

Wenn Sie den laufenden Basic-Kurs aufmerksam verfolgt haben, kennen Sie sich jetzt mit der Handhabung von Variablen und der Bildschirmverarbeitung aus. Vielleicht haben Sie sich auch schon mit der mehrfach verwendeten IF.THEN-Anweisung beschäftigt. Diesmal setzen wir uns nicht nur eingehend mit diesem wichtigen Basic-Bestandteil, sondern auch mit einer weiteren Schleifenart auseinander. Sie erfahren alles über die Basic-Anweisungen IF.THEN und FOR..NEXT. Dazu finden Sie natürlich wieder eine Menge Informationen, die im Handbuch des C 64 fehlen. Lesen Sie sich aber zunächst die Seiten 38 bis 40 im Handbuch des C 64 durch, um sich einen kleinen Überblick über die diesmal behandelten Anweisungen zu verschaffen.

Programmschleifen

Schleifen kennen Sie aus vielen Bereichen des normalen Alltags. Seien es Geschenkverpackungen oder gebundene Schnürsenkel. Doch was hat das alles mit Programmierung zu tun? Schleifen sind Programmteile, die eine oder mehrere Anweisungen ständig wiederholen, bis eine bestimmte, vom Programmierer festgelegte Bedingung eintritt. Man spricht in diesem Zusammenhang oft von wahren und falschen Aussagen, die Ihnen sicherlich aus der Mathematik bekannt sind. Programmschleifen werden also so lange durchlaufen, bis die gestellte Bedingung wahr ist. Sehen wir uns das an einem konkreten Beispiel an.

```
10 I=0
20 R=0
30 I=I+1
40 R=I*I
50 PRINT "DAS QUADRAT VON
  "I" IST "R
60 IF I < > 10 THEN 30
70 END
```

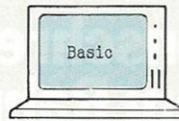
Hier haben wir eine Schleifenstruktur mit Hilfe der IF.THEN-Anweisung aufgebaut. Der Programmteil zwischen den Zeilen 30 bis 60 wird so lange durchlaufen, wie die Variable »I« nicht den Wert zehn angenommen hat.

Schleifen — Probleme

In Zeile 30 wird »I« jedesmal um eins erhöht. In Zeile 60 wird nun die Variable »I« geprüft. Die IF-Anweisung fragt hier ab, ob »I« ungleich zehn ist. Ist das der Fall springt das Programm zu dem hinter THEN stehenden Sprungziel, also in Zeile 30. Ein Programmablaufplan (PAP) veranschaulicht das hier verwendete Schleifenprinzip (Bild 1). Solche Ablaufpläne werden bei der Programmierung verwendet, um ein Problem vor der Umsetzung auf den Computer grafisch darzustellen. Die hier verwendete Methode hat den Nachteil, daß der Schleifenzähler (»I«) immer wieder im Programm hochgezählt und abgefragt (IF-Anweisung in Zeile 60) werden muß. Das Basic des C 64 bietet noch eine andere Möglichkeit zur Schleifenprogrammierung. Es handelt sich dabei um die FOR..NEXT-Anweisung. Bei dieser Schleifenform entfällt das Hochzählen und Abfragen des Zählers, wie das bei obiger IF.THEN-Konstruktion der Fall ist. Mit einer solchen FOR..NEXT-Schleife läßt sich das obige Beispiel folgendermaßen lösen:

```
10 FOR I=0 TO 10
20 R=I*I
30 PRINT "DAS QUADRAT VON
  "I" IST "R
40 NEXT I
50 END
```

In Zeile 10 wird hier als erstes festgelegt, wie oft die Schleife durchlaufen werden soll. »I« fungiert hier als Zähler, von dessen Wert es abhängt, ob der Vorgang nochmals abgearbeitet wird oder nicht. Die Obergrenze wird durch die Zahl, die hinter »TO« steht, festgelegt. Das bedeutet aber nicht, daß die Schleife jetzt zehnmal abgearbeitet wird. In diesem Fall geschieht dies genau elfmal (0 bis 10). Es ist nicht vorgeschrieben, mit welchem Wert der Zähler zu »starten« hat (in unserem Fall Null). Sie können genauso jede andere Zahl verwenden, sofern sie kleiner als die Obergrenze ist. Die Anzahl der Durchläufe errechnet sich aus der



Wollten Sie schon immer wissen, wie man Felder schnell und komfortabel behandeln kann? Wir zeigen Ihnen, wie Sie dieses Problem mit den Schleifenanweisungen des Basic V 2.0 lösen können.

