Eigene und zusammengesetzte Datentypen	92
3.3.3	Speicherbereiche für Daten	92
3.3.4	Ausdrücke	92
3.3.5	Operatoren und ihre Reihenfolgen	92
3.3.6	Anweisungen	95
3.4	Funktionen	95
3.4.1	Inline-Funktionen	96
3.4.2	Generische Funktionen	97
3.4.3	Programmstart	97
3.5	Zeiger und Referenzen	97
3.5.1	Zeiger auf Objekte	97
3.5.2	Zeiger auf Methoden	98
3.5.3	Referenzen	98
3.5.4	Dynamische Speicherverwaltung	99
3.6	Objekte und Klassen	101
3.6.1	Deklaration und Aufbau	101
3.6.2	Konstruktor und Initialisierungsreihenfolge	101
3.6.3	Destruktor und Freigabereihenfolge	104
3.6.4	Attribute	104
3.6.5	Methoden	105
3.6.6	Zugriffsrechte	107
3.7	Geschachtelte Klassen	108
3.8	Generische Klassen	108
3.9	Vererbung	109
3.9.1	Konstruktoren und Initialisierungsreihenfolge	110
3.9.2	Destruktoren und Freigabereihenfolge	111
3.9.3	Überschreiben von Methoden	111
3.9.4	Abstrakte Klassen	112
3.9.5	Mehrfachvererbung	113
3.10	Überladen von Operatoren	114

3.10.1	Elementoperatoren	114
3.10.2	Nichtelementoperatoren	116
3.10.3	Vordefinierte Operatoren	116
3.11	Fehlerbehandlung	117
3.11.1	Zusicherungen (Assertions)	117
3.11.2	Ausnahmen (Exceptions)	118
3.12	Einbettung der Sprache	119
3.12.1	Strings	119
3.12.2	Arrays	119
3.12.3	Mathematik	120
4	C#	121
4.1	Übersicht	121
4.1.1	Laufzeitumgebung CLR	121
4.1.2	Compiler	121
4.2	Organisation der Sprache	122
4.2.1	Modularisierung	122
4.2.2	Die Einsprungstelle (Main)	123
4.2.3	Kommentare, Namen und Literale	124
4.2.4	Compiler-Direktiven	125
4.3	Value-Types	126
4.3.1	Vordefinierte Value-Types	126
4.3.2	Benutzerdefinierte Value-Types	128
4.4	Reference-Types	130
4.4.1	Vordefinierte Reference-Types	130
4.4.2	Benutzerdefinierte Reference-Types	131
4.5	Klassen, Interfaces und Delegates	131
4.5.1	Klassen	131
4.5.2	Interfaces	134
4.5.3	Delegates (System.Delegate)	136
4.5.4	Generische Typen	136
4.6	Arrays (System.Array)	138
4.6.1	Eindimensionale Arrays	138
4.6.2	Mehrdimensionale Arrays	139
4.7	Pointer	139
4.8	Typ-Konvertierungen	140
4.9	Imperative Aspekte	141
4.9.1	Variablen und Parameter	141
4.9.2	Ausdrücke (Expressions)	142
4.9.3	Operatoren und ihre Reihenfolge	142
4.9.4	Anweisungen (Statements)	143
4.10	Fehlerbehandlung und Exceptions	145
4.11	Multithreading	146
4.11.1	Domänen (System.AppDomain)	146
4.11.2	Threads (System.Threading.Thread)	147
4.11.3	Synchronisation von Threads	148

