

Inhaltsverzeichnis

I Grundbegriffe des Programmierens	7
1 Eine Einführung in die Grundlagen von Pascal	9
1.1 Überblick	9
1.2 Beispiel: Die Lösung einer quadratischen Gleichung	10
1.2.1 Lösung ohne Verständnis	10
1.3 Ein informeller Algorithmus	11
1.3.1 Auflösung komplizierter Schritte	12
1.3.2 Unübersichtlichkeit durch Länge	13
1.3.3 Mehrdeutigkeiten im Algorithmus	13
1.3.4 Auflösung von Mehrdeutigkeiten	13
1.4 Ein erstes Pascalprogramm	14
1.4.1 Struktur eines Pascalprogramms	15
1.4.2 Deklaration von Konstanten und Variablen	17
1.5 Übersetzung der ersten Programme	19
1.6 Ausgabe von Zahlen	21
1.7 Rechnerzahlen sind mangelhaft	22
1.8 Die Auswahlanweisung	23
1.8.1 Abwandlungen der Auswahlanweisung	25
1.9 Testausdrücke	26
1.10 Parametereingabe	28
1.11 Schleifen	29
1.11.1 Die WHILE-Schleife	29
1.11.2 Die FOR-Schleife	31
1.12 Syntaxdiagramme	32
1.13 Zusammenfassung	37
2 Ein umfangreicheres Beispiel: Das Spiel des Lebens	39
2.1 Überblick	39
2.2 Die Spielidee	40
2.2.1 Lebewesen in einer Schachbrettwelt	40
2.2.2 Verschiedene Nöpelvölker	41
2.2.3 Ein Pflichtenheft zur Simulation der Nöpelwelt	43
2.2.4 Modularisierung von Algorithmen	45
2.2.5 Die Grobaufteilung der Aufgabe	45

2.2.6	Die Modulstruktur des Lebensspiel-Algorithmus	46
2.2.7	Der informelle Algorithmus	46
2.2.8	Die Einzelaufgaben	47
2.3	Ein Programm „Lebensspiel“	49
2.3.1	Die Darstellung der Nöpelwelt im Rechner mit einem Feld	49
2.3.2	Mehrdimensionale Felder	50
2.3.3	Prozeduren zur Lösung von Teilaufgaben	52
2.3.4	Sichtbarkeit von Namen	54
2.3.5	Wiederverwendung von Namen	55
2.3.6	Parameter für Unterprogramme	56
2.3.7	Das Hauptprogramm ohne Unterprogramme	57
2.3.8	Setzen der Anfangskonstellation	58
2.3.9	Einlesen der Laufparameter	59
2.3.10	Ausgabe der Nöpelwelt	60
2.3.11	Berechnung aller Generationen	61
2.3.12	Ausgabeparameter	62
2.3.13	Die Berechnung einer neuen Generation	63
2.3.14	Zählung der Nachbarn	64
2.3.15	Test auf Lebendigkeit	65
2.3.16	Rückschau auf das Programm	66
2.3.17	Probleme bei der Modularisierung von Algorithmen	66
2.4	Syntaxgraphen von Feldern und Prozeduren	68
2.5	Variationen der Aufgabenstellung	68
2.6	Eine sichere Eingabefunktion	69
2.6.1	Der Datentyp <code>char</code>	70
2.6.2	Typendeklarationen	70
2.6.3	Funktionen	71
2.6.4	Die Erkennung einer gültigen Zahl mit endlichen Automaten	72
2.6.5	Ein Programm mit Zustandsvariable	74
2.6.6	Die Berechnung des Zahlenwertes einer Ziffernfolge	75
2.7	Eine zufällige Anfangskonstellation	75
2.7.1	Ein Algorithmus zur Erzeugung einer Pseudozufallszahlenfolge	76
2.8	Eine Neuinterpretation der Zustände	78
2.8.1	Sortieren der Statistik	79
2.9	Rekursive Algorithmen	80
2.9.1	Rekursive Programme	82
2.9.2	Mergesort	83
2.10	Zusammenfassung	85
3	Hardware	87
3.1	Beschreibungsebenen der Hardware	87
3.2	Bits und Bytes	89
3.3	Funktionsgruppen im Rechner	91
3.4	Zusammenfassung	98