Differenz der Untergrenze (hier Null) und der Obergrenze (in unserem Beispiel zehn) plus eins. Jetzt wissen Sie zwar wie eine Schleife gestartet wird, doch wo ist der Endpunkt zu suchen? Dieser wird mit der Anweisung NEXT gesetzt. Hinter NEXT steht noch einmal die Zähl-Variable. Diese kann bei diesen einfachen Schleifen auch weggelassen werden. Bei umfangreicheren Konstruktionen, wie wir sie weiter unten behandeln, wird die Angabe des Zählers hinter NEXT allerdings zwingend notwendig. Auch bei kleineren Schleifen sollte der Zähler wegen der besseren Übersicht mit angegeben werden. Stößt der Com-

puter auf NEXT wird als erstes der Zähler, in unserem Falle »I«, um eins erhöht. Danach prüft der Computer, ob die Obergrenze überschritten ist. Ist das nicht der Fall, kommen die zwischen den Anweisungen FOR.TO und NEXT eingeschlossenen Zeilen erneut zur Ausführung.

Die Anwendung

Jetzt haben Sie die theoretischen Grundlagen für die Programmierung mit IF.THEN und FOR..NEXT. Im folgenden sehen wir uns mögliche Anwendungen und Einsatzgebiete dieser Anweisungen etwas näher an.

```
10 REM BERECHNUNG BRUTTO AUS NETTO
20 DIM NETT(20)
30 DIM MWST(20)
40 DIM BRUT(20)
50 REM EINLESEN DER NETTO-WERTE
60 FOR I=0 TO 20
70 PRINT CHR$(147)
80 PRINT "NETTOWERT "I":";:INPUT NETT(I)
90 IF NETT(I)=9999 THEN 110
100 NEXT I
110 REM BERECHNEN
120 FOR I=0 TO 20
130 IF NETT(I)=9999 THEN 160
140 MWST(I)=NETT(I)*0.14
150 NEXT I
160 REM BERECHNUNG BRUTTO
170 FOR I=0 TO 20
180 IF NETT(I)=9999 THEN 210
190 BRUT(I)=NETT(I)+MWST(I)
200 NEXT I
210 REM AUSGEBEN DER WERTE
220 PRINT CHR$(147)
230 PRINT "NETTO", "MWST", "BRUTTO"
240 FOR I=1 TO 20
250 IF NETT(I)=9999 THEN END
260 PRINT NETT(I), MWST(I), BRUT(I)
270 NEXT I
280 END
```

