

Pascal

**Anleitung zur
systematischen Programmierung
und Konstruktion
zuverlässiger Programme
mit Anwendungsbeispielen
in Standard- und Turbo-Pascal
Band I: Einführung**

von

Prof. Dr.-Ing. Fevzi Belli

Hochschule Bremerhaven
unter Mitarbeit von

Dipl.-Ing. Hansjörg Troebner
SOFTECH, Bremerhaven

Wissenschaftsverlag
Mannheim/Wien/Zürich

Inhaltsverzeichnis

and I: EINFÜHRUNG

	Vorwort	5
	Eingetragene Warenzeichen, besondere Produkte und Bezugsquellen für Pascal- Übersetzer	11
	Verwendete Notation und Konventionen ...	12
	Bemerkungen zu Beispielprogrammen	13
	Abkürzungen	14
1.	Einleitung	20
1.1	Historischer Überblick	20
1.2	Einige Einsatzgebiete elektronischer Rechensysteme	23
1.3	Sprachen als Kommunikationsmittel zwischen verschiedenen Dialogpartnern	26
1.4	Programmiersprachen	27
1.5	Erstellung und Ausführung von Programmen	32
1.6	Hauptkomponenten elementarer Rechner . . .	41
1.7	Pascal und andere Programmiersprachen . .	44
1.8	Anleitung zur Bearbeitung der Aufgaben . .	49
1.9	Aufgaben	50
2.	Algorithmus und Programm	53
2.1	Was ist ein Algorithmus?	53
2.2	Hauptkomponenten der Algorithmen: Kontrollstrukturen und Datenstrukturen . .	65
2.3	Grafische Darstellung der Algorithmen: Ablaufdiagramme und Struktogramme	68
2.4	Automatisch ausführbares Programm	76
2.5	Aufgaben	78
3.	Syntax, Semantik und Pragmatik	82
3.1	Einführung in Syntaxdiagramme	88
3.2	Backus-Naur-Form	91
3.3	Aufgaben	94
4.	Einstieg in Pascal: einfache Sprachkon- strukte und allgemeiner Programmaufbau . .	96
4.1	Bezeichner	97
4.2	Bedingung (Vergleich), Zusammengesetzte Bedingung ("Filter"); Bedingte Anweisung	101
4.3	Variablendeklaration	107
4.4	Datentypen	109
4.5	Zusammengesetzte Anweisung	111
4.6	Schreibanweisung (Ausgabe)	113
4.7	Leseanweisung (Eingabe)	118
4.8	Zuweisung	120
4.9	Allgemeine Darstellung der Anweisungen . .	121
4.10	Mein "erstes "algorithmisches" Programm"	123
4.11	Aufgaben	127

5.	Die engste Programmierumgebung : Vom Editor zum Linker - oder: Wie bringe ich meinem Programm "das Laufen" bei?	130
5.1	Erstellen des Quellprogramms	132
5.2	Grundfunktionen eines Editors dargestellt am Beispiel Wordstar	134
5.3	Die Programmier-Umgebung bei Turbo-Pascal	143
5.4	Erstellung des Objektcodes	146
5.5	Ausführen des Programms	146
5.6	Hilfe, mein Programm hat "bugs" ("Wanzen")! Typische Schritte einer Debugging- ("Entwanzungs-") Prozedur	147
5.7	Allgemeine Anleitung zur Bearbeitung der Programmieraufgaben	149
5.8	Aufgaben	150
6.	Einfache skalare Datentypen	152
6.1	Dezimale, duale, oktale, hexadezimale Zahlendarstellung und der Zahlenring . . .	152
6.2	Ganze Zahlen (INTEGER)	161
6.3	Datentypen BYTE, WORD, LONGINT	165
6.4	Reelle Zahlen (REAL)	166
6.5	Weitere Operationen mit numerischen Daten	170
6.6	Zeichen (CHARACTER)	172
6.7	Ordnungs-Operationen auf skalaren Datentypen	174
6.8	Konstanten (CONST)	175
6.9	Aufgaben	180
7.	Schleifen (Wiederholungen, Iterationen, "loops")	188
7.1	Abweisende Schleife (WHILE - Anweisung)	191
7.2	Nichtabweisende Schleife (REPEAT-Anweisung)	192
7.3	Zählschleife (FOR-Anweisung)	194
7.4	Auswahlkriterien für verschiedene Schleifenformen	197
7.5	Aufgaben	198
8.	Tücken der Schleifen und anderer Sprachkonstrukte	203
8.1	Verwendung der Semikolons	203
8.2	Fehler bei Schleifenkonstrukten	204
8.3	Aufgaben	209
9.	Logische Größen	217
9.1	Skalarer Standardtyp BOOLEAN	217
9.2	Operationen auf BOOLEAN	218
9.3	Ausgabe von BOOLEAN	221
9.4	Aufgaben	223

