



Methoden der Programmerstellung für Tisch- und Taschenrechner

**Grundlagen,
Anwendungen, Grenzen**

**Helmut Schauer
Georg Barta**

Springer-Verlag
Wien New York



Dipl.-Ing. Dr. Helmut Schauer
Institut für Informationssysteme
Technische Universität Wien

Dipl.-Ing. Georg Barta
Schule für Datenverarbeitung
Siemens Data Gesellschaft m.b.H., Wien

Das Werk ist urheberrechtlich geschützt.
Die dadurch begründeten Rechte,
insbesondere die der Übersetzung, des Nachdruckes,
der Entnahme von Abbildungen, der Funksendung,
der Wiedergabe auf photomechanischem oder ähnlichem Wege
und der Speicherung in Datenverarbeitungsanlagen,
bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten.
© 1979 by Springer-Verlag/Wien

IBM-Composersatz: Springer-Verlag Wien;
Umbruch und Offsetdruck: Ferdinand Berger & Söhne OHG,
A-3580 Horn, NÖ.

Mit 134 Abbildungen

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Schauer, Helmut:

Methoden der Programmerstellung für Tisch- und
Taschenrechner: Grundlagen, Anwendungen, Grenzen /
Helmut Schauer; Georg Barta.

Wien, New York: Springer, 1979.

ISBN-13:978-3-211-81476-5 e-ISBN-13:978-3-7091-8513-1

DOI: 10.1007/978-3-7091-8513-1

NE: Barta, Georg:

ISBN-13:978-3-211-81476-5

Wer soll dieses Buch lesen ? (Vorwort)

Dieses Buch ist gedacht für alle diejenigen, die einen *programmierbaren* Tisch- oder Taschenrechner besitzen (vielleicht auch geschenkt bekommen haben), die aber *keine ausgebildeten Programmierer* sind und es auch nicht werden wollen, aber dennoch die beachtlichen Fähigkeiten ihres Rechners ausnützen möchten, um ihre Probleme – seien es nun mathematische, kaufmännische, technische oder andere – zu lösen, und dazu Grundkenntnisse über die Logik der Programmierung und die Erstellung von Programmen benötigen. Die Art des Kleinrechners ist dabei von untergeordneter Bedeutung: Ob es sich um Rechner mit oder ohne Druckwerk, Magnetstreifenspeicherung oder austauschbaren Speicher-Chips handelt, beeinflusst die allgemeinen Richtlinien nur wenig. (Und wo das der Fall ist, haben wir natürlich darauf hingewiesen.)

Vom Leser dieses Buches setzen wir voraus, daß er die *Bedienungsanleitung seines Rechners gelesen* hat und mit der „mechanischen Bedienung“ seines Gerätes vertraut ist. Er sollte ferner *den Rechner* während der Lektüre stets *griffbereit haben* und alle Beispiele nachvollziehen. Sonst wird nichts vorausgesetzt, was über die Schulmathematik hinausginge. Ausgenommen sind die Beispiele in den Kapiteln 8, 9 und 11, die fachspezifische Aufgaben lösen.

Die Entwicklung der elektronischen Rechenmaschinen, die gegen Ende des Zweiten Weltkrieges begann, verläuft nach wie vor äußerst stürmisch. Leistungen, wie sie heute programmierbare Rechner mit Leichtfertigkeit vollbringen, wären vor einem Vierteljahrhundert bestenfalls von einer Maschine in der Größe eines Zimmers zu erwarten gewesen. Entsprechend beträgt die Leistung heutiger Großcomputer ein Vielfaches derjenigen von damals.

VI Vorwort

Zwischen diesen Großrechnern und den vergleichsweise winzigen Tisch- und Taschenrechnern gibt es heute eine Vielfalt von Rechnern mittlerer Größe, die als „Minicomputer“ oder „Rechner der mittleren Datenverarbeitung“ bezeichnet werden und zu deren Aufgaben unter anderem die Steuerung von industriellen Prozessen, die Textverarbeitung und die Bewältigung kaufmännischer Probleme in kleineren Firmen zählen.

Die Entwicklung der Programmiermethodik hat mit der Entwicklung der Elektronik nicht Schritt halten können. Bei der Programmierung der Klein- und Kleinstrechner werden heute oftmals dieselben veralteten Methoden und Werkzeuge gebraucht, die man vor 20 Jahren bei Großrechnern eingesetzt hat, wodurch Programme entstehen, die unlesbar, unverständlich, un-änderbar und oftmals auch falsch sind. Während bei der Programmierung von Großrechnern heute Ansätze zur Überwindung dieser „Softwarekrise“ vorhanden sind, läuft die Programmierung von Kleinrechnern offenbar mitten in diese hinein. Wir haben daher in diesem Buch versucht, die Methoden und Werkzeuge der „Software-Technologie“ soweit wie möglich für Kleinstrechner zu übernehmen bzw. anzupassen*, ohne den Leser mit der dahintersteckenden Theorie zu belasten. Der Leser sollte dadurch imstande sein, Programme zu entwickeln,

- die er selbst nach einem Jahr noch versteht
- die andere verstehen können
- die in ihrem Funktionsumfang änderbar sind und
- die richtig sind.

