

# Praktische Anwendungen mit dem **SINCLAIR**

**J. Hillreiner**

Strukturiertes BASIC  
Textverarbeitung  
Tabellenkalkulation  
Datenbank  
Grafik

ISBN 3-88963-222-X

Es kann keine Gewähr dafür übernommen werden, daß die in diesem Buche verwendeten Angaben, Schaltungen, Warenbezeichnungen und Warenzeichen, sowie Programmlistings frei von Schutzrechten Dritter sind. Alle Angaben werden nur für Amateurzwecke mitgeteilt. Alle Daten und Vergleichsangaben sind als unverbindliche Hinweise zu verstehen. Sie geben auch keinen Aufschluß über eventuelle Verfügbarkeit oder Liefermöglichkeit. In jedem Falle sind die Unterlagen der Hersteller zur Information heranzuziehen.

Nachdruck und öffentliche Wiedergabe, besonders die Übersetzung in andere Sprachen verboten. Programmlistings dürfen weiterhin nicht in irgendeiner Form vervielfältigt oder verbreitet werden. Alle Programmlistings sind Copyright der Fa. Ing. W. Hofacker GmbH. Verboten ist weiterhin die öffentliche Vorführung und Benutzung dieser Programme in Seminaren und Ausstellungen. Irrtum, sowie alle Rechte vorbehalten.

**COPYRIGHT by Ing. W. HOFACKER ©1985**  
**Tegernseer Str. 18, 8150 Holzkirchen**

1. Auflage 1985

Gedruckt in der Bundesrepublik Deutschland — Printed in West-Germany —  
Imprime'en RFA.

**J. Hillreiner**

**Praktische  
Anwendungen  
mit dem  
SINCLAIR QL**



# Vorwort

Der SINCLAIR QL ist einer der ersten Rechner der neuen Generation auf dem Micro-Computer Markt. Sein supermoderner 32-Bit Prozessor, der Motorola 68008, gehört zur Familie der leistungsfähigsten Prozessoren, die in Kleinrechnern verwendet werden.

Aber nicht nur die Hardware ist neu. Für den SINCLAIR QL wurden ein neues Betriebssystem, das QDOS, und ein völlig neues strukturiertes BASIC, das SuperBASIC entwickelt.

Dieses Buch soll Ihnen helfen, diesen aussergewöhnlichen Rechner praktisch einzusetzen, und die mitgelieferte Software (Tabellenkalkulation, Datenbank, Textverarbeitung und Graphik) von Anfang an optimal zu nutzen.

Für das Programmieren mit dem QL wünsche ich Ihnen weiterhin viel Spaß.

Holzkirchen

Johann Hillreiner



# Inhaltsverzeichnis

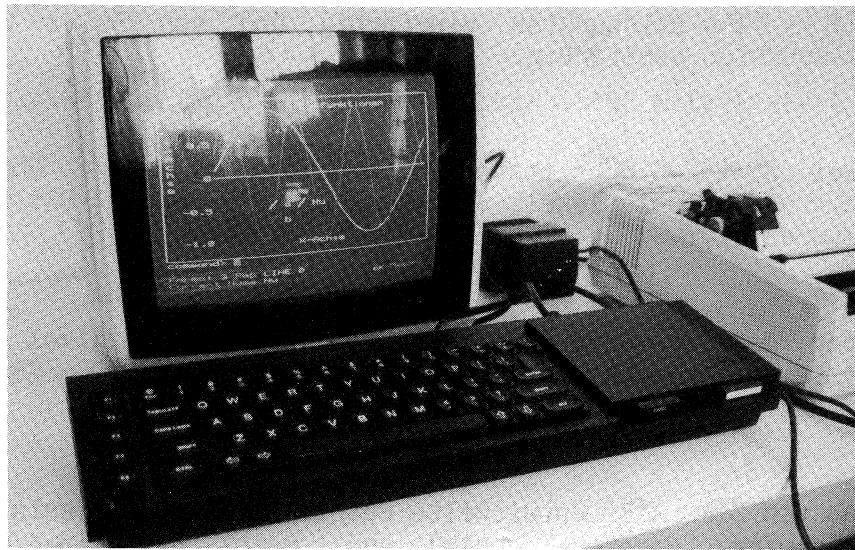
<b>1. Allgemeines über den SINCLAIR QL</b> .....	<b>1</b>
1.1    Einführung .....	1
1.2    Grundlagen .....	5
<b>2. Das strukturierte SuperBASIC</b> .....	<b>11</b>
2.1    Befehle .....	11
2.2    Programmieren in SuperBASIC .....	51
2.2.1    Elemente des strukturierten BASIC .....	51
2.2.2    Beispielprogramme .....	55
Programm 23Holz .....	55
Programm Notitzbuch .....	57
Programm Uhr .....	62
Programm Farben ändern .....	65
Programm Tonerzeugung .....	67
<b>3. Tabellenkalkulation mit ABACUS</b> .....	<b>71</b>
3.1    Die Befehle .....	71
3.2    Anwendungsbeispiele .....	85
Kassenbuch .....	85
Lineare Regression .....	88
Funktionswerte berechnen .....	93
<b>4. Das Datenbanksystem ARCHIVE</b> .....	<b>97</b>
4.1    Befehle .....	97
4.2    Adressenverwaltung unter ARCHIVE .....	116
<b>5. Das Grafikprogramm EASEL</b> .....	<b>127</b>
5.1    Befehle .....	127
5.2    Anwendungen .....	135
5.2.1    Importieren einer mit ABACUS erstellten Datei .....	135

5.2.2 Eingeben einer Funktion . . . . .	139
6. Das Textverarbeitungsprogramm QUILL . . . . .	143
7. Druckeranschluß an Star Delta-10 . . . . .	157
<b>ANHANG – Befehlsübersicht . . . . .</b>	<b>161</b>
SuperBASIC . . . . .	165
ZEICHENBEHANDLUNG . . . . .	169
GRAPHIK . . . . .	170
MATHEMATISCHE FUNKTIONEN . . . . .	173
Zeichen-Funktionen . . . . .	174
OPERATOREN . . . . .	175

# Allgemeines über den SINCLAIR QL

## 1.1 Einführung

Der SINCLAIR QL (QL heisst Quantum Leap = Quantensprung) ist die neueste Entwicklung der britischen Firma SINCLAIR auf dem Heimcomputermarkt.



SINCLAIR QL

Die Zentraleinheit des QL ist der Motorola 68008, ein 32-Bit Prozessor mit 7,5 MHz Taktfrequenz. Die Kapazität des Speichers beträgt in der Grundversion 128 KB und kann auf 640 KB ausgebaut werden. Davon werden 32 KB für das Betriebssystem QDOS und das neue SuperBASIC, ein strukturiertes BASIC benötigt.

Zur Steuerung der Tastatur, der akustischen Signale, der RS-232-C-Empfangs- und der Echtzeituhr-Funktionen ist ein Intel 8049 eingebaut.

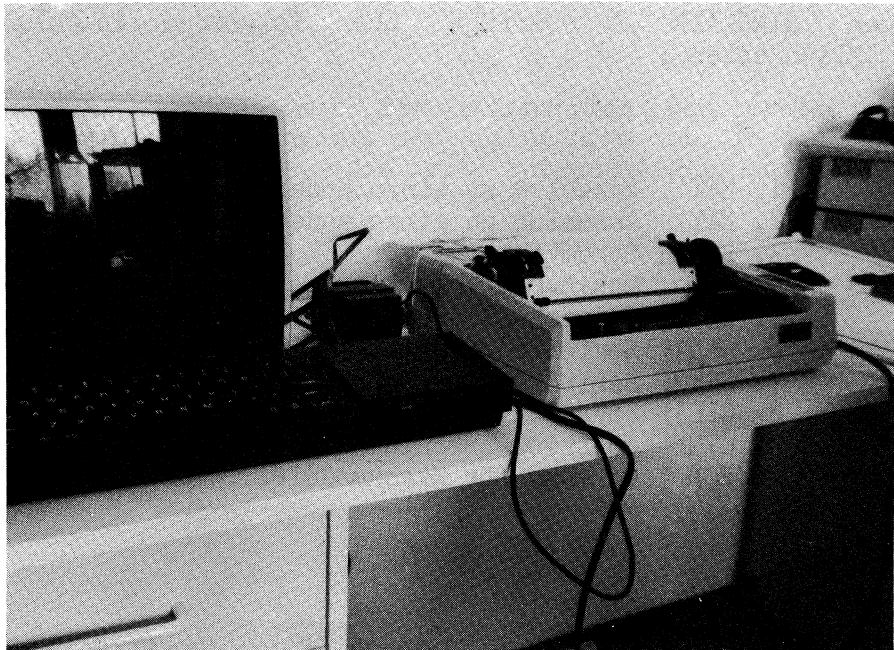
Als Massenspeicher dienen zwei eingebaute Microdrives, mit je ca. 110 KB Speicherkapazität. Microdrives sind von SINCLAIR entwickelte Bandlaufwerke für besonders kleine Endloskassetten. Ihre durchschnittliche Zugriffsgeschwindigkeit beträgt 3,5 Sekunden und die Ladegeschwindigkeit von Daten in den internen RAM beträgt maximal 15 KB pro Sekunde. Es können noch zusätzlich sechs Microlaufwerke angeschlossen werden.

Die hochauflösende Graphik erlaubt die Darstellung von 512 x 256 Bildpunkten vierfarbig und von 256 x 256 Bildpunkten achtfarbig.

Zwei RS-232-C Schnittstellen arbeiten mit einer Übertragungsgeschwindigkeit von 75 bis 19200 Baud (= Bit pro Sekunde).

Über das Netzwerk können Daten zwischen maximal 64 Systemen mit einer Geschwindigkeit von 100 KBaud ausgetauscht werden.

Das von SINCLAIR entwickelte Betriebssystem QDOS erlaubt Windowing (Fenster-Technik) und macht die Ein- und Ausgabe peripherieunabhängig.



**SINCLAIR QL mit Drucker**

Folgende Software wird auf Mikrokassetten mitgeliefert :

QUILL (Textverarbeitung)

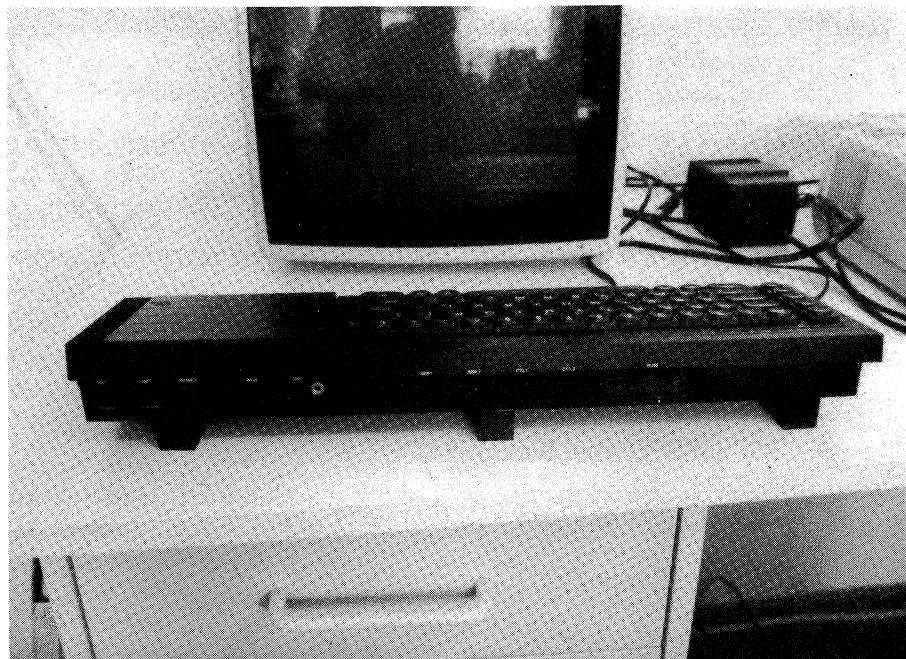
ARCHIVE (Datenbank)

ABACUS (Tabellenkalkulation)

EASEL (Graphik)

Der SINCLAIR QL verfügt über folgende Anschlußmöglichkeiten :

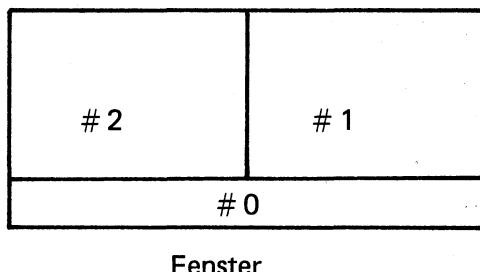
1. NET Anschluß an einen anderen QL.
2. NET Anschluß an einen anderen QL.
3. POWER Stromversorgung.
4. RGB Monitor-Stecker.
5. UHF Fernseh-Stecker.
6. SER1 Serielle RS-232-C Schnittstelle.
7. SER2 Serielle RS-232-C Schnittstelle.
8. CTL1 Joystick-Anschluß.
9. CTL2 Joystick-Anschluß
10. ROM Öffnung für Steckmodule.



