

Inhalt

Vorwort	1
1 Algorithmen	
1.1 Beispiele für Algorithmen	3
1.2 Definition des Begriffs "Algorithmus"	7
1.3 Objekte und Aktionen	11
1.4 Darstellungsarten von Algorithmen	19
1.5 Einige einfache Algorithmen	26
1.6 Algorithmen und Programme	32
<i>Übungsaufgaben</i>	34
2 Prinzipien der Programmentwicklung	
2.1 Das Prinzip der schrittweisen Verfeinerung	37
2.2 Datenkapselung	45
2.3 Der Begriff des Moduls	50
<i>Übungsaufgaben</i>	52
3 Die Programmiersprache Modula-2	
3.1 Notation für die Sprachbeschreibung	55
3.2 Lexikalische Elemente von Modula-2	57
3.2.1 Bezeichner	57
3.2.2 Schlüsselwörter	59
3.2.3 Zahlen	59
3.2.4 Zeichenketten	62
3.2.5 Operatoren und Begrenzer	64
3.2.6 Trennung von Symbolen	65
3.2.7 Kommentare	65
<i>Übungsaufgaben</i>	66
3.3 Elementare Programmstruktur	67
3.4 Deklarationen	70
3.4.1 Datentypen	71
3.4.1.1 Vordeklarierte Datentypen	71
3.4.1.2 Einfache Datentypen	76
3.4.1.3 Strukturierte Datentypen	78
3.4.2 Konstantendeklarationen	85
3.4.3 Typdeklarationen	89
3.4.4 Variablen-deklarationen	90
3.4.5 Typgleichheit von Objekten	91
<i>Übungsaufgaben</i>	92
3.5 Ausdrücke	93
3.5.1 Klassen von Ausdrücken	93
3.5.2 Operanden	93
3.5.3 Operatoren	95

3.5.4	Regeln zum Schreiben von Ausdrücken	96
3.5.5	Ausdruckskompatibilität von Operanden	100
<i>Übungsaufgaben</i>		101
3.6	Anweisungen	102
3.6.1	Wertzuweisungen	104
3.6.2	Verzweigungen	105
3.6.2.1	IF-Anweisung	105
3.6.2.2	CASE-Anweisung	107
<i>Übungsaufgaben</i>		110
3.6.3	Schleifen	111
3.6.3.1	WHILE-Anweisung	111
3.6.3.2	REPEAT-Anweisung	113
3.6.3.3	FOR-Anweisung	115
3.6.3.4	LOOP- und EXIT-Anweisung	117
<i>Übungsaufgaben</i>		119
3.6.4	WITH-Anweisung	120
<i>Übungsaufgaben</i>		121
3.7	Prozeduren	122
3.7.1	Deklaration von Prozeduren	123
3.7.2	Aufruf von Prozeduren	126
3.7.3	RETURN-Anweisung	128
3.7.4	Funktionsprozeduren	130
3.7.5	Gültigkeitsbereiche und Lebensdauer von Objekten in Prozeduren	132
3.7.6	Datenaustausch zwischen Prozeduren und rufenden Programmeinheiten	135
3.7.7	ARRAY-Parameter	138
3.7.8	Standardprozeduren	139
3.7.9	PROCEDURE-Typen und Prozedurvariablen	144
<i>Übungsaufgaben</i>		148
3.8	Das Modulkonzept	149
3.8.1	Innere Modulen	149
3.8.2	Gültigkeitsbereiche und Lebensdauer von Objekten in Modulen	153
3.8.3	Äußere Modulen und getrennte Übersetzung	155
3.8.3.1	Definitionsmodulen	156
3.8.3.2	Implementierungsmodulen	158
<i>Übungsaufgaben</i>		161
3.9	Systemabhängige Spracheigenschaften	161
3.9.1	Der Modul SYSTEM	161
3.9.2	Typtransferfunktionen	166
3.9.3	Absolute Adressierung von Variablen	168
3.10	Prozesse und Co-Routinen	169
3.10.1	Erzeugung und Synchronisation von Prozessen	171
3.10.2	Unterbrechungen und Prioritäten	176
4	Das Schreiben von Modula-2-Programmen	
4.1	Einige einfache Modula-2-Programme	180
4.2	Ein/Ausgabe	186

4.2.1	Der Modul Terminal	188
4.2.2	Die Moduln InOut und RealInOut	189
4.2.3	Der Modul FileSystem	193
4.3	Dynamische Datenstrukturen	197
4.3.1	Das Arbeiten mit dynamischen Datenstrukturen	198
4.3.2	Dynamisch erzeugte RECORDs mit Varianten	204
4.4	Rekursion	205
4.4.1	Rekursive Prozeduren	205
4.4.2	Innerer Aufbau und Ablauf von rekursiven Prozeduren	208
4.4.3	Einige Beispiele für rekursive Algorithmen	209
4.4.4	Vor- und Nachteile der rekursiven Programmierung	211
4.5	Moduln zur Spracherweiterung	212
4.5.1	Der Modul MathLib0	212
4.5.2	Der Modul String	213
4.5.3	Der Modul Storage	215
4.6	Parallele Prozesse	215
	<i>Übungsaufgaben</i>	225
5	Programmierstil und Programmtest	
5.1	Programmierstil	227
5.1.1	Strukturierung	227
5.1.2	Namenwahl	233
5.1.3	Kommentierung	234
5.1.4	Äußere Form von Programmen	235
5.2	Testen von Programmen	236
5.2.1	Fehlerarten	236
5.2.2	Vorgangsweise beim Testen	236
5.2.3	Im Programm eingebaute Testhilfen	238
5.2.4	Testwerkzeuge	240
6	Software-Engineering mit Modula-2	
6.1	Modula-2 als Hilfsmittel zur Spezifikation	241
6.2	Modularisierung und Strukturierung	242
6.3	Datenkapselung und Datenabstraktion	243
6.4	Getrennte Übersetzung und Typenbindung	247
Anhang A:	Syntaxdiagramme	251
Anhang B:	Vordeklarierte Bezeichner	269
Anhang C:	Kompatibilitätsregeln	270
Anhang D:	ASCII-Tabelle	274
Literatur		275
Stichwortverzeichnis		276