

Inhalt

Vorwort	xi
1: Referenzen auf Daten und unbenannte Speicherbereiche .	1
Bereits existierende Variablen referenzieren	3
Referenzen verwenden	9
Verschachtelte Datenstrukturen	14
Eine Referenz abfragen	16
Symbolische Referenzen	17
Ein Blick in die Interna	18
Referenzen in anderen Sprachen	22
Ressourcen	24
2: Komplexe Datenstrukturen	25
Benutzerdefinierte Strukturen	26
Beispiel: Matrizen	27
Professoren, Studenten und Veranstaltungen	31
Reichen Sie mir den Umschlag!	35
Pretty-Printing	37
Ressourcen	40
3: Typeglobs und Symboltabellen	41
Perl-Variablen, Symboltabelle und Gültigkeitsbereich	42
Typeglobs	45
Typeglobs und Referenzen	50
Dateihandles, Verzeichnishandles und Formate	51

4: Referenzen auf Subroutinen und Closures	55
Referenzen auf Subroutinen	56
Referenzen auf Subroutinen verwenden	58
Closures	61
Closures benutzen	63
Vergleiche mit anderen Sprachen	69
Ressourcen	70
5: Eval	71
Der String-Modus: Auswertung von Ausdrücken	72
Der Blockmodus: Ausnahmebehandlung	74
Passen Sie auf Ihre Anführungszeichen auf	77
Eval zur Auswertung von Ausdrücken verwenden	78
Eval aus Effizienzgründen benutzen	80
Eval für Timeouts benutzen	86
Eval in anderen Sprachen	87
Ressourcen	89
6: Module	91
Das Schlüsselwort package	92
Packages und Dateien	94
Initialisierung und Zerstörung von Packages	95
Zugriffsrechte	96
Symbole importieren	98
Packages verschachteln	100
Autoloading	101
Zugriff auf die Symboltabelle	102
Sprachvergleiche	104
7: Objektorientierte Programmierung	107
Objektorientierung: Eine Einführung	107
Objekte in Perl	109
Wiederholung der Konventionen	125
Vergleiche mit anderen objektorientierten Sprachen	129
Ressourcen	131

8: Objektorientierung: Die nächsten Schritte	133
Effiziente Speicherung von Attributen	133
Delegation	144
Vererbung	145
Ressourcen	148
9: Tie	149
Skalare binden	150
Arrays binden	153
Hashes binden	156
Dateihandles binden	157
Beispiel: Variablen überwachen	158
Vergleiche mit anderen Sprachen	163
10: Persistenz	165
Fragestellungen der Persistenz	166
Datenströme	168
Datensatzorientierte Ansätze	171
Relationale Datenbanken	173
Ressourcen	180
11: Objektpersistenz implementieren	181
Adaptor: Eine Einführung	183
Anmerkungen zum Design	186
Implementierung	193
Ressourcen	201
12: Netzwerkprogrammierung mit Sockets	203
Einführung in Netzwerke	204
Socket-API und IO::Socket	205
Mehrere Clients bedienen	208
Server für die rauhe Wirklichkeit	213
IO-Objekte und Dateihandles	214
Vorgefertigte Client-Module	215
Ressourcen	217

13: Netzwerkprogrammierung: RPC-Implementierung	219
Msg: ein Toolkit für Nachrichten	220
Remote Procedure Calls (RPC)	232
Ressourcen	238
14: Benutzerschnittstellen mit Tk	239
Einführung in GUIs, Tk und Perl/Tk	240
Der Einstieg in Perl/Tk	241
Ein Rundgang durch die Widgets	244
Geometriemanagement	262
Timer	265
Bindungen von Ereignissen	265
Ereignisschleifen	268
Ressourcen	269
15: GUI-Beispiel: Tetris	271
Einführung in Tetris	271
Entwurf	272
Implementierung	274
16: GUI-Beispiel: Anzeigeprogramm für Man-Pages	281
man und perlman	282
Implementierung	283
Ressourcen	290
17: Mustergesteuerte Codegenerierung	291
Einführung in die Codegenerierung	291
Jeeves-Beispiel	294
Überblick über Jeeves	299
Die Implementierung von Jeeves	302
Beispiel eines Spezifikations-Parsers	309
Ressourcen	311
18: Perl erweitern: Eine Einführung	313
Eine Erweiterung programmieren: Überblick	314
Beispiel: Fraktale mit Perl	317
Eigenschaften von SWIG	321

XS-Merkmale	324
Freiheitsgrade	328
Ein Abstecher zu den Fraktalen	328
Ressourcen	332
19: Perl einbetten auf die leichte Tour	333
Warum Einbetten?	333
Das Einbetten im Überblick	335
Beispiele	338
Erweiterungen hinzufügen	341
Ressourcen	342
20: Perl-Interna	343
Die Quellen lesen	344
Architektur	345
Die Wertetypen von Perl	353
Stacks und das Datenübertragungsprotokoll	377
Raffinierte Erweiterungen	383
API zum leichten Einbetten von Perl	393
Ein Blick in die Zukunft	396
Ressourcen	397
A: Tk Widget-Referenz	399
B: Syntax-Zusammenfassung	415
Index	423