4.12	Attribute	149
4.12.1	Deklaration von Attributen	149
4.12.2	Vordefinierte Attribute	150
4.13	Einbettung der Sprache	150
4.13.1	Datenstrukturen	150
4.13.2	Ein- und Ausgabe	151
4.13.3	Stream, TextReader und StreamWriter	151
5	Java	153
5.1	Übersicht	153
5.2	Organisation der Sprache	153
5.2.1	Modularisierung des Quelltextes durch Pakete	154
5.2.2	Namen und Literale	155
5.2.3	Genereller Quelltextaufbau	156
5.3	Imperative Aspekte	156
5.3.1	Primitive Datentypen	156
5.3.2	Eigene und zusammengesetzte Datentypen	157
5.3.3	Speicherbereiche für Daten	159
5.3.4	Ausdrücke	159
5.3.5	Operatorn und ihre Reihenfolgen	160
5.3.6	Anweisungen	162
5.4	Referenzen	163
5.4.1	Starke Referenzen	163
5.4.2	Schwache Referenzen	164
5.4.3	Dynamische Speicherverwaltung	164
5.5	Objekte und Klassen	165
5.5.1	Deklaration und Aufbau	165
5.5.2	Klassenrechte	165
5.5.3	Konstruktor und Initialisierungsreihenfolge	165
5.5.4	Destruktor (finalize) und Freigabereihenfolge	166
5.5.5	Attribute	167
5.5.6	Methoden	167
5.5.7	Attribut- und Methodenrechte	170
5.6	Geschachtelte Klassen	171
5.6.1	Nicht statische lokale Klassen	171
5.6.2	Anonyme Klassen	172
5.6.3	Statische lokale Klassen	172
5.7	Generische Klassen und Methoden	173
5.7.1	Deklaration generischer Klassen	173
5.7.2	Einsatz generischer Klassen	174
5.7.3	Wildcards	174
5.7.4	Generische Methoden	174
5.8	Vererbung	174
5.8.1	Konstruktor und Initialisierungsreihenfolge	175
5.8.2	Destruktor (finalize) und Freigabereihenfolge	175
5.8.3	Überschreiben von Methoden	176
5.8.4	Abstrakte Klassen	177
5.8.5	Schnittstellen	177

5.9	Fehlerbehandlung	178
5.9.1	Zusicherungen (Assertions)	178
5.9.2	Ausnahmen (Exceptions)	179
5.10	Weitere wichtige Spracheigenschaften	181
5.10.1	Metadaten (Annotationen)	181
5.10.2	Autoboxing	182
5.11	Programmierrichtlinien	182
5.11.1	Namensschemata	182
5.11.2	Kommentierung	182
5.12	Einbettung der Sprache	183
6	Visual Basic.NET	185
6.1	Übersicht	185
6.2	Organisation der Sprache	186
6.2.1	Modularisierung des Quelltextes	186
6.2.2	Kommentare, Name und Operatoren	188
6.3	Imperative Aspekte	191
6.3.1	Einfache Datentypen	191
6.3.2	Aufzählungstypen	192
6.3.3	Konstanten, Variablen und Arrays	192
6.3.4	Eigene und zusammengesetzte Datentypen	194
6.3.5	Speicherbereiche für Daten	196
6.3.6	Ausdrücke	196
6.3.7	Operatoren und ihre Reihenfolge	196
6.3.8	Anweisungen	197
6.4	Referenzen	199
6.4.1	Delegates	199
6.4.2	Dynamische Speicherverwaltung	199
6.5	Klassen, Module, Strukturen und Namensräume	200
6.5.1	Klassen	200
6.5.2	Deklaration und Aufbau	200
6.5.3	Module	202
6.5.4	Erzeugung und Löschung	203
6.5.5	Methoden	204
6.5.6	Überladen von Methoden	210
6.5.7	Zugriffsrechte	211
6.5.8	ME-Schlüsselwort	211
6.6	Fehler- und Ausnahmebehandlung	212
6.6.1	Strukturierte Ausnahmebehandlung	212
6.6.2	Unstrukturierte Ausnahmebehandlung	214
6.7	Vererbung	214
6.7.1	Überschreiben von Methoden	215
6.7.2	Abstrakte Klassen	215
6.7.3	Schnittstellen	216
6.8	Überladen von Operatoren	217
6.9	Vordefinierte oder wichtige Klassen	218