**SINCLAIR QL von hinten**

## 1.2 Grundlagen

Der QL arbeitet mit der sogenannten Fenster-Technik (Windowing), das heißt, der Bildschirm ist in einzelne Fenster aufgeteilt, die mit unterschiedlichen Kanalnummern angesprochen werden können.



Bei Monitorbetrieb werden in Fenster #0 Eingaben gemacht, in #1 wird die Ausgabe angezeigt, und in Fenster #2 steht das Programm. Bei Anschluß eines Fernsehers fallen Kanal 1 und Kanal 2 zusammen.

### EDITOR :

Der SINCAIR QL hat einen Zeileneditor, d.h. es ist nur das Editieren einer Programmzeile möglich.

Das Ändern einer Programmzeile geschieht durch den Befehl EDIT xx, wobei xx eine beliebige Zeilennummer ist. Dann können Sie mit den Cursor-Tasten die Zeile ändern. CTRL und <= löscht das Zeichen links vom Cursor, CTRL und => löscht das Zeichen auf dem der Cursor steht.

### Bildschirrmodi :

#### MODE 512 :

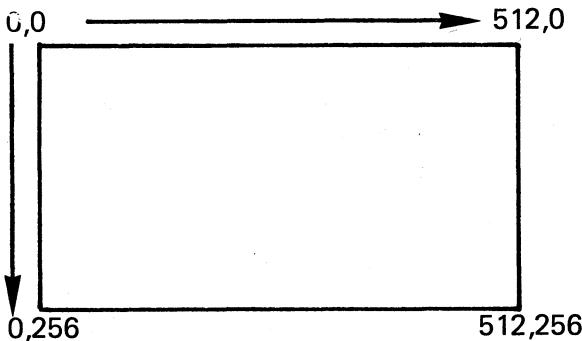
Hochauflösende Graphik mit 512 Bildpunkten (Pixel) Breite und 256 Punkten Höhe. Es sind nur die Farben Schwarz, Rot, Grün und Weiß verfügbar.

## MODE 256 :

Graphik mit geringer Auflösung (256 mal 256 Pixel). Es sind alle sieben Farben darstellbar : Schwarz, Blau, Rot, Magenta, Grün, Cyan, Gelb, Weiß. Außerdem kann mit FLASH 1 das Blinken von Zeichen "angeschaltet" werden.

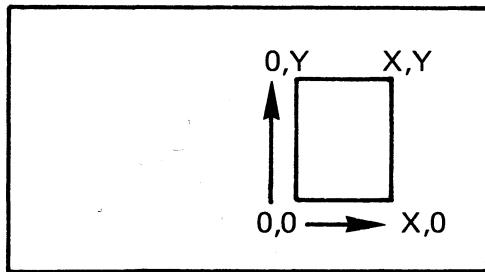
## Graphik :

Das Pixel-Koordinatensystem wird benutzt, um die Position und Größe von Fenstern (Windows), Blöcken (Blocks) und des Cursors anzugeben. Der Ursprung des Koordinatensystems ist in der linken oberen Ecke des jeweiligen Fensters. Die Koordinatenangaben laufen von 0 bis 512 horizontal und von 0 bis 256 vertikal.



PIXEL – Koordinatensystem

Das Graphik-Koordinatensystem hat den Nullpunkt in der linken unteren Ecke des Fensters. Da die Bildpunkte, die der QL darstellt, nicht quadratisch sind, wurde ein davon unabhängiges Format gewählt. Es wird das kartesische Koordinatensystem benutzt, mit den Angaben 100 mal 100. Durch den Befehl SCALE kann der Maßstab geändert werden, und somit ist es möglich, beliebige Figuren oder Funktionen maßstabsgerecht ohne Umrechnungen darzustellen.



GRAPHIK – Koordinatensystem

### Datentypen :

Die Vereinbarungen der verschiedenen Datentypen entsprechen dem in anderen BASIC-Dialekten üblichen Format. Eine Besonderheit ist die Verwendung des Unterstrichs \_ für Angaben, die das System betreffen, zum Beispiel MDV1\_ oder COPY\_N.

Ganze Zahlen (Integer) werden durch Anhängen eines % gekennzeichnet.

a%

Breite%

ber%

Gleitpunktzahlen sind Standard und werden ohne Kennzeichnung festgelegt. Der Bereich läuft von -10^615 bis +10^615.

Ergebnis

Zähler

i

Zeichenketten (Strings) werden durch Anfügen eines \$ an den Namen gekennzeichnet. Bei der Wertzuweisung können einzelne (') oder doppelte (") Hochkommas verwendet werden.

```
Name$  
Wohnort$  
a$ = "Januar"  
x$ = 'Februar'
```

## Die Operatoren

+	Addition
-	Subtraktion
*	Multiplikation
/	Division
^	Potenzierung
DIV	Ganzzahlige Division
MOD	Divisionsrest
OR	ODER
AND	UND
XOR	Exklusives ODER
NOT	Negierung
	Bitweise ODER
&&	Bitweise UND
^^	Bitweise exklusives ODER
~~	Bitweise NICHT
=	Gleich
==	Fast gleich
>	Grösser
<	Kleiner
>=	Grösser gleich
<=	Kleiner gleich
<>	Ungleich

## Speicherbelegung des Motorola 68008 :

Der Motorola 68000 Speicherbaustein kann bis zu einem MegaByte an Speicherplatz adressieren, d.h. den Bereich von 00000 bis FFFFF Hexadezimal.

FFFFF	RESERVIERT	Erweiterungs E/A
C0000	RESERVIERT	Speichererweiterung RAM
40000	RAM 96 KB	Haupt RAM
28000	RAM 32 KB	Bildschirmspeicher
20000	RESERVIERT	Erweiterung E/A
1C000	I/O	QL E/A
18000	RESERVIERT	Erweiterung E/A
10000	ROM 16 KB	ROM Steckmodul
0C000	ROM 48 KB	System ROM
00000		

Programme laden.

Nach dem Einschalten des Rechners müssen Sie F1 bzw. F2 drücken, um Monitor- oder Fernsehbetrieb auszuwählen. Danach läuft das Mikrolaufwerk 1 an, um festzustellen, ob sich darin eine Kassette befindet. Wenn ja, dann wird das Programm mit dem Namen "BOOT" automatisch geladen. Da das Ladeprogramm bei allen vier mitgelieferten Programmpaketen auch "BOOT" heißt, können Sie das gewünschte Programm einfach laden, indem Sie nach dem Einschalten die benötigte Kassette in Laufwerk 1 einschieben.

Dieses automatische Laden habe auch ich mir zunutze gemacht, um, wenn ich mit SuperBASIC arbeite, andere Farben für Bildschirm und Schrift zu wählen. Dabei habe ich folgendes Programm auf einer Mikrokassette unter dem Namen "BOOT" abgespeichert.

```
10 MODE 4
20 FOR I = 0 TO 2
30 PAPER #I,0 : INK #I,7 : CLS #I : CSIZE #I,1,0
40 END FOR I
50 DLINE 10 TO 50
```

Für alle drei Fenster #0, #1, und #2 wird die Hintergrundfarbe Schwarz, die Schriftfarbe Weiß und die Zeichengröße 8 mal 10 Bildpunkte gesetzt. Dies ergab bei meinem monochromen grünen Monitor das angenehmste Bild.

# 2

# Das strukturierte SuperBASIC

## 2.1 Befehle

Syntax der Befehle :

Zur Darstellung der Befehle wurden folgende Zeichen verwendet :

a,b,c Alle Parameter müssen angegeben werden.

[ ] Die Parameter in Klammer sind wahlfrei.

x,... Die Punkte geben an, daß der vorhergenannte Parameter beliebig oft wiederholt werden kann.

(x) Der Parameter in Klammern muß angegeben werden.

### **ABS(x)**

Absolutbetrag von x.

ABS(-4) ist 4.

ABS(4) ist 4.

## **ACOT (Winkel)**

Arkuscotangens eines Winkels im Bogenmaß.

```
PRINT ACOT(A + B)  
PRINT ACOT(3.1256)
```

## **ADATE sek**

Einstellung der eingebauten Uhr. ADATE 60 stellt die Uhr um eine Minute vor. ADATE -7200 stellt um zwei Stunden zurück.

## **ARC x,y TO x,y,Winkel [TO x,y,Winkel ...]**

Zeichnet einen Kreisbogen zwischen zwei im Graphik-Koordinatensystem angegebenen Punkten.

## **ARC\_R TO x,y,Winkel [TO ...]**

Zeichnet relativ (von Graphik-Cursor aus).

Zeichnen zweier Kreisbögen :

```
ARC 15.23 TO 45.66,PI/4  
ARC R TO 87,56,PI/8
```

## **AT Zeile,Spalte**

Festlegen der Position für den Print-Befehl (im Pixel-Koordinatensystem). Abhängig von der Zeichengröße.

AT 5,10 druckt in Zeile 5, Spalte 10.

## ATAN (Winkel)

Arkustangens eines Winkels im Bogenmaß.

```
PRINT ATAN(W)
PRINT ATAN(2.456)
```

## AUTO a,b

Automatische Zeilennummer-Generierung, beginnend mit Zeile a, Inkrementierung von b.

AUTO 100,10 generiert Zeilennummern 100, 110, 120 ...

## BAUD Rate

Setzt Übertragungsrate für Datenübertragung. Erlaubt sind 75, 300, 600, 1200, 2400, 4800, 9600, 19200 Baud.

BAUD 1200.

## BEEP Dauer, Tonhöhe, Tonhöhe\_2, Grad\_x,Grad\_y, Schwingung, Verstimmung, Zufall

Erzeugen von Tönen.

Dauer : Dauer des Tones in Einheiten zu 72 Mikrosekunden.  
Von -32768 bis 32767

Tonhöhe : Von 1 (hoch) bis 255 (niedrig).

Tonhöhe\_2 : Obere Tonhöhe. Ton schwingt zwischen den beiden Tonhöhen. von 0 bis 255.

Grad\_x : Die Rate, mit der der Ton Grad\_y (Werte von -8 bis 7) mal schwingt. Von 0 bis 15 .  
Schwingung : Anzahl der Schwingungen des Tones.  
Bereich von 0 bis 15.  
Verstimmung : Bereich von 0 bis 15  
Zufall : Bereich von -32768 bis 32767

Wird Tonhöhe\_2 angegeben, so müssen auch Grad\_x und Grad\_y angegeben werden.

BEEP 10000,30  
BEEP 8000,20,69,0,0,1,1,1000  
BEEP 30000,20,69,1,15,2,2,1000

BEEP ohne Parameter stoppt jede Tonerzeugung.

Beispielprogramm :

```
10 REMark LIED
20 RESTORE
30 REPeat TON
40 READ D,P
50 IF D = 0 THEN EXIT TON
60 BEEP 1000*D,P
70 PAUSE 12
80 END REPeat TON
90 DATA 3,50,3,40,3,38,3,50,3,40,3,38
100 DATA 3,50,3,40,3,38,8,50
110 DATA 0,0
```

## BEEPING

Ist eine Funktion, die den Wert 0 (Falsch) hat, wenn der QL keine Töne erzeugt, und einen Wert ungleich 0 (Wahr) hat, wenn der QL Töne erzeugt.

## **BLOCK [Kanal,] Breite,Höhe,X,Y,Farbe**

Malt einen Block der angegebenen Farbe, Größe und Form an den Koordinaten X, Y. Mit Kanal kann ein Fenster angegeben werden.

BLOCK 20,15,0,0,4

ergibt einen 20 mal 15 großen Block bei 0,0 in Grün.

Beispielprogramm : Erzeugt ein zufälliges Muster bunter Flächen.

10 CLS

20 BLOCK 20,15,RND(220),RND(185),RND(7)

30 GO TO 20

## **BORDER [Kanal,] Breite[,Farbe]**

Umrundung für das Fenster des angegebenen Kanals. Farbe und Kanal sind wahlfreie Parameter. Die Größe des Fensters bleibt bis zu einem neuen BORDER - Befehl reduziert.

BORDER #2,2,0 Umrändung in Fenster 2 in Schwarz, zwei Pixel breit.

## **CALL Adresse,Daten[,...]**

Zugriff auf Maschinen-Code. Es können maximal 13 Parameter eingegeben werden.

CALL 124012,1,0,0

## **CHR\$(X)**

Zeichen, dessen Ordnungszahl X ist.

PRINT CHR\$(65) ergibt A.

## CIRCLE [Kanal,] X,Y,Radius,Exzentrizität,Winkel

Kreis an den Koordinaten X, Y mit dem angegebenen Radius. Exzentrizität ist das Verhältnis der beiden Durchmesser einer Ellipse.

Winkel ist der Winkel zwischen der Hauptachse der Ellipse zur Vertikalen.

CIRCLE 40,50,30    Kreis bei 40, 50 mit dem Radius 30

CIRCLE 40,50,20,0.6,PI/6

Ellipse bei 40, 50 mit dem 1. Radius 20 und dem 2. Radius 0.6 mal 20, um PI/6 gedreht.