10.	Weitere Konstrukte zur Realisierung von Daten- und Kontrollstrukturen	229
10.1	Arithmetische und boolesche Ausdrücke; Operationen auf Ausdrücken	229
10.2	Allgemeine Fallunterscheidung (CASE) (Mehrfachauswahl/Schalter/Switch)	236
10.3	Aufgaben	241
11.	Kleines Zwischenresümee	243
11.1	Allgemeine Betrachtung der Kontrollstrukturen	244
11.2	Allgemeine Betrachtung der Datenstrukturen	245
11.3	Kontrollstrukturen in Pascal	246
11.4	Datenstrukturen in Pascal	247
11.5	Aufgaben	248
12.	Einfache selbstdefinierte Datentypen (TYPE)	249
12.1	Aufzählungstyp	251
12.2	Ausschnittstyp/Unterbereichsdattentyp	256
12.3	Zusammenfassung einfacher selbstdefinierter Datentypen und Reihenfolge der Deklarationen	260
12.4	Aufgaben	262
13.	Strukturierte Datentypen	266
13.1	Mengentypen (SET)	266
13.2	Felder/Reihungen (ARRAY)	274
13.3	Mehrdimensionale Felder	282
13.4	Komprimierte Darstellung von Feldern (PACKED-Attribut)	284
13.5	Zeichenkette (STRING)	286
13.6	Verbund/Struktur (RECORD)	294
13.7	WITH-Anweisung	298
13.8	Einige Bemerkungen zum Programmier-Stil	304
13.9	Aufgaben	310
A1.	Programmbeispiele "zum Lesen"	320
A2.	Mit Vorsicht genießen: GOTO und RECORD mit CASE	330
A3.	Übersicht über Struktur und Elemente eines Pascal-Programms	335
A4.	Strukturiertes Verzeichnis der Pascal-Konstrukte	336
A5.	Besondere Zeichen, Reservierte Wörter und Standard-Bezeichner (Standard- und Turbo-Pascal)	338

A6.	Standard-Funktionen und Prozeduren (Standard- und Turbo-Pascal)	341
A7.	Hinweise zur Rechnerbenutzung und MS-DOS- Befehle (Auszug)	343
A8.	WordStar-Befehle (Auszug)	347
A9.	Turbo-Pascal-Umgebung, Compiler-Anweisungen (Auszug)	350
A10.	ASCII-Code	354
All.	Pascal-Syntax in Syntax-Diagrammen (Standard und Turbo-Pascal)	356
A12.	Lösungen der Übungsaufgaben	371
A13.	Literaturhinweise	392
	Stichwortverzeichnis	398

Pascal

Anleitung zur
systematischen Programmierung
und Konstruktion
zuverlässiger Programme
mit Anwendungsbeispielen
in Standard- und Turbo-Pascal

Band II: Weitere Methoden

von

Prof. Dr.-Ing. Fevzi Belli

Universität Gesamthochschule
Paderborn

unter Mitarbeit von

Dipl.-Ing. Hansjörg Troebner

High Soft Tech, Bremerhaven

Wissenschaftsverlag

Mannheim/Wien/Zürich

Inhaltsverzeichnis

BAND I: EINFÜHRUNG (Übersicht)

1. Einleitung
2. Algorithmus und Programm
3. Syntax, Semantik und Pragmatik
4. Einstieg in Pascal: einfache Sprachkonstrukte und allgemeiner Programmaufbau
5. Die engste Programm!erumgebung :
Vom Editor zum Linker - oder:
Wie bringe ich meinem Programm
"das Laufen" bei?
6. Einfache skalare Datentypen
7. Schleifen (Wiederholungen, Iterationen,
"Ioop8")
8. Tücken der Schleifen und anderer
Sprachkonstrukte
9. Logische Größen
10. Weitere Konstrukte zur Realisierung von
Daten- und Kontroll Strukturen
11. Kleines Zwischenresumee
12. Einfache selbstdefinierte Datentypen
(TYPE)
13. Strukturierte Datentypen