7	Delphi	219
7.1	Übersicht	219
7.2	Organisation der Sprache	219
7.2.1	Programmstart	220
7.2.2	Modularisierung des Quelltextes	220
7.2.3	Namen und Literale	221
7.2.4	Genereller Quelltextaufbau	221
7.3	Imperative Aspekte	222
7.3.1	Primitive Datentypen	222
7.3.2	Eigene und zusammengesetzte Datentypen	224
7.3.3	Arrays	226
7.3.4	Speicherbereiche für Daten	227
7.3.5	Ausdrücke	227
7.3.6	Operatoren und ihre Reihenfolgen	227
7.3.7	Anweisungen	229
7.4	Zeiger und Referenzen	230
7.4.1	Deklaration und Funktionsweise	230
7.4.2	Zeiger auf Funktionen und Prozeduren	231
7.5	Objekte und Klassen	232
7.5.1	Deklaration und Aufbau	232
7.5.2	Konstruktor	232
7.5.3	Destruktor	233
7.5.4	Attribute und Eigenschaften	234
7.5.5	Methoden	235
7.5.6	Zugriffsrechte	239
7.5.7	Helferklassen	239
7.6	Vererbung	240
7.6.1	Konstruktor	241
7.6.2	Destruktor	241
7.6.3	Verbergen von Methoden	241
7.6.4	Überschreiben von Methoden	242
7.6.5	Abstrakte Klassen	243
7.6.6	Schnittstellen	243
7.6.7	Überladen von Operatoren	244
7.7	Fehlerbehandlung	245
7.7.1	Zusicherungen (Assertions)	245
7.7.2	Ausnahmen (Exceptions)	245
7.8	Metadaten und Reflexion	246
7.9	Programmierrichtlinien	247
7.9.1	Namensschemata	247
7.9.2	Kommentierung	247
7.10	Einbettung der Sprache	247
7.10.1	Strings	247
7.10.2	Mathematik	248
8	ABAP	249
8.1	Übersicht	249

8.1.1	SAP NetWeaver Application Server ABAP.....	249
8.2	Organisation der Sprache	251
8.2.1	Quelltext und Programmgenerierung	251
8.2.2	ABAP-Syntax	251
8.2.3	Programmstruktur	252
8.2.4	Programmtypen und Programmausführung	254
8.3	Imperative Aspekte	256
8.3.1	Datentypen und Datenobjekte	256
8.3.2	Deklarationen	257
8.3.3	Zuweisungen	258
8.3.4	Ausdrücke	259
8.3.5	Programmablaufsteuerung	260
8.3.6	Ausnahmebehandlung	261
8.3.7	Byte- und Zeichenkettenverarbeitung	262
8.3.8	Interne Tabellen	263
8.3.9	Zeiger	264
8.3.10	Dynamische Programmierung	265
8.4	Ereignissteuerung	266
8.5	Prozedurale Programmierung	267
8.5.1	Unterprogramme	267
8.5.2	Funktionsbausteine	268
8.6	Objektorientierte Programmierung	269
8.6.1	Klassen	269
8.6.2	Interfaces	270
8.6.3	Objekte	270
8.7	Benutzerschnittstellen	271
8.7.1	Dynpros	271
8.7.2	Selektionsbilder	272
8.7.3	Listen	272
8.8	Zugriff auf persistente Daten	272
8.8.1	Datenbankzugriffe	272
8.8.2	Daten-Cluster	274
8.8.3	Dateischnittstelle	274
8.9	Weitere Sprachelemente	274
8.9.1	Funktionale Schnittstelle	274
8.9.2	Externe Datenschnittstelle	274
8.9.3	Testunterstützung	275
8.9.4	Erweiterungskonzepte	275
9	FORTRAN	276
9.1	Übersicht	276
9.2	Organisation der Sprache	276
9.2.1	Formate für Quelltext	276
9.2.2	Genereller Quelltextaufbau	277
9.2.3	Modularisierung des Quelltextes	278
9.3	Imperative Aspekte	279
9.3.1	Kommentare, Namen und Literale	279

9.3.2	Primitive Datentypen	280
9.3.3	Zusammengesetzte Datentypen	283
9.3.4	Arrays	285
9.3.5	Speicherorte für Daten	287
9.3.6	Operatoren	288
9.3.7	Anweisungen	290
9.4	Prozeduren	294
9.4.1	Parameterübergabe und Aufrufe	294
9.4.2	Vektorisierung, Parallelisierung und Rekursion	296
9.4.3	Module	298
9.5	Zeigervariablen, dynamischer Speicher	298
9.5.1	POINTER und TARGET	298
9.5.2	Dynamischer Speicher	299
9.6	Objektorientierte Aspekte	299
9.6.1	Abstrakte Datentypen, Variablen und Prozeduren	300
9.6.2	Operatordefinitionen und Überladung	300
9.7	Input und Output	302
9.7.1	Listengesteuerte Ein- und Ausgabe	302
9.7.2	Formatierte Ein- und Ausgabe	302
9.7.3	FORMAT-Anweisung	303
9.7.4	Ein- und Ausgabe auf Dateien	304
9.8	Intrinsische Funktionen	304
9.8.1	Mathematische Funktionen	304
9.8.2	String-Funktionen	306
9.8.3	Array-Funktionen	306
9.8.4	POINTER-Funktionen	307
9.9	Software und Literatur	308
10	Smalltalk	310
10.1	Übersicht	310
10.2	Organisation der Sprache	311
10.2.1	Namen und Literale	311
10.2.2	Objekte, Nachrichten und Methoden	312
10.3	Imperative Aspekte	314
10.3.1	Datentypen zur Zahlendarstellung	314
10.3.2	Ausdrücke	315
10.3.3	Variablen	316
10.3.4	Kontrollstrukturen	318
10.4	Objekte und Klassen	319
10.4.1	Erzeugung neuer Klassen	319
10.4.2	Erzeugen neuer Methoden	320
10.4.3	Vererbung	322
10.4.4	Überschreiben von Methoden	323
10.4.5	Objektidentität	323
10.4.6	Objektkopien	325
10.4.7	Abstrakte Klassen	325
10.5	Objektbehälter	325