### Programmbeispiele :

```
10 FOR X = 1 TO 20
20 CLS
30 FOR I = 1 TO 40 STEP X
40 CIRCLE 50,50,I
50 END FOR I
60 END FOR X
70 CLS
80 FOR Y = 1 TO 7
90 FOR X = 1 TO 50
100 CIRCLE RND(100),RND(100),Y
110 END FOR X
120 END FOR Y
```

```
10 CLS
20 E = 45
30 F = 0
40 FOR X = .2 TO 10 STEP .2
50 INK F
60 F = F + 2
70 FOR I = 1 TO E STEP X
80 CIRCLE 50.50,I
90 END FOR I
100 E = E - 1
110 END FOR X
```

```
10 CLS
20 FOR i = 0 TO PI STEP PI/6
30 FOR a = 0 TO 4.5 STEP .5
40 CIRCLE 47,50,10,a,i
50 END FOR i
60 END FOR a
```

```
10 CLS : INK 6
20 FOR a = 0 TO 4.5 STEP .5
30 FOR i = 0 TO PI STEP PI/6
40 CIRCLE 47,50,10,a,i
50 END FOR i
60 END FOR a
70 PAUSE 50 : INK 2
80 FOR a = 4.5 TO 0 STEP -.5
90 FOR I = PI TO 0 STEP -PI/6
100 CIRCLE 47,50,10,a,i
110 END FOR i
120 END FOR a
130 INK 6
```

## **CLEAR**

Löscht den Bereich der Variablen und gibt Speicherplatz für QDOS frei.

CLEAR

## **CLOSE Kanal**

Schließt den angegebenen Kanal. Angesprochene Fenster werden gelöscht.

CLOSE #5

## **CLS [Kanal,] [Teil]**

Löscht den Bildschirm teilweise oder ganz, je nach angegebenen Parametern.

CLS 0 ganzer Schirm  
CLS 1 Schirm außer Cursor-Zeile  
CLS 2 Unterer Teil außer Cursor-Zeile  
CLS 3 Ganze Cursor-Zeile  
CLS 4 Rechter Teil der Cursor-Zeile mit Cursor-Position

CLS #1,0 oder CLS #1 Löscht ganzes Fenster #1

## **CODE ("Zeichen")**

Ergibt den internen Code des angegebenen Zeichen.  
Umkehrung zu CHR\$(Zahl).

CODE ("B") ist 66.

## **CONTINUE**

## **RETRY**

Mit Continue kann man ein abgebrochenes Programm fortführen. Mit Retry kann man einen fehlerhaften Befehl nochmals ausführen.

## **COPY Eingabe TO Ausgabe**

## **COPY\_N Eingabe TO Ausgabe**

Kopiert eine Datei von Eingabe nach Ausgabe bis zur Dateiende-Marke (End-of-file).

COPY\_N kopiert ohne den Kopf einer Datei auf Microdrive.

COPY\_N mdv2 Programm TO ser1.

## **COS(Winkel)**

Berechnet den Kosinus eines Winkels (Im Bogenmaß).

COS (1.507)

## **COT (Winkel)**

Berechnet den Kotangens eines Winkels.

COT (2.456)

## **CSIZE [Kanal,] Breite,Höhe**

Ändert die Zeichengröße. Standard ist 2,0 in Mode 8 und 0,0 in Mode 4.

Breite :	0	1	2	3	Höhe :	0	1
----------	---	---	---	---	--------	---	---

Pixel :	6	8	12	16	10	20
---------	---	---	----	----	----	----

CSIZE 2,1 definiert eine Breite von 12 und eine Höhe von 20 Bildpunkten.

## **CURSOR [Kanal,] x,y [,x,y]**

Positioniert den Cursor an die angegebenen Koordinaten (im Pixel-Koordinatensystem) des mit Kanal angesprochenen Fensters. Mit vier Koordinaten werden die ersten zwei als Koordinaten des Graphik-Koordinatensystems interpretiert, und die letzten beiden geben die Position relativ dazu an.

CURSOR 0,5

CURSOR 0,0,6,3

## DATA Ausdruck,...

Mit DATA können Daten in einem Programm definiert werden. Diese Daten können mit dem READ-Befehl gelesen werden. Siehe auch READ und RESTORE.

```
DATA 3,6,5.22,1
DATA "Januar", "Februar"
```

## DATE\$

### DATE

Date\$ ist eine Funktion, die das Datum als Zeichenkette im Format Jahr Mon Tag HH:MM:SS anzeigt.  
Date zeigt das Datum als Gleitpunktzahl.  
Eine Zahl hinter DATE\$, z.B. Date\$(1.234532E6) wird als Datum im Gleitpunktformat interpretiert und in eine Zeichenkette umgewandelt.

```
DATE$
```

## DAY\$

Funktion zur Abfrage des Wochentages. Steht ein Datum nach DAY\$, z.B. DAY\$(01/01/1985) wird der Wochentag dieses Datums angezeigt.

```
DAY$(24/02/1985)
```

**DEFIne FuNction Name [Parameter]**

**[LOCAL Name, Name, ...]**

**Anweisungen**

**RETURN Ausdruck**

**END DEFIne**

Definiert eine Funktion in SuperBASIC.

Die Funktion wird aufgerufen durch Angabe des Namens. Rekursive Aufrufe sind auch möglich. Formale und lokale Parameter müssen in Klammern stehen. Wird eine Variable nicht lokal definiert (durch LOCAL), so ist sie global. Ergebnisse werden mit dem RETURN-Befehl übergeben.

Beispiel:

```
..  
50 DEFIne FuNction Hochvier (x)  
60 LOCAL y  
70 y = x * x * x * x  
80 RETURN y  
90 END DEFIne  
..  
200 PRINT Hochvier (5)
```

**DEFIne PROCEDURE Name [Parameter]**

**[LOCAL Name, Name, ...]**

**Anweisungen**

**[RETURN]**

**END DEFIne**

Diese Befehlsfolge definiert eine SuperBASIC Prozedur.

Die formalen Parameter müssen bei der Prozedur-Definition in Klammern stehen, beim Prozedur-Aufruf nicht. Lokale Variablen werden mit LOCAL definiert. Der Prozedur-Aufruf erfolgt durch Angabe des Prozedurnamens mit der Liste der aktuellen Parameter. Auch rekursive Aufrufe sind möglich.

Beispiel : Zeichnen mit der "Turtle"-Graphik.

```
10 REMark MUSTER MIT QUADRATEN
20 POINT 50.50
30 PENDOWN
40 INK 4
50 FOR QU = 1 TO 36
60  QUADRAT (20)
70  TURN 10
80 END FOR QU
100 DEFINE PROCedure QUADRAT(SEITE)
110  FOR S = 1 TO 4
120    MOVE (SEITE)
130    TURN 90
140 END FOR S
150 END DEFine
```

### **DEG(Winkel)**

Funktion zur Umwandlung eines Winkels im Bogenmaß in einen Winkel im Gradmaß.

DEG (PI) ergibt 360 Grad.

### **DELETE microdrive\_Name**

Löscht eine Datei aus dem Inhaltsverzeichnis eines Microdrives.

DELETE MDV2 TestProg

### **DIM Array[,...]**

Definiert einen Array (Feld), wobei Ganze Zahlen (Integer), Gleitpunktzahlen und Zeichenketten erlaubt sind. Die Indizes laufen von 0 bis zum angegebenen Index.

DIM Feld(10,20)

## **DIMN(Feld[Dimension])**

Funktion zum Feststellen der maximalen Größe der Dimension des angegebenen Feldes. Ohne Dimensionsangabe wird 1 angenommen.

**DIM X(10.5.8)**  
PRINT DIMN(X,2) gibt 5 aus.

## **DIR Microdrive**

Anzeigen des Inhaltsverzeichnisses eines Laufwerkes.

**DIR mdv1\_ oder DIR "mdv1\_" oder DIR "mdv\_" 1**

## **Zahl DIV Zahl**

Ganzzahlige Division zweier ganzer Zahlen.

**10 DIV 3 ergibt 3**

## **DLINE Zeilennummer**

Löscht angegebene Zeilennummer

**DLINE TO Zeilennummer**

Löscht vom Anfang bis zu Zeilennummer

**DLINE Zeilennummer TO**

Löscht von Zeilennummer bis zum Ende.

**DLINE Zeilennummer1 TO Zeilennummer2**

Löscht von Zeilennummer1 bis Zeilennummer2.

Beispiele:

**DLINE 20**

**DLINE TO 120**

**DLINE 30 TO 80**

**DLINE 200 TO**

## **EDIT Zeilennummer [,Zwischenraum]**

Aufruf des Zeileneditors. Zeilennummer gibt die zu editierende Zeile an, Zwischenraum gibt die Schrittweite für alle folgenden zu editierenden Zeilen an. Mit den Tasten Cursor nach oben und Cursor nach unten anstatt der ENTER-Taste kann die vorherige bzw. die folgende Zeile editiert werden.

EDIT 50      Editieren von Zeile 50

EDIT 50,10    Editieren von Zeile 50 und allen in  
                  Zehnerschritten folgenden Zeilen.

## **EXEC Proze~ [,...]**

## **EXEC\_N Proze~ [,...]**

EXEC lädt eine Folge von Programmen und führt sie gleichzeitig aus.

Nach dem Start der Prozesse ist wieder Befehlseingabe möglich.

Bei EXEC\_N ist erst nach dem Ende der Prozesse wieder Befehlseingabe möglich.

EXEC MDV1\_Testprog , MDV1\_Drucken

EXEC\_N MDV2\_Datentransfer , MDV2\_Termine

## **EXIT Name**

Erlaubt das Verlassen einer REPeat oder FOR Schleife.

..  
50 FOR A = 1 TO 100

..  
80 IF ... THEN EXIT A  
90 END FOR A

10 REPeat SCHLEIFE

...

70 IF ... THEN EXIT SCHLEIFE  
80 END REPeat SCHLEIFE

## EXP(Zahl)

Berechnet e (=Eulersche Zahl , 2,7182..) hoch die angegebene Zahl.

EXP(2) ist 7.389...

## FILL S

Schaltet das Ausfüllen einer geschlossenen Figur An (mit 1) oder Aus (mit 0). Vor jedem Ausfüllen muß FILL 1 neu angegeben werden.

FILL 1

CIRCLE 40,60,15,.5,PI/6

zeichnet eine ausgefüllte, gedrehte Ellipse.

## FILL\$(Zeichenkette, Zahl)

Funktion zum Vervielfachen eines Zeichens oder einer Zeichenkette aus zwei Zeichen. Es werden "Zahl" Zeichen ausgegeben.

PRINT FILL\$("Hi",8) ergibt HiHiHiHi

## FLASH [Kanal,] S

Bildschirmausgabe mit blinkenden Zeichen. Mit S gleich 1, blinken alle folgenden ausgegebenen Zeichen, mit S gleich 0 wird der Normalmodus eingestellt. Nur möglich bei geringer Auflösung (MODE 8).

Beispiel :

```
10 CLS : INK 6 : MODE 8
20 FOR I = 1 TO 420
30 FLASH RND(1)
40 PRINT CHR$(RND(50 TO 120));
50 END FOR I
```

## FOR x TO y STEP z ... END FOR x

Schleife, die von x bis y mit der Schrittweite z durchlaufen wird.

Andere Möglichkeiten:

FOR x .... END FOR x

FOR x TO y ... END FOR x

Statt END FOR ist auch, wie in anderen BASIC-Versionen NEXT möglich.

Beispiel :

```
10 FOR x = 5,7,9 TO 23 STEP .5
```

```
20 PRINT x
```

```
30 END FOR x
```

5 7 9 9.5 10 10.5 ... 22.5 23

Wenn alle folgenden Anweisungen in derselben Zeile wie FOR .. stehen, ist der Abschluß mit END FOR nicht nötig.

```
FOR i = 1 TO 5 DO : PRINT i
```

```
10 FOR x = 1 TO 15
20 PAUSE 50
30 CLS
40 FOR i = 0 TO PI-PI/x STEP PI/x
50 FOR a = 0 TO 4.5 STEP .5
60 CIRCLE 47,50,10,a,i
70 END FOR a
80 END FOR i
90 END FOR x
```

## FORMAT [Kanal,] Microdrive Bezeichnung

Formatiert eine Kassette, legt ein Inhaltsverzeichnis an und schreibt die Bezeichnung auf das Band. Angezeigt werden die Anzahl der benutzbaren und die Gesamtzahl der verfügbaren Sektoren.

Meistens ist es günstiger, ein Microdrive mehrmals zu formatieren, da dies die Oberfläche verbessert und mehr Kapazität zur Verfügung stellt.

FORMAT MDV1 Spiele

## GOSUB Zeile

Springt zu einem Unterprogramm in der angegebenen Zeile. Mit dem Befehl RETURN im Unterprogramm wird die Verarbeitung bei der Anweisung nach dem GOSUB-Befehl fortgesetzt. GOSUB wird in SuperBASIC nicht benötigt und wird nur wegen der Kompatibilität zu anderen BASIC-Versionen beibehalten.

GOSUB 500
GOSUB 2\*Eingabe

## GO TO Zeile

Unbedingte Verzweigung zu einer anderen Zeile.