Listing 1. Tabellenbearbeitung mit FOR..NEXT-Schleifen

wältigung mit Komfort

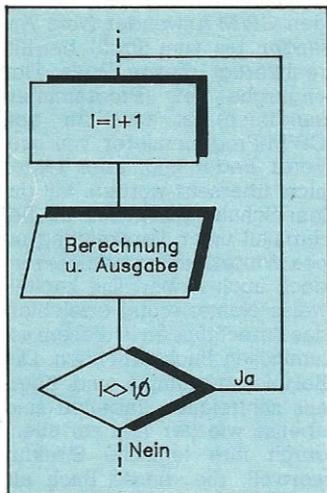


Bild 1. Ablauf einer IF...THEN-Schleife

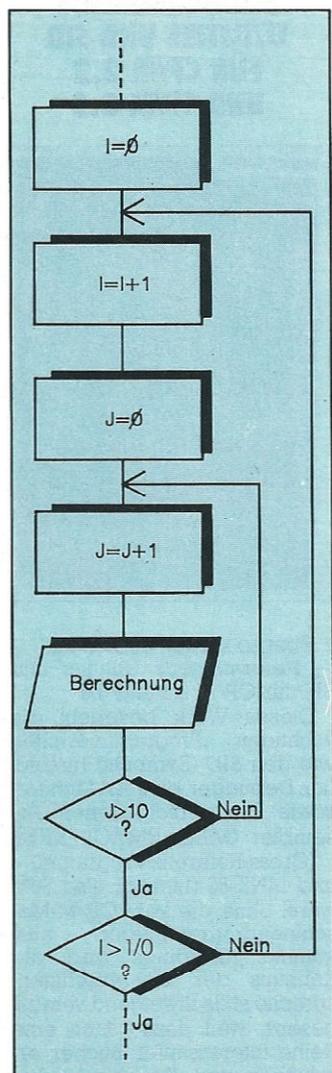


Bild 2. Schema einer verschachtelten FOR...NEXT-Schleife

Betrachten wir als erstes die IF...THEN-Anweisung. Sie haben gesehen, daß man mit IF...THEN Werte abfragen und entsprechend verzweigen kann. Damit aber nicht genug. Nach THEN kann nicht nur ein Sprungziel angegeben werden, wie Sie das bisher gesehen haben. Jede beliebige Basic-Anweisung kann THEN folgen. Mehrere Anweisungen werden durch einen Doppelpunkt getrennt. Dazu ein kleines Beispiel:

```

10 A$ = "TEST"
20 INPUT B$
30 IF B$="AENDERN" THEN A$="TEST1":GOTO 50
40 IF B$="NEU" THEN A$="TITEL":50 PRINT A$

```

Als erstes wird die Variable A\$ mit dem String »TEST« vorgelegt. Danach verlangt der Computer eine Eingabe. Haben Sie »AENDERN« eingegeben, tritt die IF...THEN-Anweisung in Zeile 30 in Aktion. A\$ erhält den String »TEST1« zugewiesen. Danach wird nach Zeile 50 gesprungen, wo A\$ mit dem veränderten Wert ausgegeben wird. Wird »NEU« eingegeben, arbeitet der Computer zwar ebenfalls die Zeile 30 ab, erkennt aber, daß die Variable B\$ nicht gleich »AENDERN« ist, bricht die IF...THEN-Anweisung ab und fährt sofort mit Zeile 40 fort. Dort findet sich aber eine in diesem Fall wahre Aussage, folglich wird die hinter THEN stehende Anweisung diesmal ausgeführt. Geben Sie weder »AENDERN« noch »NEU« ein, werden beide IF-Abfragen ergebnislos durchlaufen und der ursprüngliche String ausgegeben.

Sehen wir uns nun einige Anwendungen für die FOR...NEXT-Anweisung näher an. Am einfachsten ist hier die Möglichkeit, das Programm für kurze Zeit anzuhalten, wenn beispielsweise eine Meldung am Bildschirm ausgegeben wird.

```
10 FOR I=1 TO 500 : NEXT
```

Damit wird einfach der Schleifenzähler »I« von eins bis 500 hochgezählt. Wäh-

rend dieser Zeit passiert am Bildschirm nichts. Dem Programm-Benutzer kommt es so vor, als wenn der Computer eine Ruhepause einlegt.

Größtenteils wird die FOR...NEXT-Anweisung bei der Bearbeitung von dimensionierten Feldern eingesetzt. Eine genaue Erklärung zu Feldern finden Sie in Ausgabe 11/86 des 64'er-Magazins auf Seite 21. Konstruieren wir dazu ein kleines Beispiel. Über den INPUT-Befehl soll eine Tabelle mit Bedienerangaben gefüllt und danach verändert ausgegeben werden. Erfasst werden Netto-Preise, die dann mit Mehrwertsteuer und Bruttowert ausgegeben werden (Listing 1). Hier werden als erstes drei Felder dimensioniert. Danach liest die erste Schleife (Zeile 60 bis 100) die Netto-Werte ein. Die IF-Anweisung zwingt die Schleife zum Abbruch, wenn als Netto-Wert »9999« eingegeben wird. Sie sehen also, daß man eine FOR...NEXT-Schleife ohne weiteres verlassen darf. Der Schleifenzähler »I« behält dann den Wert, den er zum Zeitpunkt des Abbruchs hat. »I« wird in diesem Beispiel auch als Feldindex verwendet. Es wird also automatisch immer das nächsthöhere Feld bearbeitet, wenn ein neuer Schleifendurchlauf erfolgt. Nach der Eingabe-Schleife folgt die Berechnung der Mehrwertsteuer (Zeile 110 bis 150). Die Steuer wird in einer eigenen Tabelle abgelegt. In Element Null des »MWST«-Feldes steht also die Mehrwertsteuer zu dem Betrag, der in Element Null des »NETT«-Feldes untergebracht ist. Stößt die IF-Anweisung auf die ebenfalls im Netto-Feld gespeicherte Endebingung »9999« wird auch dieser Programmteil abgebrochen. Danach wird noch das »BRUT«-Feld mit den Bruttowerten versehen (Zeile 160 bis 200). Die Ausgabe erfolgt nach dem Bildschirmlöschen und ist tabellenartig aufgebaut (Zeile 210 bis 270).