10.5.1	Ungeordnete Behälter	326
10.5.2	Geordnete Behälter	328
10.5.3	Behälter-Iteratoren	331
10.6	Fehlerbehandlung	332
11	Ada	334
11.1	Einleitung	334
11.2	Aspekte zur Programmentwicklung	334
11.3	Algorithmen und Datenstrukturen	335
11.3.1	Typen in Ada	336
11.3.2	Anweisungen	340
11.3.3	Unterprogramme	341
11.3.4	Ausnahmebehandlung	342
11.4	Modulkonzept	343
11.4.1	Pakete	343
11.4.2	Generische Pakete	345
11.4.3	Vordefinierte Pakete	346
11.5	Objektorientierung	347
11.6	Echtzeitfähigkeit und Multitasking	348
11.7	Hochzuverlässige Echtzeitsysteme	349
11.8	Einbettung der Sprache, Umgebung	350
12	Eiffel	352
12.1	Übersicht	352
12.2	Organisation der Sprache	354
12.2.1	Genereller Quelltextaufbau	354
12.2.2	Namen und Literale	355
12.3	Imperative Aspekte	356
12.3.1	Primitive Datentypen	356
12.3.2	Arrays	357
12.3.3	Operatoren und ihre Reihenfolgen	358
12.3.4	Anweisungen	359
12.4	Objekte und Klassen	360
12.4.1	Deklaration und Aufbau	361
12.4.2	Klassenrechte	362
12.4.3	Objekterzeugung	363
12.4.4	Attribute	364
12.4.5	Routinen	365
12.4.6	Generische Klassen	367
12.4.7	Abstrakte Klassen	368
12.4.8	Expandierte Klassen	369
12.4.9	Veralteute Klassen	370
12.5	Vererbung	370
12.5.1	Ableiten einer Klasse	371
12.5.2	Fortgeschrittene Vererbungskonzepte	374

12.6	Fehlerbehandlung	376
12.6.1	Zusicherungen	376
12.6.2	Aktivierung der Fehlerbehandlung	379
12.6.3	Ausnahmen (Exceptions)	380
12.7	Die Eiffel-Standardbibliothek	381
13	JavaScript	382
13.1	Übersicht	382
13.2	Organisation der Sprache	382
13.2.1	Interpretationsschema	383
13.2.2	Einbettung in HTML-Code	383
13.3	Imperative Aspekte	384
13.3.1	Kommentare, Namen und Literale	384
13.3.2	Eingebaute Datentypen	385
13.3.3	Ausdrücke und Operatoren	386
13.3.4	Anweisungen	386
13.3.5	Ausnahmeverarbeitung	388
13.4	Prozedurale und OO-Aspekte	389
13.4.1	Funktionen	389
13.4.2	Objekte	389
13.4.3	Spezielle prozedurale Aspekte	391
13.5	Eingebaute Objekte	391
13.5.1	String-Objekt	392
13.5.2	Math-Objekt	393
13.5.3	Array-Objekt	394
13.5.4	Date-Objekt	395
13.5.5	RegExp-Objekt	395
13.6	Browser-API	396
13.6.1	DOM HTML	397
13.6.2	Navigator-Objekt	398
13.6.3	Window-Objekt	399
13.6.4	Document-Objekt	402
13.6.5	Event-Objekt und Event-Handling	404
13.6.6	Software	405
14	Perl	407
14.1	Übersicht	407
14.2	Organisation der Sprache	407
14.2.1	Quelltext und Aufruf	407
14.2.2	Modularisierung des Quelltextes	409
14.2.3	Ablauf eines Programms	409
14.2.4	Kommentare, Namen und Literale	410
14.3	Imperative Aspekte	411
14.3.1	Skalare, Arrays und Zeiger	411
14.3.2	Zuweisungen	413
14.3.3	Operatoren	414
14.3.4	Reguläre Ausdrücke	415