Band II: WEITERE METHODEN

Vorwort	5
Eingetragene Warenzeichen, besondere Produkte und Bezugsquellen für Pascal- übersetzer	12
Verwendete Notation und Konventionen	13
Bemerkungen zu Beispielprogrammen	14
Abkürzungen	15

14.	Dateien und Datei Verarbeitung.21
14.1	Dateien in Pascal.23
14.2	GrundoperatIonen auf Datelen.26
14.3	Weitere Datei-Operationen.34
14.4	Text-Dateien.42
14.5	Standard-Dateien.50
14.6	Aufgaben.52
15.	Unterprogramm-Techniken.58
15.1	Grundbegriffe.60
15.2	Prozeduren.62
15.3	Parameter-Übergabe.71
15.4	Werteparameter ("call by value").76
15.5	Variablenparameter ("call by reference").79
15.6	Funktionen.	- 87
15.7	Standardprozeduren und -Funktionen ("built-in subroutines").93
15.8	Funktions- und Prozedurparameter.93
15.9	Aufgaben.95
16.	Rekursion.106
16.1	Funktionsweise und Arten der Rekursionen.107
16.2	Realisierung und klassische Beispiele110
16.3	Beispiele zu nicht-linearer Rekursion116
16.4	Rekursion oder Iteration?.121
16.5	Aufgaben.123
17.	Dynamische Datenstrukturen.129
17.1	Einführendes zum Umgang mit dynamischen Variablen.133
17.2	Belegung und Freigabe des dynamischen Speicherplatzes.139
17.3	Weitere Operationen mit dynamischen Variablen.151
17.4	Lineare Listen.154
17.5	Nicht-lineare Listen.168
17.6	Baumstrukturen.172
17.7	Aufgaben.177

18.	Nebenläufige Prozesse und Echtzeit-Eigenschaften183
18.1	Eigenschaften von Prozessen186
18.2	Darstellung von Prozessen in Programmiersprachen.189
18.3	Nebenläufige Prozesse, Ihre Koordination und Synchronisierung	193
18.4	Prozeß-Kommunikation198
18.5	Semaphor-Konzept.200
18.6	Darstellung nebenläufiger Prozesse	206
18.7	Aufgaben.	214
19.	Modularer Programmaufbau217
19.1	Was zeichnet einen Modul aus?221
19.2	Programm-Aufbau und Programm-Einheiten bei Turbo-Pascal (Units).234
19.3	System-Bibliothek mit Standard-Units und ihre Erweiterung.234
19.4	Mehrfach deklarierte und qualifizierte Bezeichner.239
19.5	Vergleich der Modul-Konzepte in Turbo-Pascal und Modula-2.243
19.6	Modulare Programmierung in Turbo-Pascal 5.0.246
19.7	Aufgaben247

Anhang

A1.	Programmentwicklungs-Umgebung des Turbo-Pascal-Systems 5.0 und Unterschiede zwischen Versionen Turbo-Pascal 4.0 und 5.0.250
A2.	Überblick über UNIX.259
A3	Lösungen der Aufgaben.261
A4.	Literaturhinweise.274

Stichwortverzeichnis.280
-------------------------------	------

Pascal

**Anleitung zur
systematischen Programmierung
und Konstruktion
zuverlässiger Programme
mit Anwendungsbeispielen
in Standard- und Turbo-Pascal**