II Formale und ethische Grenzen	99
4 Die formale Spezifikation von Problemen	101
4.1 Unzulänglichkeit informeller Aufgabenstellungen	101
4.2 Die Aussagen- und Prädikatenlogik	103
4.2.1 Logische Aussagen	103
4.2.2 Logische Operatoren	105
4.2.3 Distributivität, Assoziativität, Kommutativität, Modus Ponens	107
4.2.4 Modus Ponens	109
4.2.5 Prädikatenlogik	110
4.2.6 Formale Spezifikation: Das Beispiel Sortieren	117
4.3 Grenzen der Spezifikation	118
4.3.1 Bedeutungsverlust durch Spezifikation	119
4.3.2 Spezifikation von umfangreichen Aufgaben	120
4.3.3 Wirkungen auf die Gesellschaft	120
4.4 Zusammenfassung	121
5 Validierung von Programmen	123
5.1 Überblick	123
5.2 Der Test von Programmen	123
5.2.1 Teststrategie	125
5.2.2 Modultest und Systemtest	126
5.2.3 White-Box-Tests und Black-Box-Tests	126
5.3 Zur Leistungsfähigkeit von Black-Box-Tests	127
5.3.1 Leistungen und Mängel von Tests	128
5.4 Die formale Verifikation von Programmen	129
5.4.1 Ein Beweis	129
5.4.2 Die Beweisidee	131
5.4.3 Die Zuweisung	131
5.4.4 Folgerungen	132
5.4.5 Sequenz	133
5.4.6 Beispiel: Austausch zweier Werte	133
5.4.7 Auswahlanweisung	135
5.4.8 Ein Beispiel für den Beweis eines Programms mit Auswahlanweisung	135
5.4.9 Schleife	138
5.4.10 Ein Beispiel für einen Beweis mit Hilfe einer Schleifeninvariante	138
5.4.11 Terminierung von Schleifen	141
5.4.12 Ein etwas größeres Beispiel: Bestimmung des Maximums einer Liste	142
5.4.13 Einige Probleme bei formalen Verifikationen	146
5.4.14 Das Halteproblem	148
5.4.15 Folgerungen aus den Grenzen der formalen Beweise	149
5.5 Reviews	150
5.6 Zusammenfassung	151

6	Der Zeitbedarf beim Programmablauf	153
6.1	Die Stoppuhr-Zeit	153
6.2	Beispiel: Die Suche eines Elementes in einer Liste	154
6.3	Die Messung der „Größe“ von Eingabedaten	156
6.4	Ein schon bekanntes Beispiel: Selectionsort	157
6.5	Ein neues Beispiel: Heapsort	159
6.5.1	Der Algorithmus	160
6.5.2	Laufzeit	165
6.6	Vergleich der Laufzeiten von Selectionsort und Heapsort	166
6.6.1	Das O -Kalkül	167
6.6.2	„Schwere“ Probleme und Heuristiken	168
6.7	Zusammenfassung	171
7	Wirkungen der Informatik	173
7.1	Ein Fehler im Space Shuttle	173
7.2	Ein Programmierfehler in einem Bestrahlungsgerät	174
7.3	Gefahren durch Prozeßrechner	176
7.4	Gefahren durch Datenbanken	176
7.5	Begriffssphären	177
7.6	Verantwortbarkeit des Einsatzes von Informatik in der Großtechnik	180
7.7	Programmieren als ethisches Problem	180
7.8	Ein letztes Beispiel: Militärische Frühwarnsysteme	181
7.9	Zusammenfassung	182
III	Anhang	187
	Symbolverzeichnis	189
	Musterlösungen	191
	Literaturverzeichnis	231
	Index	232