14.3.5	Anweisungen	416
14.3.6	Iteratoren für Arrays	417
14.3.7	Input/Output in Perl	418
14.4	Prozedurale Aspekte	420
14.4.1	Unterprogramme	420
14.4.2	Packages	421
14.5	Objektorientierte Aspekte	422
14.5.1	Klassendefinition	422
14.5.2	Vererbung	423
14.6	Eingebaute Funktionen und Objekte	423
14.6.1	Kontroll- und Dateifunktionen	423
14.6.2	Mathematische und String-Funktionen	425
14.6.3	Funktionen für Listen und Hashes	426
14.7	Perl/Tk	426
14.7.1	Programmstruktur	426
14.7.2	Widgets und Optionen	427
14.7.3	Widgets und Methoden	430
14.7.4	Geometriemanagement in Tk	433
14.7.5	Ereignissteuerung	434
14.8	Software und Literatur	434
14.8.1	Open Source und freie Software	434
14.8.2	Perl-Hilfesystem	435
15	PHP	436
15.1	Übersicht	436
15.2	Organisation der Sprache	436
15.2.1	Quellcode und Programmstruktur	436
15.2.2	Konfiguration von PHP über die php.ini-Datei	437
15.2.3	Kommentare, Namen, Literale	439
15.2.4	Datentypen	440
15.2.5	Referenzen	440
15.2.6	Vordefinierte Variablen und Konstanten	441
15.3	Imperative Aspekte	442
15.3.1	Operatoren	442
15.3.2	Anweisungen	442
15.4	Prozedurale Elemente und Funktionen	445
15.4.1	Funktionen	445
15.4.2	Parameterübergabe	445
15.5	Objektorientierte Aspekte in PHP 5	446
15.5.1	Deklaration und Aufbau	446
15.5.2	Konstruktor	446
15.5.3	Destruktor	447
15.5.4	Attribute	447
15.5.5	Methoden	448
15.5.6	Zugriffsrechte	450
15.5.7	Vererbung	450
15.5.8	Abstrakte Klassen	450

15.5.9	Interfaces	451
15.6	Objektorientierte Aspekte in PHP 4	452
15.7	Fehlerbehandlung in PHP 5	452
15.8	Wichtige Funktionen	453
15.8.1	String-Funktionen	453
15.8.2	Funktionen für Arrays	455
15.8.3	Mathematische Funktionen	456
15.8.4	Kontrollfunktionen	456
15.8.5	Zeit- und Datumsfunktionen	457
15.8.6	Dateifunktionen	457
15.8.7	Klassen- und Objekt-Funktionen	458
15.8.8	Open Source und freie Software	458
16	Python	460
16.1	Übersicht	460
16.2	Organisation der Sprache	460
16.2.1	Interpreter	461
16.2.2	Modularisierung	461
16.2.3	Namen und Literale	462
16.2.4	Genereller Quelltextaufbau	465
16.2.5	Kommentare und Dokumentation	465
16.3	Imperative Aspekte	466
16.3.1	Standarddatentypen	466
16.3.2	Operatoren und ihre Reihenfolgen	468
16.3.3	Variablen	470
16.3.4	Objekte/Typen	471
16.3.5	Anweisungen	471
16.3.6	Zusammengesetzte Anweisungen	472
16.4	Funktionen	473
16.4.1	Parameter	474
16.4.2	Aufruf und Rückgabe	475
16.4.3	Generator-Funktionen	475
16.4.4	Anonyme Funktionen	476
16.5	Speicherverwaltung	476
16.6	Objekte und Klassen	477
16.6.1	Klasseninstanzen	477
16.6.2	Methoden	477
16.6.3	Vererbung	478
16.6.4	Datenkapselung	478
16.6.5	Überladen von Operatoren	479
16.7	Fehlerbehandlung	479
16.7.1	Benutzerdefinierte Ausnahmen	480
16.7.2	finally-Anweisung	480
16.8	Multithreading	480
16.8.1	Threads erzeugen und starten	481
16.8.2	Threads beenden und löschen	481
16.8.3	Synchronisation von Threads	482