GO TO wird in SuperBASIC nicht benötigt und wird nur wegen der Kompatibilität zu anderen BASIC-Versionen beibehalten.

```
GO TO 200
GO TO A - B
```

**IF Ausdruck THEN Befehle [ELSE Befehle]**  
**[END IF]**

Verzweigung nach der Abfrage einer Bedingung.  
Es gibt drei Formen der IF-Anweisung :

IF Ausdruck THEN Anweisungen

IF Ausdruck THEN Anweisungen END IF

IF Ausdruck THEN Anweisungen ELSE Anweisungen END IF

Bei ineinandergeschachtelten IF-Befehlen, wird ELSE und END IF immer zu dem als erstem vorher stehenden IF zugehörig angenommen.

## INK [Kanal,] Farbe

Setzt die Farbe für alle folgenden Ausgabebefehle im angegebenen Fenster fest.

```
INK 0
INK #1,0
```

## Beispielprogramm Kreismuster :

```
10 REMark KREISMUSTER
20 REMark -----
30 FOR SCH = 10 TO 1 STEP -1
40 PAPER SCH * 2 : CLS
50 X = 50 : Y = 60 : D = 40
60 PAPER 0
70 FOR WIN = 0 TO 360 STEP SCH
80 INK 1 + WIN MOD 7
90 BOGEN = WIN*PI/180
100 LINE X,Y TO X-D*SIN(BOGEN),Y+D*COS(BOGEN)
110 END FOR WIN
120 END FOR SCH
```

## INKEY\$ [(Kanal [,Zeit])]

Funktion die ein eingegebenes Zeichen von einem Kanal abfrägt. Dabei kann eine Zeitspanne angegeben werden (in Einheiten zu 50 ms), die die Wartezeit bestimmt.

Wird -1 für die Zeit angegeben, dann wartet INKEY\$, bis ein Zeichen eingegeben wird. Bei 0 kehrt die Funktion sofort wieder zurück.

```
PRINT INKEY$(100)
PRINT INKEY$(#2.70)
PRINT INKEY$
```

## INPUT [[Kanal] Trennzeichen] [Anzeige] [Kanal] [Variable]

Liest Daten, die von der Tastatur eingegeben werden. Das Programm wartet, bis alle Daten eingegeben sind.

```
INPUT "a eingeben ";a
```

## Zeichenkettel INSTR Zeichenkette2

Mit INSTR kann man feststellen, ob eine angegebene Zeichenkette1 in Zeichenkette2 enthalten ist, und an welcher Position sie beginnt.

```
PRINT "ei" INSTR "Meier"  
Das Ergebnis ist 2
```

```
PRINT "pute" INSTR "Computer"  
Das Ergebnis ist 4
```

## INT(Zahl)

Ergibt den ganzzahligen Anteil der angegebenen Gleitpunktzahl.

```
INT(5.78) ist 5.
```

```
10 a = 7.3 : b = 1.1  
20 PRINT INT(a+b)
```

Druckt 8.

## KEYROW (Reihe)

Zeigt an, welche Taste der angegebenen Reihe im Augenblick gedrückt ist.

```
10 PRINT KEYROW(0) : GOTO 10
```

Druckt 1 bei Drücken von F4, 2 bei F1, 4 bei Drücken von 5 usw.

	1	2	4	8	16	32	64	128
--	---	---	---	---	----	----	----	-----

0	F4	F1	5	F2	F3	F5	4	7
1	ENTER	←	↑	ESC	→	\	Leer-taste	↓
2	]	Z	.	C	B	£	M	~
3	[	CAPS LOCK	K	S	F	=	G	;
4	L	3	H	1	A	P	D	J
5	9	W	I	TAB	R	-	Y	O
6	8	2	6	Q	E	Ø	T	U
7	SHIFT	CTRL	ALT	X	V	/	N	,

## LBYTES Gerät, Startadresse

Lädt ein Programm oder eine Datei (abgespeicherter Bildschirminhalt) in den Speicher an der vorgegebenen Adresse.

LBYTES MDV1\_Testprog,62892

## LEN (Zeichenkette)

Funktion die die Länge einer Zeichenkette feststellt.

LEN(A\$)  
LEN ("Dies ist ein Test")

## [LET] Variable = Ausdruck

Dies ist ein Schlüsselwort, das eine Zuordnung angibt.  
Da LET aber nicht unbedingt nötig ist, wird es meist weggelassen.  
Bei Zuordnungen werden Datentypen, die nicht zueinander passen, umgewandelt, wo es möglich ist.

```
LET x = 4 - 3      oder x = 4 - 3  
Za$ = 1223
```

**LINE [Kanal,] Punkt TO Punkt [TO Punkt] ...**  
**LINE\_R [Kanal,] Punkt TO Punkt [TO Punkt] ...**

LINE zieht eine Linie zwischen den angegebenen Punkten.  
Mit Kanal kann ein bestimmtes Fenster angegeben werden.  
LINE\_R zieht eine Linie von der momentanen Stellung des Graphik-Cursors aus.  
Mit einem Befehl können mehrere Linien gezeichnet werden.

Mögliche Befehle :

LINE Punkt Bewegt nur den Graphik-Cursor.  
LINE TO Punkt Linie vom Graphik-Cursor aus zum Punkt.  
LINE Punkt TO Punkt TO Punkt TO ...  
Verbinden von mehreren Punkten mit Linien.

```
LINE 10,10  
LINE TO 20,30  
LINE 10,10 TO 10,30 TO 30,30 TO 30,10 TO 10,10 TO  
     30,30 TO 20,40 TO 10,30 TO 30,10
```

Beispielprogramm Linienmuster :

```
10 REMark LINIENMUSTER  
20 INK 7 : PAPER 0 : CLS  
30 LINE 50,60 TO 50,100  
40 LINE 50,60 TO 10,60
```

```
50 LINE 50,60 TO 50,20
60 LINE 50,60 TO 90,60
70 PAN 3 : SCROLL 3
80 GO TO 30
```

## **LIST [Kanal,] Zeile TO Zeile**

Auflisten von Zeilen eines Programmes.

LIST zeigt das ganze Programm

LIST Zeile  
LIST 20 zeigt eine Zeile.

LIST TO Zeile  
LIST TO 100 zeigt vom Anfang bis Zeile 100.

LIST Zeile TO  
LIST 40 TO zeigt von 40 bis zum Ende.

LIST Zeile TO Zeile  
LIST 100 TO 200 zeigt von Zeile 100 bis Zeile 200.

## **LN (Zahl)**

Natürlicher Logarithmus (zur Basis e) des angegebenen Argumentes.

LN (3.589)

## **LOAD Gerät Programm**

Lädt ein Programm von einem angegebenen Gerät. Dabei wird automatisch der Speicher gelöscht, wie beim NEW-Befehl.

Enthält das zu ladende Programm einen Fehler, so wird das Wort MISTAKE in der Zeile angezeigt.

LOAD MDV1\_Testprog

## **LOCAL Name[,...]**

Definiert Variablen als lokal in einer Funktion oder Prozedur. Diese Variablen können dieselben Namen haben, wie Variable im Hauptprogramm. Diese Anweisung muß ganz am Anfang eines Unterprogrammes stehen.

LOCAL a,b,c,d(4,3)

## **LOG10 (Zahl)**

Logarithmus zur Basis 10 des angegebenen Argumentes.

LOG10(234.2897)

## **LRUN Gerät Programm**

Lädt ein Programm von dem angegebenen Kassettenlaufwerk und startet es. Der Speicher wird vorher gelöscht.

LRUN MDV1\_BOOT  
LRUN MDV2 Kreis

## **MRUN Gerät Programm**

Lädt ein Programm von Kassette, kombiniert es mit dem im Speicher stehenden Programm und führt es aus.

**MRUN MDV2\_Erweit**

## **MERGE Gerät Programm**

Lädt ein Programm von Gerät und kombiniert es mit dem im Speicher stehenden Programm. Zeilen mit gleicher Nummer werden durch das geladene Programm ersetzt, alle anderen werden hinzugefügt.

**MERGE MDV2\_Berech**

## **Zahl MOD Zahl**

Modulo-Funktion, die als Ergebnis den Divisionsrest zweier ganzer Zahlen hat.

7 MOD 3	ist 1
2 MOD 1	ist 0
6 MOD 4	ist 2

## **MODE Zahl**

Setzen des Bildschirmmodus (Bildschirmauflösung).

MODE 256 oder MODE 8 setzt den Bildschirm auf geringe Auflösung.

MODE 512 oder MODE 4 setzt den Bildschirm auf hohe Auflösung.

**MODE 8**

## MOVE [Kanal,] Distanz

Bewegt den Graphik-Cursor im angegebenen Fenster um eine bestimmte Distanz.

MOVE #2,30 Bewegt in Fenster #2 um 30 vorwärts.

MOVE -25 Bewegt um 25 zurück.

## NET Station

Setzen der Stationsnummer im Netzwerk (Standard = 1).

NET 5

## NEW

Löscht alle im Speicher stehenden Programme, Variablen und Kanäle außer 0, 1, 2.

NEW

## NEXT Name

Beendet eine REPeat- oder FOR-Schleife.

```
FOR i = 1 to 10
  PRINT i
NEXT i
```

## ON Variable GOSUB Ausdruck [,...]

Verzweigung zu einem Unterprogramm aufgrund des Wertes einer Variablen.

Kann durch SElect ersetzt werden.

ON a GOSUB 200,350,420

## ON Variable GOTO Ausdruck [,...]

Verzweigung zu bestimmten Zeilen aufgrund des Wertes einer Variablen.

Kann durch SELECT ersetzt werden.

ON zz GOTO 50,100,230

## OPEN\_NEW Kanal, Gerät

Verbindet einen logischen Kanal mit einem reellen Gerät.

OPEN #6,MDV2 Prog1

## OPEN\_IN Kanal, Gerät.

Eröffnet eine vorhandene Datei für Eingabe.

OPEN\_IN #8,MDV2\_Test

## OPEN Kanal, Gerät.

Eröffnen einer neuen Datei für Ausgabe.

OPEN\_NEW #8,MDV1\_Test

## OVER [Kanal,] Schalter

Angabe von Parametern zum Überschreiben des Bildschirm-  
inhaltes mit PRINT. Bleibt bestehen bis zum nächsten  
OVER-Befehl.

OVER 0 Normaler Schreibmodus

OVER 1 Überschreibt mit transparentem Untergrund.

OVER -1 Überschreibt die Bildschirmanzeige.

## PAN [Kanal,] Distanz [,Teil]

Verschiebt das Fenster um die angegebene Anzahl an Bildpunkten nach links (negativer Wert) oder rechts (positiver Wert). Mit Teil wird der Bildschirmabschnitt spezifiziert.

PAN #2,-10

PAN 10

Beispiel :

```
10 CLS
20 CIRCLE 40,50,20
30 FOR I = 1 TO 80 STEP 3
40 PAN I : PAN -I
50 END FOR I
```

## PAPER [Kanal,] Farbe [,Kontrast] [,Muster]

Setzen der Hintergrundfarbe im angegebenen Fenster mit einem bestimmten Muster. Die Farben bleiben bis zum nächsten PAPER-Befehl erhalten.

```
PAPER 5
PAPER #2,1
PAPER #2,0,1
PAPER #2,0,7,3
```

Beispielprogramm :

```
10 REMark PAPER-DEMO
20 MODE 8 : CSIZE 3,1
30 FOR A = 0 TO 7
40 FOR B = 0 TO 7
50 FOR C = 0 TO 3
60 PAPER A,B,C
```

```
70 CLS
80 CURSOR 10.50
90 PRINT "HALLO"
100 PAUSE 10
110 END FOR C
120 END FOR B
130 END FOR A
```

## **PAUSE Verzögerung**

Stoppt ein Programm für die angegebene Zeit. Die Zeit wird in Einheiten zu 20 ms (0.02 s) angegeben. Ohne eine Angabe stoppt das Programm bis eine Taste gedrückt wird.

PAUSE 50 wartet 1 Sekunde

**PEEK(Adresse)**

**PEEK\_W(Adresse)**

**PEEK\_L(Adresse)**

Die Funktion PEEK ermittelt den Inhalt eines Speicherplatzes (1 Byte). Mit PEEK\_W wird ein Speicherwort abgefragt (2 Byte), und mit PEEK\_L ein Doppelwort (4 Byte).

Für PEEK\_W und PEEK\_L muß eine gerade Adresse angegeben werden.

PEEK(64221)

PEEK\_W(2256)

PEEK\_L(12456)

**PENDOWN [Kanal]**

**PENUP [Kanal]**

Bewegt den "Zeichenstift" der Turtle-Graphik. Bei PENDOWN werden die Bewegungen in Linien umgesetzt, bei PENUP wird nichts gezeichnet und nur der "Zeichenstift" bewegt.

**PI**

Funktion, die den Wert von Pi (ca. 3.14159) enthält.

**POINT [Kanal,] Parameter**

**POINT\_R [Kanal,] Parameter**

Zeichnet einen Punkt an der im Graphik-Koordinatensystem spezifizierten Stelle in einem Fenster. Mit POINT\_R werden die Koordinaten auf die jeweilige Stellung des Graphik-Cursors bezogen.

POINT 40,50

POINT #1,10,20

Mehrere Punkte können mit einem POINT-Befehl angegeben werden :

POINT 10,20,30,30,40,40,0,30

**POKE Adresse,Datum**  
**POKE\_W Adresse,Datum**  
**POKE\_L Adresse,Datum**

Ändert einen Speicherinhalt. POKE ändert ein Byte, POKE\_W ändert ein Speicherwort(2 Byte), POKE\_L ändert ein Doppelwort (4 Byte). Bei POKE\_W und POKE\_L muß eine gerade Speicheradresse angegeben werden.