Sobald innerhalb einer Schleife eine weitere Schleife programmiert wird, bezeichnet man dies als verschachtelte Schleife. Das klingt jetzt vielleicht unheimlich kompliziert, wird aber nach folgendem Beispiel schnell seine Schrecken verlieren. Um den ganzen Bildschirm mit einem Zeichen aufzufüllen, benötigen Sie zwei Schleifen. Eine muß die Position innerhalb der Bildschirmzeile bestimmen, die andere die Bildschirmzeile selbst.

```

10 PRINT CHR$(147)
20 FOR I=1 TO 24
30 FOR Y=1 TO 40
40 PRINT "A";
50 NEXT Y
60 NEXT I

```

Verschachteln von Schleifen

Die in Zeile 20 geöffnete Schleife ist für die Bildschirmzeile verantwortlich, in der das nächste »A« erscheinen soll. Allerdings muß, bevor eine neue Zeile beschrieben wird, die aktuelle mit Zeichen aufgefüllt werden. Dies erledigt die Schleife in Zeile 30 bis 50. Sie wird 40mal durchlaufen und bringt fortlaufend ein »A« auf den Bildschirm. Ist diese Schleife abgearbeitet, beginnt eine neue Bildschirmzeile und die Schleife in der Schleife tritt wieder in Aktion. Das geht so lange, bis der Zähler »I« die Obergrenze 24 überschritten hat, was der Anzahl der zu beschreibenden Bildschirmzeilen entspricht. In der Schleife in Zeile 20 bis 60 ist also festgelegt, wie oft die Schleife zwischen Zeile 30 und 50 aktiviert wird. Auf diese Art und Weise können Sie auch zwei verschiedene Felder bearbeiten. Das Ganze könnte zum Beispiel so aussehen:

```

10 FOR I=1 TO 10
20 FOR Y=1 TO 10 STEP 2
30 A(I)=A(I)+B(Y)
40 NEXT Y
50 NEXT I

```

In Zeile 20 sehen Sie hier eine weitere Variante bei der Programmierung von FOR...NEXT-Schleifen, die STEP-Anweisung. Diese veranlaßt das Erhöhen des Schleifenzählers um die hin-

Fortsetzung auf Seite 99

Fortsetzung von Seite 94

te Adresse eingeben (Zeile 1010). Drücken Sie eine andere Taste, so wird noch ein Zeichen mehr in die Datei geschrieben (print #3,1), da Master-Text dies als Endkennung benötigt. Danach wird die Datei geschlossen. Diese können Sie nun in Master-Text nutzen, indem Sie die Datenfelder im Text mit dem »A«-Befehl kennzeichnen. Zum Ausdruck wählen Sie »Rundschreiben«. Sie werden wieder aufgefordert, die Anzahl der Daten einzugeben. Hier tragen Sie dieselbe Zahl wie im Basic-Programm ein. Nun müssen Sie nur noch den Dateinamen eingeben.

Natürlich ist das kleine Basic-Programm zum Erstellen von Rundschreibendateien nur sehr begrenzt geeignet. Wenn Sie jedoch im Besitz einer Daten- oder Adreßverwaltung sind, können Sie in der Regel auch mit dieser Dateien erstellen. Sie müssen jedoch darauf achten, daß der Dateiname 16 Buchstaben lang sein muß (notfalls mit RENAME umbenennen) und daß die Datei mindestens ein Zeichen mehr enthalten muß, als nachher gedruckt werden soll. Dies können Sie nachträglich zum Beispiel so erreichen:

```
OPEN3,8,3,"dateiname,S,A":PRINT#3:CLOSE3
```

Sollten Sie eine eigene Routine entwickelt haben, die Dateien anderer Daten- oder Adreßverwaltungsprogramme für Master-Text umsetzt, so können Sie uns diese gerne unter dem Stichwort »Master-Text« zusenden.