16.8.4	Kommunikation zwischen Threads	482
16.9	Weitere wichtige Spracheigenschaften	483
16.9.1	Decorators	483
16.9.2	Grafische Benutzeroberflächen	483
16.9.3	Software	484
17	Ruby	486
17.1	Übersicht	486
17.2	Organisation der Sprache	487
17.2.1	Code und Programmstruktur	487
17.2.2	Quellcode-Dateien	487
17.2.3	Module und Namensräume	488
17.2.4	Mixins	488
17.2.5	Interpreter	489
17.2.6	Bezeichner und Namen	489
17.2.7	Literale	491
17.3	Imperative Aspekte	493
17.3.1	Operatoren	493
17.3.2	Variablen	495
17.3.3	Objekte/Typen	495
17.3.4	Standarddatentypen	496
17.3.5	Kontrollstrukturen	497
17.4	Methoden	500
17.4.1	Methoden und Blöcke	500
17.5	Klassen und Objekte	501
17.5.1	Objekte und Attribute	501
17.5.2	Vererbung	502
17.5.3	Mehrfachvererbung und Mixins	503
17.5.4	Virtuelle Attribute	503
17.6	Fehlerbehandlung	504
17.6.1	Exceptions	504
17.6.2	Catch und Throw	505
17.7	Weitere wichtige Spracheigenschaften	506
17.7.1	eRuby und CGI-Einsatz	506
17.7.2	Grafische Benutzeroberflächen	506
17.8	Programmierrichtlinien	506
17.8.1	Kommentare	506
17.8.2	Dokumentations-Strings	507
18	Tcl/Tk	508
18.1	Übersicht	508
18.2	Organisation der Sprache	508
18.2.1	Quelltext und Aufruf	508
18.2.2	Modularisierung des Quelltextes	509
18.2.3	Kommentare, Namen und Literale	509
18.3	Imperative Aspekte	511

18.3.1	Datentypen	511
18.3.2	Anweisungen, Ausdrücke und Operatoren	512
18.3.3	Reguläre Ausdrücke	513
18.3.4	Anweisungen	514
18.3.5	Input/Output in Tcl	516
18.4	Prozedurale Aspekte	519
18.4.1	Prozeduren	519
18.4.2	Parameterübergabe	519
18.5	Eingebaute Funktionen und Objekte	520
18.5.1	Kontroll- und Dateifunktionen	520
18.5.2	Mathematische Funktionen	521
18.5.3	Anweisungen für Strings	522
18.5.4	Anweisungen für Listen	522
18.6	Tcl/Tk	524
18.6.1	Programmstruktur und Widgets	524
18.6.2	Geometriemanagement in Tk	525
18.7	Software	526
19	Maple	527
19.1	Grundlagen	527
19.1.1	Programmbedienung	527
19.1.2	Hilfefunktion	528
19.1.3	Modularisierung durch Pakete	528
19.2	Einfache Berechnungen	529
19.2.1	Mathematische Formeln	529
19.3	Funktionen	531
19.3.1	Eigene Funktionen	531
19.3.2	Grafische Darstellung von Funktionen	532
19.3.3	Gleichungen	533
19.4	Programmieren in Maple	534
19.4.1	Maple und Programmiersprachen	534
19.4.2	Programmstrukturen in Maple	535
19.4.3	Programmieren eines Maple-Pakets	536
19.4.4	Zeichenketten	537
19.4.5	Bereiche, Listen und Mengen	537
19.4.6	Felder und Matrizen	538
19.4.7	Tabellen	540
19.5	Ausgabe und Speicherung	540
20	Mathematica	542
20.1	Grundlagen	542
20.1.1	Modularisierung durch Packages	543
20.1.2	Online-Hilfe	544
20.2	Einfache Berechnungen	544
20.2.1	Operatoren	544
20.2.2	Standardfunktionen	545