POKE 2234,100  
POKE\_W 4566,20  
POKE\_L 63768,255

**PRINT [Kanal,] Ausdruck [,Trennzeichen]**

Sendet die Ausgabe zum angegebenen Kanal, ohne Kanalangabe zum Bildschirm.

Trennzeichen :

! Drückt ein Leerzeichen vor Ausdruck.  
PRINT ! B\$

, Der nächste Ausdruck kommt nach 8 Leerzeichen.

\ Neue Zeile.

; Der folgende Ausdruck wird direkt angehängt.

**RAD (Zahl)**

Funktion, die das angegebene Gradmaß in Bogenmaß umrechnet.

RAD(90) ist 1.507 (PI/2)

## RANDOMISE [Zahl]

Initialisiert den Zufallszahlengenerator mit der angegebenen Zahl.

RANDOMISE 1567

RANDOMISE

## READ Name [,Name] ...

Liest die Daten aus den DATA-Zeilen hintereinander.  
Siehe auch RESTORE und DATA.

```
10 FOR X = 1 TO 10
20 READ ZAHL : PRINT ZAHL
30 END FOR X
40 DATA 1,67,8,34,90,123,45,12,67,33
```

## RECOL f1,f2,f3,f4,f5,f6,f7,f8

Ändert die Farbe aller Bildpunkte auf dem Bildschirm. Es müssen immer alle Parameter angegeben werden.

- f1 neue Farbe für Schwarz.
- f2 neue Farbe für Blau.
- f3 neue Farbe für Rot.
- f4 neue Farbe für Magenta.
- f5 neue Farbe für Grün.
- f6 neue Farbe für Cyan.
- f7 neue Farbe für Gelb.
- f8 neue Farbe für Weiß.

```
10 CLS : MODE 8
20 FOR I = 0 TO 7
30 INK I : PRINT "*****"***"
```

```
40 PAPER RND(7)
50 END FOR I
60 RECOL 3,0,2,1,7,4,6,5
70 GO TO 20
```

## REMark Text

Alle REMark folgenden Wörter werden vom Programm als Kommentar betrachtet. Verbessert die Lesbarkeit von Programmen.

REMark Programmende

REMark Einlesen von Daten

**RENUM** [Von [TO Bis;] [Anfangszahl],  
[Schrittweite]]

Erlaubt es, ein Programm oder einen Teil davon mit neuen Zeilennummern zu versehen. Ohne Parameter werden alle Zeilennummern des Programms geändert. Die erste Zeile bekommt die Nummer 100, die Schrittweite ist 10.

RENUM 200 TO 400  
neue Zeilennummern von Zeile 200 bis 400 mit Schrittweite 10.

RENUM 10.5 neue Zeilennummern mit Schrittweite 5.

## **REPeat Name ... END REPeat Name**

Befehl zum Erstellen einer Schleife. Mit EXIT wird die Schleife verlassen, und das Programm fährt mit der Verarbeitung nach END REPeat fort.

```
10 x = 0
20 REPeat Drucke
30 x = x + 1
40 PRINT x
50 IF x > 19 THEN EXIT Drucke
60 END REPeat
```

## **RESTORE [Zeilennummer]**

Stellt den Zeiger auf die DATA-Anweisungen der angegebenen Zeilennummer zurück; ohne Zeilennummer auf die erste DATA-Zeile.

```
10 READ A,B,C,D
20 RESTORE
30 READ E,F,G,H
40 DATA 25.78,24.19
50 PRINT A,B,C,D,E,F,G.H
```

Ergebnis :

25 78 24 19 25 78 24 19

## **RETurn [Ausdruck]**

Beendet eine Prozedur oder Funktion und übergibt in einer Funktion den angegebenen Wert als Ergebnis. Das Programm fährt mit der Verarbeitung bei der ersten Zeile nach dem Unterprogramm-Aufruf fort.

## RETRY

Wiederaufnahme eines abgebrochenen Programmlaufes nach der Korrektur eines Fehlers.

## RND ([Zahl] [TO Zahl])

Erzeugt eine Zufallszahl. Ohne Parameter wird eine Zahl zwischen 0 und 1 erzeugt.

RND(X) erzeugt eine Zahl von 0 bis X.

RND(X TO Y) erzeugt eine Zahl zwischen X und Y.

RND(1 TO 100) Zufallszahl zwischen 1 und 100

## RUN [Zeilennummer]

Startet ein BASIC-Programm von einer bestimmten Zeile ab, wenn angegeben, oder vom Anfang.

RUN

RUN 100

RUN 30+50

## SAVE Gerät Name [Zeile] [TO Zeile]

Speichert ein SuperBASIC Programm auf dem angegebenen Gerät.

SAVE ser1

Sichert das ganze Programm nach dem seriellen Kanal.

SAVE MDV1\_Spiel5

Sichert das ganze Programm unter dem Namen Spiel5 auf Microdrive 1.

SAVE MDV2\_Spiel5 10 TO 200

Schreibt das Programm von Zeile 10 bis Zeile 200 unter dem Namen Spiel5 auf Microdrive 2.

SAVE MDV2 Spiel5 40,60,70

Schreibt die Zeilen 40,60,70 unter dem Namen Spiel5 auf Microdrive 2.

### **SBYTES Gerät,Startadresse,Länge**

Speichert Teile des Speichers, angegeben durch Startadresse und Länge auf einem Gerät ab.

SBYTES MDV2\_Testa 60000,5000

### **SCALE [Kanal,] Masstab,x,y**

Ändert den Maßstab für Graphiken im angegebenen Fenster. Standardwert ist 100. Auch der Nullpunkt kann neu festgelegt werden (mit x,y).

SCALE 50.10,0 Maßstab 50 mit dem Nullpunkt auf 10,0

SCALE #2.50,10,0 Angabe in Fenster 2.

### **SCROLL [Kanal,] Zahl [,Teil]**

Verschiebt das über Kanal angegebene Fenster nach oben (Zahl < 0) oder nach unten (Zahl > 0).

Teil = 0 : Ganzer Bildschirm (Standardwert)

Teil = 1 : Fenster ohne letzte Zeile.

Teil = 2 : Fenster ohne erste Zeile.

SCROLL 20 20 Bildpunkte nach unten.

```
10 X = 15 : CLS
20 FOR I = 1 TO 65
30 CIRCLE X,20,10
40 CIRCLE X+32,20,20
60 CIRCLE X+55,45,10
70 X = X + .6
80 SCROLL -1
90 END FOR I
```

**SDATE Jahr,Monat,Tag,Stunde,Minute,Sekunde**

Setzt die interne Uhr des QL auf den angegebenen Wert.

```
SDATE 1984.12.24.1,0,0
```

**SElect ON Variable**

**ON Variable = Auswahlliste**

**Anweisungen**

**[ON Variable] = REMAINDER**

**Anweisungen**

**END SElect**

Auswahl zwischen verschiedenen Fällen. Mit REMAINDER werden alle Fälle abgefangen, die nicht in den davor stehenden Auswahllisten enthalten sind.

Kurze Form :

```
SESelect ON Eingabe = 0 TO 60 : PRINT "Eingabe richtig"
```

Lange Form :

```
10 SESelect ON Monat
20   ON Monat = 1 TO 12
30     PRINT "Richtig"
40   ON Monat = REMAINDER
50     PRINT "Falsch"
60 END SESelect
```

## SEXEC Gerät, Startadresse, Länge, Platz für Daten

Speichert einen Speicherbereich in einer für den EXEC-Befehl ausführbaren Form. Das Programm sollte ein Maschinenprogramm sein.

```
SEXEC MDV2 prog1,67254,2000,300
```

## SIN (Winkel)

Berechnet den Sinus des im Bogenmaß spezifizierten Winkels.

```
PRINT SIN(PI/2) Ergebnis ist 1.
```

## SQRT(Zahl)

Berechnet die Quadratwurzel einer Zahl größer als Null.

```
a = SQRT(b^2 + c^2)
PRINT SQRT (17)
```

## **STOP**

Beendet die Ausführung eines Programmes.

IF a > 5 THEN STOP

## **STRIP [Kanal,] Farbe,2.Farbe,Muster**

Festlegen der Buchstaben-Hintergrundfarbe im angegebenen Fenster.

STRIP 0 Schwarzer Hintergrund

MODE 8

STRIP 7,0,2 Schwarz-Weiß gestreifter Hintergrund

```
10 REMark STRIP-DEMO
20 CLS : MODE 8
30 FOR A = 0 TO 7
40 FOR B = 0 TO 7
50 FOR C = 0 TO 3
60 STRIP A,B,C
70 PRINT "      "
80 END FOR C
90 END FOR B
100 END FOR A
```

## **TAN(Winkel)**

Berechnet den Tangens des im Bogenmaß angegebenen Winkels.

PRINT TAN(PI)

## TURN [Kanal,] Winkel

Bewegt den Kopf der "Turtle" um den in Grad angegebenen Winkel. Mit einer negativen Zahl wird im Uhrzeigersinn bewegt, mit einer positiven entgegen.

TURN -30 Dreht um 30 Grad gegen den Uhrzeigersinn.

## TURNT0 [Kanal,] Winkel

Dreht den Kopf der "Turtle" auf einen bestimmten, im Gradmaß angegebenen Winkel.

TURNT0 10 Bewegt auf 10 Grad.

## UNDER [Kanal,] Schalter

Unterstrichen von Ausgabewörtern.

UNDER 1 schaltet das Unterstrichen ein.

UNDER 0 schaltet das Unterstrichen aus.

10 UNDER 1

20 PRINT "Unterstrichen ";

30 UNDER 0

40 PRINT "oder nicht"

## WINDOW [Kanal,] Breite,Höhe,x,y

Definiert ein Fenster in der angegebenen Größe an den Koordinaten x,y.

WINDOW 200.200.250.0

## 2.2 Programmieren in SuperBASIC

### 2.2.1 Elemente des strukturierten BASIC

Definition von Unterprogrammen :

DEFIne FuNction

...

RETURN ..  
END DEFIne

DEFIne PROCedure

...

RETURN ..  
END DEFIne

Die Unterprogramme ersetzen den Befehl GOSUB und bieten darüberhinaus die Möglichkeit der Rekursion. Der Programmablauf wird übersichtlicher gestaltet und die Programme sind leichter verständlich.

Programmschleifen :

FOR ... [IF .. THEN EXIT ..] ... END FOR

Diese Befehlsfolge wird anstatt einer FOR ... NEXT Schleife verwendet, und hat den Vorteil, daß man mit EXIT die Schleife auf einfache Weise verlassen kann.

REPeat ...

...

IF .. THEN EXIT ..

...

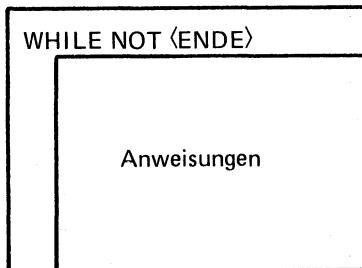
END REPeat

Je nachdem, ob die Abbruchbedingung am Anfang oder am Ende der REPEAT-Schleife abgefragt wird, ergeben sich unterschiedliche Schleifen.

a. Solange eine Bedingung erfüllt ist, werden Anweisungen ausgeführt :

Abfrage am Anfang der Schleife. Entspricht in PASCAL WHILE ... DO

```
REPeat Schleife
  IF <Ende> THEN EXIT Schleife
  <Anweisungen>
END REPeat Schleife
```



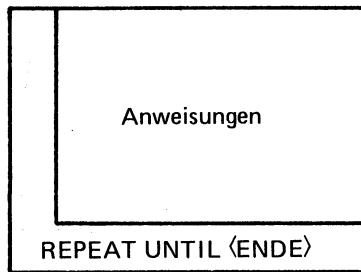
Beispiel :

```
REPeat Lesen
  IF <Ende der Daten erreicht> THEN EXIT Lesen
  <Drucke Daten>
END REPeat
```

b. Wiederhole die Anweisungen, bis eine Bedingung erfüllt ist :

Abfrage am Ende der Schleife. Entspricht in PASCAL REPEAT ... UNTIL

```
REPeat Schleife
  <Anweisungen>
  IF <Ende> THEN EXIT Schleife
END REPeat Schleife
```



Beispiel :

```

REPeat Lesen
  <Drucke Daten>
  IF <Ende der Daten erreicht> THEN EXIT Lesen
END REPeat Lesen

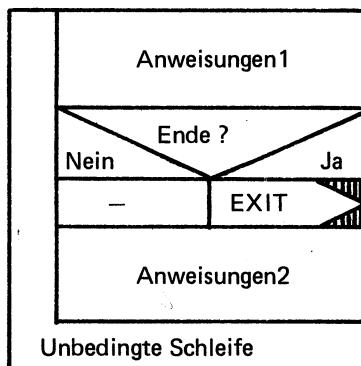
```

c. Schleife mit Unterbrechung zwischen zwei Anweisungs-  
teilen. Abfrage zwischen den Anweisungen

```

REPeat Schleife
  <Anweisungen1>
  IF <Ende> THEN EXIT Schleife
  <Anweisungen2>
END REPeat Schleife

```



Beispiel :

```
REPeat Lesen
  <Drucke Daten>
  IF <Ende der Daten erreicht> THEN EXIT Lesen
    <Berechne Mittelwert>
  END REPeat Lesen
```

Auswahl aus einem von mehreren Fällen. Entspricht in PASCAL dem Befehl CASE .. OF.

```
SElect ON
  ON .. = x
  ...
  ON ...= y
  ...
  .
  .
  .
END SElect
```

Beispiel :

```
SElect ON Monat
  ON Monat = 1
    PRINT "Januar"
  ON Monat = 2
    PRINT "Februar"
  ...
END SElect
```

## 2.2.2 Beispielprogramme

### Programm 23Holz

Das bekannte Spiel mit den 23 Streichhölzern, in Super-BASIC programmiert.