4. Frage: Wie läßt sich die 80-Zeichenausgabe anhalten?

Die 80-Zeichenausgabe läßt sich mit der Leertaste anhalten. Da die Taste aber nur nach jeder Zeile einmal abgefragt wird, kann es sein, daß Sie diese schon mal etwas länger drücken müssen. Wenn der Rahmen seine Farbe verändert, hat der Computer den Tastendruck registriert. Das Anhalten der Ausgabe ist auch bei der Ausgabe auf dem Drucker möglich. Jedoch ist hier zu beachten, daß manche Drucker einen Pufferspeicher besitzen, dessen Inhalt auf jeden Fall noch zu Papier gebracht wird.

5. Frage: Wie nutze ich Steuerzeichen auf dem MPS 801?

Da die Drucker MPS 801 und MPS 802 keine verschiedenen Schriftarten beherrschen, erübrigt sich die Frage nach der Einstellung der Schriftarten von selbst. Die einzigen Möglichkeiten, die diese Drucker bieten, sind Breitschrift und Reversdruck. Softwaremäßig wurde in der Installation für den MPS 802 noch Unterstreichen eingebaut. Beim MPS 801 werden die Umlaute jedoch nicht revers gedruckt, da Master-Text die Umlaute nur in normaler Schrift enthält. Die Umlaute werden auf beiden Druckern auch nicht in Breitschrift gedruckt. Hier sind Sie als Leser gefordert, leistungsfähigere Druckroutinen zu schreiben und zur Veröffentlichung anzubieten, damit auch andere Leser davon profitieren. Dies gilt ebenso für von Ihnen entwickelte Treiber-Routinen für andere, seltenere Drucker. (Martin Pahl/sk)

Fortsetzung von Seite 27

ter STEP angegebene Schrittweite. So werden in diesem Beispiel konkret die Elemente 1, 3, 5, 7 und 9 des Feldes »B« angesprochen. Die Inhalte dieser Elemente werden zu jedem einzelnen des Feldes »A« addiert. Das Prinzip der verschachtelten Schleifen sehen Sie wieder in einem kleinen Programmablaufplan zusammengefaßt (Bild 2).

Die IF.THEN-Anweisung läßt sich ebenfalls verschachteln. Dazu wieder ein kleines Beispiel:

```
10 INPUT A
20 IF A < > 1 THEN
IF A=2 THEN
B$="OK"
30 PRINT B$
```

Hier wird eine zweite IF-Anweisung aktiviert sobald die Eingabe ungleich eins ist. Solche IF-Verschachtelungen sind natürlich durch die maximale Länge von 80 Zeichen einer Basic-Zeile nur begrenzt anwendbar.

Komfortable Schleifentechniken

Um Ihnen zu zeigen, daß es noch komfortabler geht, stellen wir Ihnen hier noch im Überblick die Möglichkeiten des Basic 7.0 vor, das im C 128 eingebaut ist. So bietet die IF-Anweisung einige Varianten, die diese Anweisung von der Einschränkung durch die maximalen Zeilenlänge entheben. Die IF-An-

weisung ist beim C 128 um die Anweisungen ELSE und BEGIN erweitert. Mit ELSE wird ein Alternativpfad eingeleitet. BEGIN ermöglicht das Programmieren von mehreren Programmzeilen innerhalb einer IF-Anweisung.

DO ist eine weitere Komponente des C 128-Basic zum Programmieren von Schleifen. Der Interpreter wird angewiesen, eine Schleife so lange zu durchlaufen, bis eine bestimmte Bedingung, die in der DO-Schleife näher deklariert wird, erfüllt ist. Diese Anweisung erlaubt sogar zwei verschiedene Abbruchbedingungen, deren Erklärung aber jetzt zu weit führen würde. Außerdem kann diese Schleife durch ei-

ne spezielle Anweisung jederzeit abgebrochen werden.