20.3	Variablen und Gleichungen	546
20.3.1	Testfunktionen	547
20.3.2	Substitution	547
20.3.3	Gleichungen	547
20.3.4	Numerische Daten	548
20.3.5	Differenzialrechnung	548
20.4	Grafik und Sound in Mathematica	549
20.4.1	Funktionsplots	549
20.4.2	Grafikobjekte in Mathematica	551
20.4.3	Sound-Objekte in Mathematica	553
20.5	Programmieren mit Mathematica	553
20.5.1	Kontrollstrukturen	553
20.5.2	Zeichenketten	554
20.5.3	Listen und Arrays	555
20.5.4	Vektoren und Matrizen	556
20.5.5	Eigene Funktionen	557
20.6	Ausgabe und Speicherung	558
20.6.1	Ausgabeformate	559
20.6.2	Export	559
20.7	Software und Literatur	559
21	MATLAB	561
21.1	Grundlagen	561
21.2	Einfache Berechnungen	561
21.2.1	Variablen und Typen	562
21.2.2	Funktionen	563
21.2.3	Speicherung von Sitzungen	563
21.2.4	Skriptdateien	564
21.2.5	Steuerungsbefehle	564
21.3	Grafische Darstellung von Funktionen	565
21.3.1	Einfache Plots	565
21.3.2	Weitere Grafikbefehle	566
21.3.3	Zeichnen von Oberflächen	567
21.4	Vektoren und Vektoroperationen	567
21.4.1	Einfache Operationen	567
21.4.2	Operationen zwischen Vektoren	569
21.5	Matrizen	570
21.5.1	Einfache Operationen	570
21.5.2	Operationen zwischen Matrizen	570
21.6	Vergleiche und Kontrollstrukturen	571
21.6.1	Logische Operatoren	571
21.6.2	Schleifen	572
21.6.3	if-Anweisungen	572
21.6.4	Weitere eingebaute Funktionen	573
21.6.5	Eigene MATLAB-Funktionen	573
21.6.6	Zeitnahme	574
21.7	Software und Literatur	574

22 LISP	575
22.1 Übersicht	575
22.1.1 Entwicklung von LISP	575
22.1.2 Einsatz von LISP	575
22.2 Aufbau der Sprache	575
22.2.1 LISP-Ausdrücke	576
22.2.2 Funktionsaufrufe	576
22.2.3 Spezielle Ausdrücke	577
22.2.4 Makros	577
22.3 Datentypen und Variablen	578
22.3.1 Zahlen	578
22.3.2 Symbole und Variablen	578
22.3.3 Listen	579
22.3.4 Arrays und Vektoren	580
22.3.5 Characters und Strings	581
22.3.6 Functions	581
22.4 Funktionen und Prädikate	582
22.4.1 Definition benannter Funktionen	582
22.4.2 Prädikate	583
22.4.3 Funktionen als Variablen	583
22.4.4 Lambda-Ausdrücke	584
22.5 Programmsteuerung	584
22.5.1 Einfache Programmsequenz	584
22.5.2 Neue Variablenumgebung	585
22.5.3 Bedingungen	585
22.5.4 Iteration	586
22.5.5 Mapping	587
22.5.6 Rekursion	587
22.6 Vordefinierte Funktionen	588
22.6.1 Mathematische Funktionen	588
22.6.2 Funktionen für Characters und Strings	589
22.6.3 Funktionen für Sequenzen	590
22.6.4 Funktionen für Listen	591
22.7 Objektorientierte Programmierung	593
22.7.1 Strukturen	593
22.7.2 CLOS: Objekte und Klassen	594
23 PROLOG	597
23.1 Übersicht	597
23.1.1 Entwicklung von PROLOG	597
23.1.2 Einsatz der Sprache	597
23.1.3 Aufbau der Sprache	598
23.2 Syntax	598
23.2.1 Allgemeine Syntax	599
23.2.2 Konstanten	599
23.2.3 Variablen	599
23.2.4 Strukturen	600
23.2.5 Operatoren	600

23.3	Programmstruktur	601
23.3.1	Fakten	601
23.3.2	Fragen	601
23.3.3	Matching	602
23.3.4	Regeln	603
23.4	Datenstrukturen und Arithmetik	604
23.4.1	Arithmetik	605
23.4.2	Darstellung von Listen	606
23.4.3	Arbeiten mit Listen	606
23.5	Backtracking und Cut	608
23.5.1	Zielerfüllung und Backtracking	608
23.5.2	Der Cut-Befehl	610
Sachwortverzeichnis		613