Programmbeschreibung :

Zeile :

20 - 40 : Parameter setzen.

50 - 130 : Titelbild

140 - 230 : Spielbeschreibung

240 : Wertzuweisung an a\$

250 - 560 : Das Spiel wird durchlaufen und wiederholt, bis es durch Eingabe von "N" oder "n" beendet wird. Das verwendete REPeat mit EXIT wird in normalem BASIC mit IF a\$ <> "N" AND a\$ <> "n" THEN GOTO ... programmiert.

280 - 520 : REPeat weg wird verlassen durch Unterprogramm SIEG.

580 - 710 : Unterprogramm SIEG. Durch Übergabe von 1 oder 2 beim Aufruf des Unterprogrammes wird mit SElect ON angezeigt, wer Sieger ist. In BASIC mit GOSUB und IF zu programmieren.

```
10 REMark 23HOLZ
20 CLS : MODE 4 : BORDER 2,3
30 PAPER 2 : INK 4 : OVER 1
40 CSIZE 3,1
50 FOR i = 10 TO 16
60 CURSOR i-10,i
70 IF i = 16 THEN INK 0
80 PRINT "H O F A C K E R "
```

```
90 CURSOR i,i+60
100 PRINT " praesentiert"
110 CURSOR i,i+120
120 PRINT " 2 3 H O L Z"
130 END FOR i
140 PAUSE 150 : PAPER 7 : CLS : CSIZE 1,0
150 PRINT : PRINT
160 PRINT "Wir beginnen mit 23 Haelzern"
170 PRINT "Du faengst an"
180 PRINT "Du kannst 1, 2 oder 3 Haelzer wegn
ehmen"
190 PRINT "Dann nehme Ich 1, 2 oder 3 Haelzer
."
200 PRINT "Wer das letzte Holz wegnehmen muss
"
210 PRINT "    hat v e r l o r e n"
220 PRINT : PRINT "Viel Glueck !"
230 PAUSE 100
240 a$ = "J"
250 REPeat SPIEL
260 IF a$ = "N" OR a$ = "n" THEN EXIT SPIEL
270 m = 23
280 REPeat weg
290 REMark zeichnen
300 CLS
310 x=10 : y = 160
320 FOR a = 1 TO m
330 BLOCK 3,25,x,y,0
340 x = x + 10
350 END FOR a
360 CURSOR 10,100
370 PRINT "Es sind ";m;" Haelzer da"
380 INPUT " Wieviel nimmst du ";h
390 IF h > m OR h < 1 OR h > 3 OR h <> INT(h)
    THEN GO TO 370
400 m = m - h
410 IF m = 0 THEN sieg 1
420 REMark COMPUTER NIMMT
430 IF m = 1 THEN sieg 2
440 r = m-4* INT(m/4)
450 IF r <> 1 THEN GO TO 480
460 c = INT (3 * RND) + 1
470 GO TO 490
```

```

480 c = (r + 3) - 4 * INT((r + 3)/4)
490 m = m - c
500 IF m = 0 THEN sieg 2
510 PRINT " Ich nehme ";c : PAUSE 100
520 END REPeat weg
530 FLASH 0 : MODE 4 : CSIZE 1,0
540 INPUT "Nochmal ( /n) ?";a$
550 END REPeat SPIEL
560 PRINT "Also dann Tschuess " : STOP
570 REMark
580 REMark Unterprogramm SIEG
590 DEFine PROCedure sieg(sieger)
600 CURSOR 0,0 : MODE 8 : FLASH 1 : CSIZE 3,1
610 SElect ON sieger
620 ON sieger = 1
630 PRINT : PRINT "Ich habe "
640 PRINT : PRINT "gewonnen"
650 ON sieger = 2
660 PRINT "Du hast "
670 PRINT "gewonnen"
680 END SElect
690 PAUSE 100
700 EXIT weg
710 END DEFine

```

## Programm NOTIZBUCH

Das Programm NOTIZBUCH benutzt doppelt verkettete Listen. Für die Zeiger R(i) und L(i) werden zwei Felder benutzt. Der Zeiger R(i) zeigt auf das nächste größere Schlüsselwort und L(i) zeigt auf das kleinere Schlüsselwort. Ein Eintrag steht logisch rechts vom aktuellen Eintrag, der andere steht links davon. Im Speicher stehen sie an verschiedenen Plätzen.

Bei der Aufstellung einer doppelt verketteten Liste gibt es fünf Möglichkeiten :

1. Die Liste ist leer. Dann ist P = 0.

2. Es gibt nur einen Eintrag. Dann ist  $P = 1$ ,  $R(1) = 0$  und  $L(1) = 0$ .

3. Der nächste Eintrag  $j$  ist grösser als alle Einträge in der Liste. Dann sind  $R(j) = 0$ ,  $L(j) = k$ ,  $R(k) = j$ .  $K$  ist die Nummer des vorherigen letzten Eintrags.

4. Der nächste Eintrag  $j$  ist kleiner als alle Einträge. Dann sind  $R(j) = p$ ,  $L(j) = 0$ ,  $L(p) = j$  und  $P = j$ .

5. Der nächste Eintrag muß zwischen Eintrag  $k$  und  $i$  gestellt werden. Dann sind  $R(k) = j$ ,  $R(j) = i$ ,  $L(j) = k$  und  $L(i) = j$ .

Ein Schlüsselwort kann aus bis zu 15 Zeichen bestehen, und ein Eintrag kann bis zu 50 Zeichen lang sein. Insgesamt können 40 Einträge gespeichert werden (diese Werte sind in den Zeilen 110 und 120 angegeben).

Die einzelnen Menüpunkte erreicht man durch Eingeben der zugehörigen Zahl.

Mit 1 wird ein neuer Eintrag eingegeben. Auf dem Bildschirm erscheint das Zeichen  $-->$ . Drückt man die rechte Cursor-Taste, dann kann man einen weiteren Eintrag machen. Jede andere Taste veranlaßt das Programm, ins Menü zurückzuspringen.

Um einen Eintrag im Notizbuch zu finden, muß man 2 für "Suche Schlüsselwort", und dann das Schlüsselwort eingeben. Das Schlüsselwort und der Text werden auf dem Schirm angezeigt, und es erscheinen die Zeichen  $<-->$ . Mit den Cursor-Tasten kann man nun im Notizbuch blättern. Mit der rechten Cursor-Taste zum nächsten Schlüsselwort, und mit der linken zum vorigen. Das Drücken einer anderen Taste führt zurück zum Menü.

Die Meldung ENDE DER LISTE wird ausgegeben, wenn der Anfang oder das Ende der Liste erreicht ist.

Die Eingabe von 3 veranlaßt das Programm, eine Datei von Kassette zu Lesen.

Mit 4 wird das Notizbuch auf Kassette abgespeichert. Der Dateiname kann frei gewählt werden.

Die 5 zeigt alle Schlüsselworte auf dem Schirm, mit 6 wird das Inhaltsverzeichnis von MDV1\_ angezeigt.

Die Eingabe von 7 beendet das Programm.

Zeile:

110, 120 : Dimensionen vereinbaren.

130 - 260 : Unterprogramm "Verketten". Bildet die doppelt verketteten Listen.

270 - 350 : Menü.

Ausführung der Menüpunkte durch SElect ON:

360 - 460 : Neuer Eintrag.

470 - 620 : Schlüsselwort suchen.

630 - 750 : Abgespeicherte Daten einlesen.

760 - 860 : Daten auf Kassette abspeichern.

870 - 970 : Anzeigen aller Schlüsselworte.

980 - 1010 : Inhaltsverzeichnis von MDV1\_ anzeigen.

1020 - 1060 : Programmende.

1070 - 1090 : Falsche Eingabe abfangen.

```
100 REMark NOTIZBUCH
110 N=40 : Z = 0 : C = 0
120 DIM N$(N,15),R(N),L(N),E$(N,50)
130 DEFine PROCedure VERKETTEN
140 REMark DOPPELT VERKETTETE LISTEN ERSTELLE
N
150 R(1) = 0 : L(1) = 0 : P = 1
160 FOR J = 2 TO Z
170 I = P
180 IF N$(J) < N$(P) THEN R(J) = P : L(J) = 0
: L(P) = J : P = J: GO TO 240
190 REPeat SCHL
200 K = I : I = R(I)
```

```
210 IF I = 0 THEN R(K) = J : R(J) = 0 : L(J)
= K : EXIT SCHL
220 IF N$(J) < N$(I) THEN R(K) = J : R(J) = I
: L(J) = K : L(I) = J : EXIT SCHL
230 END REPeat SCHL
240 END FOR J
250 C =1
260 END DEFine
270 CLS : PRINT " NOTIZBUCH " : PR
INT \\
280 PRINT "1 = NEUER EINTRAG"
290 PRINT "2 = SUCHE SCHLUESSELWORT"
300 PRINT "3 = LESE DATEI"
310 PRINT "4 = SCHREIBE DATEI"
320 PRINT "5 = ZEIGE SCHLUESSELWORTE"
330 PRINT "6 = INHALTSVERZEICHNIS MDV1_"
340 PRINT "7 = ENDE"
350 PRINT : INPUT "?";I
360 SELECT ON I
370 REMark EINTRAG =====
380 ON I = 1
390 CLS
400 Z = Z + 1
410 INPUT "SCHLUESSELWORT :";N$(Z)
420 INPUT "TEXT :";E$(Z)
430 PRINT : PRINT "-->";
440 C = 0
450 IF CODE(INKEY$(-1)) = 200 THEN GO TO 390
460 GO TO 270
470 REMark SUCHEN =====
480 ON I = 2
490 CLS : IF C = 0 THEN PRINT "VERKETTUNG" :
VERKETTEN
500 I = P
510 IF I = 0 THEN PRINT \I$;" NICHT GEFUNDEN"
: I$ = INKEY$(-1) : GO TO 270
520 INPUT "SCHLUESSELWORT ?";U$
530 FOR I = 1 TO Z: IF N$(I) = U$ THEN GO TO
550 : END FOR I
540 I = 0 : GO TO 510
550 IF N$(I) = U$ THEN CLS : PRINT : PRINT N$(
I):PRINT \E$(I): PRINT \\\"<-- -->":GO TO 57
0
```

```

560 I = R(I) : GO TO 510
570 I$ = INKEY$(-1)
580 IF CODE(I$) = 200 THEN K = I : I = R(I) :
GO TO 610
590 IF CODE(I$) = 192 THEN K = I : I = L(I) :
GO TO 610
600 GO TO 270
610 CLS : IF I = 0 THEN I = K : PRINT \\\"END
E DER LISTE\" : I$ = INKEY$(-1) : CLS
620 PRINT \N$(I) : PRINT\E$(I) :PRINT \\\"<--
-->\" : GO TO 570
630 REMark VON KASSETTE LESEN =====
=====
640 ON I = 3
650 PRINT " KASSETTE IN MDV1_ EINLEGEN !"
660 INPUT " DATEINAME EINGEBEN :" ; D$
670 X$ = "MDV1_" & D$
680 OPEN_IN #5, X$
690 INPUT #5, Z
700 FOR I = 1 TO Z
710 INPUT #5, N$(I)
720 END FOR I
730 FOR I = 1 TO Z : INPUT #5, E$(I) : END FOR
I
740 VERKETTEN
750 GO TO 270
760 REMark AUF KASSETTE SCHREIBEN =====
=====
770 ON I = 4
780 PRINT " KASSETTE IN MDV1_ EINLEGEN !"
790 INPUT " DATEINAME EINGEBEN :" ; D$
800 X$ = "MDV1_" & D$
810 OPEN_NEW #6, X$
820 PRINT #6, Z
830 FOR I = 1 TO Z : PRINT #6, N$(I) : END FOR
I
840 FOR I = 1 TO Z : PRINT #6, E$(I) : END FOR
I
850 CLOSE #6
860 GO TO 270
870 REMark ANZEIGEN =====
880 ON I = 5
890 IF C = 0 THEN CLS : PRINT "VERKETTUNG"\ :

```

## VERKETTEN

```
900 CLS : I = P : K = 0
910 PRINT N$(I) : I = R(I)
920 IF I = 0 THEN GO TO 970
930 IF K < 18 THEN K = K + 1 : GO TO 910
940 PRINT \\\"-->"
950 I$ = INKEY$(-1) : IF CODE(I$) = 200 THEN
CLS : GO TO 910
960 GO TO 270
970 PRINT \\\"ENDE DER SCHLUESSELWORTE" : I$ =
INKEY$(-1) : GO TO 270
980 REMark INHALTSVERZEICHNIS MDV1_ =====
=====
990 ON I = 6
1000 DIR MDV1_
1010 I$ = INKEY$(-1) : GO TO 270
1020 REMark ENDE =====
1030 ON I = 7
1040 PRINT \\\"BEENDEN ( /N)"
1050 INPUT I$ : IF I$ = "N" OR I$ = "n" THEN
GO TO 270
1060 PRINT \\\" E N D E " : STOP
1070 ON I = REMAINDER
1080 PRINT "FALSCHE EINGABE " : I$ = INKEY$(-
1)
1090 GO TO 270
```

## Programm Uhr

"Uhr" ist ein einfaches Programm, das auf dem Bildschirm eine Analog- und eine Digitaluhr zeigt. Es ist für das TV-Format ausgelegt, Sie müssen also nach dem Einschalten des QL die Taste F2 drücken.