Der letzte Punkt bedeutet, daß nicht nur für manche, sondern für *alle* denkbaren Eingabewerte entweder das richtige Ergebnis berechnet oder die Eingabe zurückgewiesen wird.

Zum Abschluß noch einige Bemerkungen:

Der Aufbau der Hardware (Kapitel 1) wird an Hand der Struktur der Hewlett-Packard-Rechner beschrieben, weil diese besser mit den Methoden der Software-Technologie in Übereinstimmung zu bringen sind als andere Produkte. Bei allen Beispielen findet sich jedoch eine Programmliste von einem HP-19 (Hewlett-Packard) und einem TI-57 (Texas Instruments), die als

* Für Fachleute einige Schlagworte: Strukturierte Programmierung, Struktogramme, stepwise refinement.

Vertreter der bedeutendsten unterschiedlichen „Hardware-Philosophien“ aufgefaßt werden können. Mit Hilfe der umfangreichen Erklärungen ist der Leser aber jederzeit in der Lage, die Programme so zu verändern, daß sie seinem speziellen Rechner gerecht werden – die Unterschiede betreffen nämlich nur Details.

Aus Gründen der Lesbarkeit wurden stets *nicht optimierte* Programme vorgestellt. Wir zweifeln nicht im geringsten daran, daß der Leser in der Lage ist, Programme zu erstellen, die entweder dasselbe leisten, aber weniger Speicherplätze bzw. kürzere Zeit beim Ablauf benötigen oder aber mehr leisten bzw. komfortabler sind. Im Gegenteil, das beweist nur, daß der Leser die Beispiele *und die Methodik*, die wir zugrunde gelegt haben, verstanden hat und anwenden kann ! Doch sollte man vor der Anwendung von Tricks zur Optimierung zweierlei bedenken: Erstens, die Auswirkungen auf Lesbarkeit, Verständlichkeit und Änderbarkeit. Zweitens, daß man ein Programm immer erst dann verbessern soll, wenn es bereits richtig läuft.

Unser Dank gebührt den Wiener Niederlassungen von Texas Instruments und Hewlett-Packard für die leihweise Überlassung der Rechner, Herrn Wolfgang Küller für das Codieren zahlreicher Programme, Fräulein Eva Lachkovics für das Erstellen und Beschriften der Zeichnungen und Fräulein Claudia Hainschink für das Schreiben des Manuskripts.

Schließlich wollen wir uns noch bei allen jenen griesgrämigen Lesern entschuldigen, die daran Anstoß nehmen, daß dieses Buch aus dem gewohnten sachlich trockenen Rahmen ausbricht. Wir wünschen auch ihnen beim Lesen mindestens ebensoviel Spaß, wie uns das Schreiben gemacht hat !

Wien, im Januar 1979

Helmut Schauer
Georg Barta

Inhaltsverzeichnis

1. Wie funktioniert der Rechner ?	1
(Aufbau der Hardware)	
2. Das kann jeder !	38
(Drei einfache Beispiele)	
3. Wie man es nicht machen soll	48
(Ein kompliziertes Beispiel)	
4. Das Innenleben von Programmen	55
(Programmstrukturen)	
5. Systematische Vorgangsweise	84
(Schrittweise Verfeinerung)	
6. Dialog mit dem Taschenrechner	114
(Ein- und Ausgabemöglichkeiten)	
7. Hat der Rechner immer recht ?	125
(Gedanken zur Genauigkeit)	
8. Beispiele aus der Numerik	139
8.1. Nullstelle einer Funktion mittels Bisektion	
8.2. Nullstelle einer Funktion mittels Regula falsi	
8.3. Nullstelle einer Funktion mittels Newton-Verfahren	
8.4. Lösung einer Differentialgleichung	
9. Technische Beispiele	153
9.1. Einspannmoment und -querkraft eines Kragträgers	
9.2. Schwingkreis	
9.3. Rydberg-Frequenz	

X	Inhaltsverzeichnis	
10.	Beispiele aus dem kaufmännischen Bereich	164
10.1.	Mehrwertsteuer von Einzelposten (gerundet)	
10.2.	Kalkulation in einem Buchverlag (am Beispiel eines wissenschaftlichen Buches)	
11.	Beispiele aus der Statistik	173
11.1.	Klassentest	
11.2.	Klassentest mit Mittelwertberechnung	
11.3.	Ausgleichsgerade	
11.4.	Exponentieller Ausgleich	
	Sachverzeichnis	189