Band III: Programmierpraxis

von

Prof. Dr.-Ing. Fevzi Belli

Universität - Gesamthochschule
Paderborn

unter Mitarbeit von

Dipl.-Ing. Hansjörg Troebner

High Soft Tech, Bremerhaven

Band III: PROGRAMMIERPRAXIS

	Vorwort.	5
	Eingetragene Warenzeichen, besondere Produkte und Bezugsquellen für Pascal-Übersetzer.	11
	Verwendete Notationen und Konventionen	12
	Bemerkungen zu Beispiel Programmen	13
	Abkllrzungsverzeichnis.	14
	Inhaltsverzeichnis.	15
20.	Systemnahe Elemente der Programmierumgebung, dargestellt am Turbo-Pascal-System	20
20.1	Unterprogramm-Reihenfolge ("FORWARD")	21
20.2	Übersetzer-Anweisungen (Compiler-Options).	25
20.3	Überlagerungs-Technik (OVERLAY-Technik)	36
20.4	Systemnahe Speicher-Operationen	46
20.5	Speicher-Zugriff ("MEM") und Ansprechen der Rechner-Schnittstelle ("PORT")	54
20.6	Einbindung von Maschinen-Code ("INLINE") bzw. Objekt-Code.	57
20.7	Aufruf von MSDOS-Dienstprogrammen ("INTR")	62
20.8	Einige Änderungen in Turbo-Pascal 5.0	65
20.9	Aufgaben.	71
21.	"Pretty printing" und andere "Kochrezepte" für die Selbst-Dokumentation der Programme	75
21.1	Gleich zu Beginn: Programm-Kopf und Kommentare.	75
21.2	Allgemeiner Programm-Aufbau	76
21.3	"Nest level" der Kontrollstrukturen	78
21.4	Gestaltung der Bezeichner.	79
21.5	Ein Beispiel.	80
21.6	Aufgaben.	92
22.	Prinzipien und Hilfsmittel des Software-Engineering für die Programm-Konstruktion	94
22.1	Prinzip der Abstraktion.	97
22.2	Geheimnis-Prinzip.	109
22.3	Prinzip der Modularisierung.	111
22.4	Prinzip der hierarchischen Strukturierung	118
22.5	Prinzip der Lokalität.	124
22.6	Prinzip der integrierten Dokumentation	125
22.7	Prinzip der Standardisierung.	127
22.8	Prinzip der Mehrfachverwendung.	128
22.9	Aufgaben.	129
23.	Grundzüge der Programm-Verifikation: Formale und empirische Methoden.	134
23.1	Software und ihre "Mängel".	136
23.2	Prüfen von Software-Produkten.	139

23.3	Formale Korrektheits-Beweise.141
23.4	Strukturiertes Programmieren mit Invarianten.153
23.5	Symbolische Ausführung ("symbolic execution").155
23.6	Weitere statische Prüfverfahren.160
23.7	Generierende Prüfverfahren: Programm-Tests	161
23.8	Das LCSAJ-Testverfahren.164
23.9	Der Quelltext-Debugger von Turbo-Pascal 5.0.170
23.10	Aufgaben.174
24.	Elemente der Software-Fehlertoleranz und -Zuverlässigkeit für die Programm- Konstruktion.178
24.1	Fehlertoleranz und Verlässlichkeit178
24.2	Software-Fehlertoleranz.185
24.3	Vorwärts-Fehlerbehebung ("exception handling") und ihre Realisierung in Programmiersprachen.190
24.4	Software-Zuverlässigkeit.197
24.5	Modellierung der Software-Zuverlässigkeit	201
24.6	Ein Beispiel für Makro-Modelle: Musa-Modell.209
24.7	Ein Beispiel für Mikro-Modelle: Shooman-Modell und einige Bemerkungen . .	211
24.8	Aufgaben.214
25.	"Fehlermanagement" als in die Entwicklung integrierte Fehlerbehandlung.217
25.1	"Software-Fehler", diesmal bezogen auf den Software-Entwicklungs-Prozeß.218
25.2	Konventionelle Qualitätssicherung	222
25.3	Elemente der Software-Qualitätssicherung .	225
25.4	Ein Versuch zur systematischen Behandlung der Software-Qualitätsmerkmale.232
25.5	Methoden und Hilfsmittel des Software- Engineering zur Realisierung der Funk- tionen der Software-Qualitätssicherung .	235
25.6	Weitere Elemente der Software-Qualitäts- sicherung.243
25.7	Aufgaben.244
26.	Einblick in die Programm!er-Praxis: Exemplarische Erstellung eines "größeren" Programms.247
26.1	Aufgabenstellung.247
26.2	Entwurf.253
26.3	Aufbau des Editor-Fensters.259
26.4	Editor-Operationen	263
26.5	Implementation des Editors.266
26.6	Typ-Modul für String-Deklarationen	270

26.7	Datenobjekt-Moduln für "Editor" und "Schirm"!.270
26.8	Modul "Edit".278
26.9	Nutzen des Moduls "Edlt".299
27.	Ausblick und Schlußbemerkungen300
27.1	Objekt-orientierte Programmierung als Erweiterung der Datenabstraktion und abstrakter Datentypen302
27.2	Objekt-orientierte Programmierung mit Pascal?.304
27.3	Aufgaben309
 Anhang		
A1.	Lösungen der Übungsaufgaben310
A2.	Literaturhinweise.314
Stichwortverzeichnis.338