Die Zeit wird im 12-Stunden Format eingegeben; in der Form HHMMSS. Zum Beenden CTRL und SPACE drücken.

Zeile :

100 - 160 : Parameter festlegen.

170 - 240 : Uhrzeit auf eingegebenen Wert stellen.  
250 - 360 : Uhr zeichnen.  
370 - 400 : Unterprogramme aufrufen.  
410 - 500 : Unterprogramm Digitaluhr.  
510 - 560 : Unterprogramm Stundenzeiger zeichnen.  
570 - 600 : Unterprogramm Minutenzeiger zeichnen.  
610 - 640 : Unterprogramm Sekundenzeiger zeichnen.  
650 - 710 : Unterprogramm Linie zeichnen.

```
100 REMark QL UHR
110 REMark -----
120 MODE 8 : PAPER 2 : INK 7 : CLS
130 FLASH 1
140 CSIZE 3,1 : AT 1,10 : PRINT "QL UHR";
150 FLASH 0
160 CSIZE 0,0 : INK 5
170 AT 8,7 : PRINT "UHRZEIT EINGEBEN (HHMMSS
)"
180 AT 9,7 : INPUT T$
190 IF LEN(T$) <> 6 THEN GO TO 170
200 HR = T$(1 TO 2)
210 MN = T$(3 TO 4)
220 SE = T$(5 TO 6)
230 OH = HR : OM = MN : OS = SE
240 SDATE 1984,6,3,HR,MN,SE
250 INK 4 : CLS : CSIZE 2,1
260 FILL 0 : T = 0 : CSIZE 0,0
270 INK 4 : FILL 1 : CIRCLE 86,47,35 : FILL 0
280 INK 4,7
290 FOR I = PI/6 TO 2*PI STEP PI/6
300 T = T + 1
310 IF T > 12 THEN GO TO 340
320 X = 32*SIN(I) : Y = 30*COS(I)
330 CURSOR X+85,Y+50,0,0 : PRINT T
```

```
340 END FOR I
350 CIRCLE 86,47,35
360 CSIZE 2,1 : INK 7
370 STDZEI
380 REPeat ANZEIGEN
390 DIGITAL : SEKZEI : STDZEI : MINZEI
400 END REPeat ANZEIGEN
410 DEFine PROCedure DIGITAL
420 LOCal T$
430 T$ = DATE$
440 BEEP 100,10
450 IF T$ = DATE$ THEN GO TO 450
460 HR = T$(13 TO 14) : MN = T$(16 TO 17) : SE = T$(19 TO 20)
470 HR = HR + MN / 60
480 AT 4,0 : PRINT T$(12 TO 20) ;
490 BEEP 100,20
500 END DEFine
510 DEFine PROCedure STDZEI
520 LOCal I
530 H = -1
540 I = OH*PI/6: LINIE I,4 : I = HR*PI/6: LINIE I,7 : OH = HR
550 H = 0
560 END DEFine
570 DEFine PROCedure MINZEI
580 LOCal I
590 I = OM*PI/30 : LINIE I,4 : I = MN*PI/30 : LINIE I,7 : OM = MN
600 END DEFine
610 DEFine PROCedure SEKZEI
620 LOCal I
630 I = OS*PI/30 : LINIE I,4 : I = SE*PI/30 : LINIE I,0 : OS = SE
640 END DEFine
650 DEFine PROCedure LINIE(I,C)
660 LOCal X,Y
670 INK C
680 X = 25*SIN(I) : Y = 24*COS(I)
690 IF H THEN X = 20*SIN(I) : Y = 18*COS(I)
700 LINE 85,50 TO X+85,Y+50
710 END DEFine
```

## Programm Farben ändern

Dieses Programm zeigt, wie man den RECOL-Befehl anwendet, und wie rekursive Unterprogramme definiert werden können. Auf dem Bildschirm werden Quadrate und Kreise dargestellt, und deren Farben geändert. Durch graduelles Anpassen an die Hintergrundfarbe werden die Objekte nacheinander unsichtbar.

### Programmbeschreibung :

#### Zeile :

100 - 150 : Anfangswerte setzen.

160 - 230 : Endlosschleife des Hauptprogrammes.

180 - 210 : Unterprogrammaufrufe mit verschiedenen Parametern.

250 - 330 : Unterprogramm zum Quadrate zeichnen.

280 - 310 : Rekursive Aufrufe mit unterschiedlichen Parametern.

350 - 430 : Unterprogramm zum Kreise zeichnen.

380 - 410 : Rekursive Aufrufe mit verschiedenen Parametern.

450 - 520 : Unterprogramm zum Farbenwechseln.

```
100 REMark FARBEN AENDERN MIT RECOL
110 :
120 REMark FUER TV-FORMAT
130 :
140 WUR = SQRT(2)
150 MODE 8 : PAPER 0 : CLS
160 REPeat HAUPT
```

```
170 FOR ORD = 4 TO 6
180 QUAD 74,50,48,ORD
190 NEUFARB ORD
200 KREIS 80,50,32,ORD
210 NEUFARB ORD
220 END FOR ORD
230 END REPeat HAUPT
240 :
250 DEFine PROCedure QUAD (X,Y,L,A)
260 IF A > 0 THEN
270 INK A: FILL 1:LINE X-L/2,Y-L/2 TO X+L/2,
Y-L/2 TO X+L/2,Y+L/2 TO X-L/2,Y+L/2 TO X-L/2,
Y-L/2 : FILL 0
280 QUAD X-L/2,Y-L/2,L DIV 2,A-1
290 QUAD X+L/2,Y-L/2,L DIV 2,A-1
300 QUAD X+L/2,Y+L/2,L DIV 2,A-1
310 QUAD X-L/2,Y+L/2,L DIV 2,A-1
320 END IF
330 END DEFine
340 :
350 DEFine PROCedure KREIS (X,Y,R,A)
360 IF A > 0 THEN
370 INK A : FILL 1 : CIRCLE X,Y,R : FILL 0
380 KREIS X+R/WUR,Y+R/WUR,R DIV 2,A-1
390 KREIS X-R/WUR,Y+R/WUR,R DIV 2,A-1
400 KREIS X-R/WUR,Y-R/WUR,R DIV 2,A-1
410 KREIS X+R/WUR,Y-R/WUR,R DIV 2,A-1
420 END IF
430 END DEFine
440 :
450 DEFine PROCedure NEUFARB(ENDE)
460 FOR J =1 TO ENDE
470 FOR K = 1 TO 7
480 RECOL 0,2,3,4,5,6,7,1
490 END FOR K
500 RECOL 0,0,1,2,3,4,5,6
510 END FOR J
520 END DEFine
```

## Programm Tonerzeugung

Wenn Sie die Möglichkeiten des QL, Töne zu erzeugen, in allen Variationen ausprobieren wollen, dann ist dieses Programm genau das Richtige für Sie. Mit den Tasten a,s,d,f,h,j (nur Kleinbuchstaben eingeben) können Sie aus folgenden Parametern auswählen :

a Tonhöhe

s Tonhöhe\_2

d Grad\_x

f Grad\_y

g Schwingung

h Verstimmung

j Zufall

Nach dem Starten des Programmes drücken Sie eine dieser Tasten. Daraufhin erscheint die Angabe mit dem Parameter auf dem Bildschirm. Durch Drücken der Tasten Cursor nach oben bzw. Cursor nach unten, wird der Parameter erhöht bzw. erniedrigt. Das Drücken irgendeiner anderen Taste erlaubt es Ihnen, die Änderungen zu stoppen, und mit den oben angegebenen Buchstaben wieder einen anderen Parameter auszuwählen. Durch ENTER wird das Programm beendet.

### Programmbeschreibung :

Zeile :

100 - 150 : Überschrift anzeigen.

160 - 240 : Unterprogramm für Cursor-Tasten.

250 - 290 : Unterprogramm zur Tonerzeugung.  
300 : Anfangswerte festlegen.  
310 - 870 : Schleife für Hauptprogramm.  
320 : Unterprogrammaufruf für ersten Ton.  
330 - 860 : REPeat-Schleife für Änderungen.  
340 : Code der gedrückten Taste speichern.  
350 : Anzeige von Zeile 220 überschreiben.  
360 - 850 : Abfrage der Eingabe mit SElect ON.  
380 - 420 : Tonhöhe ändern.  
440 - 480 : Tonhöhe\_2 ändern.  
500 - 540 : Grad\_x ändern.  
560 - 600 : Grad\_y ändern.  
620 - 660 : Schwingung ändern.  
680 - 720 : Verstimmung ändern.  
740 - 780 : Zufallsparameter ändern.  
800 - 810 : Bei ENTER Programmende.  
830 - 840 : Falsche Taste gedrückt ?

```
100 REMark PROGRAMM TONERZEUGUNG
110 :
120 CLS : BL$ = "      "
130 PRINT " T O N E R Z E U G U N G "
140 AT 2,4 : PRINT "ENTER = Ende"
150 :
160 DEFIne PROCedure aendern
170 X = CODE(INKEY$(-1))
```

```
180 SElect ON X
190 ON X = 208 : vz = 1
200 ON X = 216 : vz = -1
210 ON X = REMAINDER
220 AT 18,5 : PRINT "AUSWAEHLEN" : EXIT NEU
230 END SElect
240 END DEFine
250 :
260 DEFine PROCedure LAUT
270 BEEP 0,A,S,D,F,G,H,J
280 END DEFine
290 :
300 A=100 : S=0 : D=0 : F=0 : G=0 : H=0 : J=0
310 REPeat TON
320 LAUT
330 REPeat NEU
340 PAR = CODE(INKEY$(-1))
350 AT 18,5 : PRINT "
360 SElect ON PAR
370 :
380 ON PAR = 97
390 REPeat AA
400 AT 4,2 : PRINT "a TONHOEHE ";A;BL$
410 aendern : A = A+vz : LAUT
420 END REPeat AA
430 :
440 ON PAR = 115
450 REPeat ss
460 AT 6,2 : PRINT "s TONHOEHE_2 ";S;BL$
470 aendern : S = S+vz : LAUT
480 END REPeat ss
490 :
500 ON PAR = 100
510 REPeat DD
520 AT 8,2 : PRINT "d GRAD_X ";D;BL$
530 aendern : D = D+vz : LAUT
540 END REPeat DD
550 :
560 ON PAR = 102
570 REPeat FF
580 AT 10,2 : PRINT "f GRAD_Y ";F;BL$
590 aendern : F = F+vz : LAUT
600 END REPeat FF
```

```
610 :
620 ON PAR = 103
630 REPeat GG
640 AT 12,2 : PRINT "g SCHWINGUNG " ; G ; BL$
650 andern : G = G+vz : LAUT
660 END REPeat GG
670 :
680 ON PAR = 104
690 REPeat HH
700 AT 14,2 : PRINT "h VERSTIMMUNG " ; H ; BL$
710 andern : H = H+vz : LAUT
720 END REPeat HH
730 :
740 ON PAR = 106
750 REPeat JJ
760 AT 16,2 : PRINT "j ZUFALL " ; J ; BL$
770 andern : J = J+vz : LAUT
780 END REPeat JJ
790 :
800 ON PAR = 10
810 BEEP 1,1 : STOP
820 :
830 ON PAR = REMAINDER
840 AT 18,5 : PRINT "Falsche Taste"
850 END SElect
860 END REPeat NEU
870 END REPeat TON
```

# 3

# Tabellenkalkulation mit ABACUS

## 3.1 Die Befehle

ABACUS ist ein Programm zur Tabellenkalkulation. Die Grundlage ist ein Arbeitsblatt (Spreadsheet), das alle Formeln und Einträge speichert.

Ein Arbeitsblatt besteht aus Zeilen und Spalten. Die Spalten sind durch Buchstaben, die Zeilen sind durch Zahlen gekennzeichnet. Die Breite einer Spalte kann durch Formatierung geändert werden. Durch diese Aufteilung ist das Arbeitsblatt in einzelne Zellen eingeteilt.