Aufbruch in die Datenverarbeitung

Nachdem Sie jetzt bereits mit den Grundbegriffen der Basic-Programmierung vertraut sind, lernen Sie in der nächsten Ausgabe die Arbeit mit Disketten-Daten näher kennen. Sie erfahren alles Wichtige über die Arbeit mit Dateien. Das inzwischen auch für die Heimcomputer wichtigste Speichermedium, die Diskette, verliert den vielleicht noch etwas rätselhaften Charakter und öffnet Ihnen das Tor zur Dateiverwaltung. (rf)

Fortsetzung von Seite 86, Hardcopy für Hypra-Basic

```
c020 : ad 20 f7 b7 8d d7 c1 8c 8a
c028 : d6 c1 aa 98 18 6d db c1 7c
c030 : 8d e1 c1 8a 6d dc c1 c9 c8
c038 : 01 90 07 ad e1 c1 c9 40 cd
c040 : b0 c8 20 fd ae 20 8a ad 8e
c048 : 20 f7 b7 c9 00 d0 c6 c0 ae
c050 : c8 b0 b7 8c da c1 20 fd 28
c058 : ae 20 8a ad 20 f7 b7 c9 a3
c060 : 00 d0 a7 cc da c1 90 a2 8f
c068 : c0 c8 b0 9e 8c de c1 ee 31
c070 : db c1 d0 03 ee dc c1 ee 7b
c078 : de c1 20 a1 c1 ae d6 c1 e3
c080 : ad d7 c1 8d e0 c1 ad db c5
c088 : c1 8d d8 c1 da c1 8d 62
c090 : d9 c1 a9 00 8d 6e c1 ad 63
c098 : dc c1 8d d9 c1 a2 00 a0 66
c0a0 : 07 98 48 8a 48 20 6f c1 1d
c0a8 : 6e dd c1 68 aa 68 a8 88 24
```

```
c0b0 : d0 ef 38 6e dd c1 ad dd b2
c0b8 : c1 20 d2 ff ee d8 c1 d0 9c
c0c0 : 03 ee d9 c1 e8 ec d6 c1 be
c0c8 : d0 d5 ad e0 c1 f0 07 a9 1d
c0d0 : 00 8d e0 c1 f0 c9 a9 0d 25
c0d8 : 20 d2 ff ad da c1 18 69 06
c0e0 : 07 8d da c1 ad de c1 18 a6
c0e8 : 69 06 cd da c1 b0 8e 4c 97
c0f0 : c4 c1 a5 fa a6 fb a4 fe 38
c0f8 : 98 29 f8 85 fe 85 fc a9 77
c100 : 00 85 fd 06 fc 26 fd 06 08
c108 : fc 26 fd 18 a5 fc 65 fe 70
c110 : 85 fc a5 fd 69 00 85 fd e5
c118 : 06 fc 26 fd 06 fc 26 fd c2
c120 : 06 fc 26 fd 98 29 07 18 0d
c128 : 65 fc 85 fc a5 fd 69 00 fc
c130 : 85 fd 18 a5 fa 29 f8 65 16
c138 : fc 85 fc a5 fb 65 fd 85 d8
c140 : fd 18 a9 00 65 fc 85 fc 02
c148 : a9 20 65 fd 85 fd a5 fa ef
c150 : 29 07 49 07 aa a9 01 ca c2
```

```
c158 : 30 03 0a d0 fa a0 00 8d 76
c160 : 6d c1 b1 fc 2d 6d c1 d0 a1
c168 : 02 18 60 38 60 00 00 ad f7
c170 : 6e c1 18 6d da c1 cd de 23
c178 : c1 b0 21 85 fe ad d8 c1 cf
c180 : 85 fa ad d9 c1 85 fb 20 a1
c188 : f2 c0 08 ee 6e c1 ad 6e 43
c190 : c1 c9 07 d0 05 a9 00 8d ca
c198 : 6e c1 28 60 18 08 4c 8b 07
c1a0 : c1 a9 01 a2 04 a0 07 20 74
c1a8 : ba ff 20 c0 ff 90 01 60 cb
c1b0 : a2 01 20 c9 ff a9 0d 20 d6
c1b8 : d2 ff a9 08 20 d2 ff a9 e1
c1c0 : 0d 4c d2 ff a9 0f 20 d2 e1
c1c8 : ff a9 0d 20 d2 ff 20 cc 2a
c1d0 : ff a9 01 4c c3 ff 00 00 aa
c1d8 : 00 00 00 00 00 00 00 d9
c1e0 : 00 00 2c 01 58 60 4c 48 56
```

Listing 4. (Schluß)