

Inhaltsverzeichnis

1. Grundlagen	11
1.1 Algorithmus und Berechenbarkeit	11
1.1.1 Algorithmus	11
1.1.2 Turing-Maschine	13
1.1.3 Berechenbarkeit	18
1.2 Sprache und Grammatik	21
1.2.1 Sprache	21
1.2.2 Grammatik	25
1.3 Rechner	31
1.3.1 Von-Neumann-Rechnerarchitektur	32
1.3.2 Rechnersysteme	38
1.4 Informatik als Wissenschaft	46
2. Imperative Programmierung – die Sprache Modula-2	53
2.1 Syntaxdarstellungen	53
2.2 Elementare funktionale Modula-2-Programme	57
2.2.1 Eine Modula-2-Teilsprache	57
2.2.2 Programmverzweigungen	69
2.2.3 Funktionen und Prozeduren	72
2.2.4 Elementare Datentypen, Aufzählungs- und Bereichstypen	74
2.2.5 Eingabevariablen	79
2.2.6 Rekursive Funktionen und Prozeduren	80
2.2.7 Nachteile funktional-rekursiver Programme	85
2.3 Iterative Programme	87
2.3.1 Wertzuweisungen und Referenzparameter	87
2.3.2 Gültigkeitsbereich und Lebensdauer	89
2.3.3 Anweisungen zur Iteration	96
2.3.4 Vergleich iterativer und rekursiver Lösungen	99
2.3.5 Sprunganweisungen	102
2.3.6 Prozedurtypen	104
2.4 Komplexe Datentypen	109
2.4.1 Mengen (Sets)	109
2.4.1.1 Darstellung und Manipulation von Mengen	110
2.4.1.2 Ein Beispiel für Sets	111
2.4.2 Arrays (Felder)	113
2.4.3 Records (Verbunde)	119
2.4.3.1 Einfache Records	120

2.4.3.2	Records mit Varianten	126
2.4.4	Zeiger (Pointer) und dynamische Variablen	129
2.4.4.1	Die Speicherung auf der Halde	130
2.4.4.2	Operationen auf Zeigern	132
2.4.4.3	Verkettete Listen	136
2.4.4.4	Anwendungen und Probleme dynamischer Variablen	145
2.4.5	Dateien (Files)	148
2.4.5.1	Eigenschaften und formale Beschreibung	148
2.4.5.2	Dateien in Pascal	150
2.4.5.3	Dateien in Modula-2	153
3.	Abstraktion	157
3.1	Abstraktionskonzepte in Programmiersprachen	157
3.2	Abstraktion in Modula-2	161
3.2.1	Das Prinzip der separaten Übersetzung	161
3.2.2	Modularisierung eines Programms	163
3.2.3	Datenkapselung	167
3.2.4	Abstrakte Datentypen	174
3.2.4.1	Das Prinzip des Abstrakten Datentyps	174
3.2.4.2	Abstrakte Datentypen Schlange und Stack	176
3.2.4.3	Abstrakter Datentyp für große Zahlen	183
3.2.4.4	Abstrakter Datentyp für komplexe Zahlen	188
4.	Semantik, Verifikation und Test	193
4.1	Konzepte für eine Semantikdefinition	194
4.1.1	Semantik: Begriff und Motivation	194
4.1.2	Grundprinzipien von Semantiknotationen	195
4.1.3	Ein Beispiel für die operationale Semantik	198
4.2	Spezifikation und Verifikation von Programmen	204
4.2.1	Vor- und Nachbedingungen	204
4.2.2	Schwächste Vorbedingungen	207
4.2.3	Die Verifikation	209
4.2.4	Beschreibung einer Schleife durch eine Invariante	215
4.2.5	Konstruktion iterativer Programme	217
4.2.6	Zusammenfassung	222
4.3	Test	224
4.3.1	Begriffsbildung und Prinzipien	224
4.3.1.1	Begriffliche Abgrenzung	224
4.3.1.2	Aufgabenteilung und Zielsetzung	225
4.3.1.3	Material und Resultate des Tests	226

4.3.2	Grenzen des Testens	227
4.3.3	Die Konstruktion von Testdaten	228
4.3.4	Zusammenfassung	237
5.	Programmierparadigmen und -sprachen	239
5.1	Programmierparadigmen	239
5.1.1	Imperatives Programmieren	243
5.1.2	Funktionales Programmieren	249
5.1.3	Logik-basiertes Programmieren	252
5.1.4	Objektorientiertes Programmieren	257
5.1.5	Regel-basiertes Programmieren	265
5.1.6	Programmierung von Mehrprozessor-Systemen	267
5.2	Übersicht über Programmiersprachen	269
6.	Datenstrukturen und Algorithmen	273
6.1	Komplexität und Effizienz	273
6.1.1	Motivation und Begriffsbildung	273
6.1.2	Effizienz und Komplexität von Algorithmen	274
6.1.3	Komplexität von Funktionen und Sprachen	279
6.2	Graphen und Bäume	283
6.2.1	Graphen	283
6.2.2	Bäume	293
6.3	Suchen in gegebenen Datenstrukturen	305
6.3.1	Suchen in Tabellen	305
6.3.2	Suchen von Zeichenketten	309
6.4	Datenorganisationen für effizientes Suchen	320
6.4.1	Suchverfahren auf Bäumen	320
6.4.1.1	Binäre Suchbäume	320
6.4.1.2	AVL-Bäume	328
6.4.1.3	Optimale Suchbäume	344
6.4.1.4	B-Bäume	349
6.4.1.5	Weitere balancierte Suchbäume	357
6.4.2	Hashing	360
6.4.2.1	Begriffsbildung und Anforderungen	360
6.4.2.2	Perfektes Hashing	362
6.4.2.3	Kollisionsbehandlung	363
6.4.2.4	Löschen in Hash-Tabellen	368
6.4.2.5	Aufwandsabschätzung	369
6.4.2.6	Implementierung von Kollisionsbehandlungen	371

6.5	Sortieren	377
6.5.1	Klassifizierung und allgemeine Betrachtungen	377
6.5.2	Interne Sortierverfahren	385
6.5.2.1	Einfache Sortierverfahren	385
6.5.2.2	Schnelle Sortierverfahren	389
6.5.2.3	Implementierung ausgewählter Sortierverfahren	399
6.5.2.4	Aufwandsvergleich der Sortierverfahren	404
6.5.2.5	Sortieren durch Streuen und Sammeln	405
6.5.3	Externe Sortierverfahren	409
6.5.3.1	Direktes Mischen	410
6.5.3.2	Natürliches Mischen	411
6.5.3.3	Mehrwege-Mischen	411
6.6	Speicherverwaltung	413
6.6.1	Algorithmische Konzepte	414
6.6.2	Implementierung von Stacks	422
Anhang A: Mathematische Grundbegriffe und Formeln		429
Anhang B: Syntaxdiagramme für Modula-2		432
Literatur		449
Abkürzungsverzeichnis		451
Modula-2-Index		452
Index		453