In diese Zellen werden nun Einträge gemacht. Diese Einträge können Text, Zahlen oder Formeln sein. Mit Formeln wird ein mathematischer Zusammenhang zwischen einzelnen Zellen angegeben. Das Resultat dieser Berechnung wird als Inhalt dieser Zelle angezeigt. Die Eingabe in das Programm erfolgt als Eintrag einer Zahl in eine Zelle des Arbeitsblattes. Dabei werden die Inhalte aller Zellen, die mit der Eingabezelle durch Formeln verknüpft sind, neu berechnet und auf dem Bildschirm angezeigt. Die Spaltenbezeichnungen gehen von A bis Z, AA bis AZ, und von BA bis BL, umfassen also 64 Spalten. Die Zeilen sind von 1 bis 255 nummeriert.

Legen Sie die Kassette mit ABACUS in MDV1\_ und laden Sie das Programm mit

LRUN MDV1\_BOOT

Lassen Sie die Kassette in Laufwerk 1, da ABACUS von Zeit zu Zeit Programmteile wie zum Beispiel Hilfsinformationen nachlädt.

### Die Funktionstasten

F1 = Help. Hilfe-Taste, die zusätzliche Informationen liefert.

F2 = Prompts. An- und Abschalten der Anzeige des Kontrollbereichs.

F3 = Commands. Schaltet in den Befehlsmodus um.

F4 = Bewegt den Cursor zwischen den beiden Hälften eines geteilten Fensters.

F5 = GOTO. Springt zu einer Speicherzelle.

ESC = Korrektur bei noch nicht vollständig eingegebenen Befehlen.

### Die Befehle

	ECHO	LOAD	QUIT	WINDOW
AMEND	FILES	MERGE	RUBOUT	XECUTE
COPY	GRID	ORDER	SAVE	ZAP
DESIGN	JUSTIFY	PRINT	UNITS	

## A AMEND

Der Inhalt einer Zelle wird in die Eingabezeile kopiert und kann geändert werden.

## C COPY

Kopiert einen Bereich mit Zellen an eine andere Stelle. Sie müssen die Grenzen des Bereiches angegeben (z.B. A1:C5), und die linke obere Ecke der Stelle, an die kopiert werden soll.

## D DESIGN

Mit diesem Befehl können Sie bestimmte Grundwerte von ABACUS ändern:

### Auto-calculate on input.

Automatische Neuberechnung aller Werte. Durch Drücken von "a" schalten Sie diese Option an (YES) oder aus (NO). Falls Sie NO auswählen, dann wird das Arbeitsblatt nur neu berechnet, wenn XECUTE eingegeben wird.

### Blank if zero.

Normalerweise wird in einer Zelle, in der der Wert Null steht, dieser auch angezeigt. Sie können aber auch statt Null eine leere Zelle anzeigen lassen.

### Calculation order.

Mit der Taste "c" können Sie auswählen, ob die Reihenfolge der Berechnungen zeilenweise oder spaltenweise erfolgen soll.

Display 80, 64, 40 columns.

Mit 8, 6 oder 4 kann man das Anzeigeformat auf 80, 64 oder 40 Zeichen pro Zeile festlegen.

Form feed between pages.

Legt fest, ob nach jeder gedruckten Seite ein Formularvorschub erfolgen soll. Der Startwert ist YES.

Gaps between lines on printer.

Setzt den Zeilenvorschub für den Drucker auf 0 (Startwert), 1 oder 2 Zeilen.

Lines per page.

Gibt die Anzahl der Zeilen pro gedruckter Seite an. Der Startwert ist 66, das Maximum ist 255.

Monetary sign.

Sie können statt dem Pfund-Zeichen ein anderes Zeichen eingeben, das bei Geldbeträgen angezeigt wird.

Printer paper width.

Mit diesem Befehl wird die Anzahl der Zeichen pro Zeile für den Drucker festgelegt. Der Startwert ist 80, das Maximum ist 255.

Stationery continuous, single page.

Schaltet um zwischen Endlosformularen und Einzelblättern für den Drucker. CON bedeutet Endlosformular, SIN steht für Einzelblatt.

**E ECHO**

Kopiert die Daten oder Formeln einer Speicherzelle in andere Zellen.

## F FILES

Mit diesem Befehl können Sie ABACUS-Dateien, die bereits auf Mikro-Kassette abgespeichert sind, ändern. Wenn Sie statt des Dateinamens ein Fragezeichen eingeben, dann erhalten Sie eine Liste aller vorhandenen Dateien.

Folgende Befehle stehen zur Auswahl :

### Backup

Erstellt eine Sicherungskopie einer ABACUS-Datei. Sie sollten wichtige Dateien kopieren, um eventuelle Datenverluste zu vermeiden.

### Delete

Löscht eine Datei von Kassette.

### Export

Eine Datei wird in einem Format gesichert (export), das es erlaubt, sie mit ARCHIVE, EASEL oder QUILL zu laden (import).

### Format

Formatiert die Mikro-Kassette in MDV2\_.

### Import

Lädt (imports) eine Datei, die von ARCHIVE oder EASEL erstellt wurde.

## G GRID

Mit diesem Befehl können Sie Änderungen machen, die das ganze Arbeitsblatt betreffen.

### Insert

Einfügen von Zeilen (Rows) oder Spalten (Columns). Sie müssen angeben, wieviele Zeilen oder Spalten Sie einfügen wollen, und die Bezeichnung. Dabei wird das Arbeitsblatt verschoben, und die letzten Zeilen bzw. Spalten gehen verloren.

### **Delete**

Löscht Zeilen oder Spalten aus dem Arbeitsblatt. Für gelöschte Teile werden am Ende wieder neue eingefügt. Alle Formeln werden den neuen Positionen angepaßt.

### **Width**

Dieser Befehl ändert die Breite von Spalten.

### **J JUSTIFY**

Ändert die Justierung von Text oder Zahlen in Zellen. Dabei können Sie angeben, ob Sie bestimmte Zellen ändern wollen (Cells), oder alle Zellen (Default). Die Justierung kann links (mit ENTER), rechts (R), oder zentriert (C) erfolgen.

### **L LOAD**

Lädt eine Datei von MDV2\_. Mit ? statt dem Namen der Datei wird das Inhaltsverzeichnis angezeigt. Falls Sie nichts anderes angeben, wird \_ABA als Erweiterung des Dateinamens angenommen.

### **M MERGE**

Mit Merge (mischen) werden die Daten einer bereits gespeicherten Datei mit den aktuellen kombiniert. Sie können die neuen Daten hinzufügen (ENTER) oder abziehen (S). Zellen, die Text enthalten, werden nicht geändert.

## **O ORDER**

Sortiert Speicherzellen in aufsteigender Reihenfolge.  
Die Sortierfolge ist :

Leere Zellen  
Zellen mit Zahlen, aufsteigend  
Zellen mit Text, alphabetisch.

Bitte nur für Zellen benutzen, die Daten enthalten, da Formeln unbrauchbar werden.

## **P PRINT**

Druckt einen Teil oder das ganze belegte Arbeitsblatt auf dem Drucker aus, der an SER1 angeschlossen ist. Dabei können Sie auswählen, ob Sie Werte oder Formeln ausdrucken wollen, und ob die Zellenangaben auch ausgedruckt werden sollen. Anstatt Auszudrucken, ist es auch möglich, alle Angaben auf einer Datei zu speichern (F eingeben).

## **Q QUIT**

Beendet ABACUS und kehrt zurück zu SuperBASIC.

## **R RUBOUT**

Löscht den Inhalt der angegebenen Speicherzellen.

## **S SAVE**

Speichert den Inhalt des Arbeitsblattes auf MDV2\_. Mit ? erhalten Sie eine Liste aller Dateien. Als Erweiterung wird \_ABA angehängt, wenn Sie nichts anderes angeben.

## **U UNITS**

Ändert das Anzeigeformat von Zahlen in den angegebenen Zellen. Sie können wählen, ob das neue Format für alle neu eingegebenen Werte gelten soll (D eingeben), oder für die bereits vorhandenen (ENTER).

Folgende Formate stehen zur Auswahl :

### **Decimal**

Die Zahlen werden mit der eingegebenen Zahl von Dezimalstellen angezeigt.

### **Exponent**

Die Zahlen werden im wissenschaftlichen Format mit Zehnerpotenzen angezeigt.

### **Percent**

Die Darstellung erfolgt in Prozent.  $1 = 100 \%$ .

### **Integer**

Es wird nur der ganzzahlige Anteil einer Zahl angezeigt. Negative Werte können in Klammern (B eingeben), oder mit Minuszeichen dargestellt werden.

### **General**

Normales Format, das die Zellen am besten ausnutzt.

### **Monetary**

Darstellung mit zwei Stellen hinter dem Dezimalpunkt und Währungszeichen. Mit B werden negative Werte in Klammern gestellt, mit ENTER wird ein Minuszeichen angezeigt. Kann auch für einen Teil der angezeigten Zahlen benutzt werden.

## **W WINDOW**

Mit diesem Befehl kann der Bildschirm in zwei Fenster aufgeteilt werden, die verschiedene Teile des Arbeitsblattes enthalten. Die Aufteilung erfolgt an der Position des Cursors.

Sie müssen eingeben : V für vertikale Teilung, H für horizontale Teilung oder J für das Zusammenlegen von zwei Fenstern.

Wenn sich die Zellen in beiden Fenstern parallel bewegen sollen, dann geben Sie T ein, ansonsten S.

## X XECUTE

Mit Xecute wird das Arbeitsblatt neu berechnet. Der Befehl wird benutzt, wenn die automatische Neuberechnung abgeschaltet ist, oder wenn Sie mit "askn()" oder "asks()" Werte eingeben wollen.

## Z ZAP

Zap löscht das ganze Arbeitsblatt. Da dies irreversibel ist, können Sie den Befehl mit ESC rückgängig machen.

## Die Funktionen

### ABS(x)

Absolutwert der Zahl n.

### ASKN(Text)

Diese Funktion wird zur Eingabe von Zahlen verwendet. Sie muß mit XECUTE aufgerufen werden. Die Zahl wird in die Zelle geschrieben, in der sich ASKN befindet.

**ASKT(Text)**

Wie ASKN, aber statt Zahlen werden Buchstaben eingegeben.

**ATN(x)**

Arkustangens der Zahl x im Bogenmaß.

**AVE(Bereich)**

Berechnet den Durchschnitt (Average) aller Zahlen im angegebenen Bereich.

**CHR(x)**

Zeigt das Zeichen mit dem ASCII-Code x an.

**CODE(Text)**

Gibt die Code-Nummer des ersten Textzeichens im ASCII-Code an.

**COL()**

Nummer der aktuellen Spalte.

**COS(x)**

Kosinus des im Bogenmaß gegebenen Winkels x.

**COUNT(Bereich)**

Anzahl der belegten Zellen im angegebenen Bereich.

### **DATE(x)**

Gibt das Tagesdatum an (sofern vorher in SuperBASIC eingegeben), wobei das Format mit x gewählt wird :

x      Datum

- |   |            |
|---|------------|
| 0 | JJJJ/MM/TT |
| 1 | TT/MM/JJJJ |
| 2 | MM/TT/JJJJ |

### **DAY(S(Text))**

Zeigt die Anzahl der Tage an, die seit dem 1. Januar 1583 bis zum eingegebenen Datum (in der Form "JJJJ/MM/DD") vergangen sind. Z.B. DAY(S("1985/02/24")).

### **DEG(x)**

Wandelt den im Bogenmaß angegebenen Winkel x ins Gradmaß um.

### **EXP(x)**

Berechnet e (Eulersche Zahl, ca. 2.718) hoch x.

### **IF(Ausdruck,Wahr,Falsch)**

Falls die in Ausdruck stehende Bedingung zutrifft, dann wird die Zahl oder der Text der für Wahr steht, angezeigt, ansonsten die Zahl oder der Text für Falsch.

### **INDEX(Spalte,Zeile)**

Zeigt den Inhalt der Zelle an, die im Schnittpunkt von Spalte und Zeile steht.

**INSTR(Wort1,Wort2)**

Gibt die Position des ersten Zeichens von Wort2 in Wort1 an. Die Funktion hat den Wert 0, wenn Wort2 nicht in Wort1 enthalten ist.

**INT(x)**

Ergibt den ganzzahligen Anteil der Zahl x.

**IRR(Bereich,Periode)**

Berechnet eine Verzinsung für die Daten, die im angegebenen Bereich stehen, in der eingegebenen Periode.

**LEN(Text)**

Ergibt die Anzahl der Zeichen in Text.

**LN(x)**

Berechnet den natürlichen Logarithmus (zur Basis e) der Zahl x.

**LOOKUP(Bereich,Verschiebung,Wert)**

Diese Funktion erstellt eine Such-Tabelle. Es werden zwei Tabellen angenommen. Die erste liegt im angegebenen Bereich. Die zweite liegt dazu parallel und ihr Bereich ist mit Verschiebung angegeben. Zu jedem Eintrag in der ersten Tabelle sollte ein dazugehöriger Eintrag in der zweiten Tabelle vorhanden sein. In der ersten Tabelle wird nun nach einem Wert gesucht, der kleiner oder gleich dem angegebenen Wert ist, und das ist der Funktionswert. Die erste Tabelle muß aufsteigend sortiert sein. Der erste Wert der ersten Tabelle ist eine Dummy-Zahl und muß kleiner sein als der folgende.

**MAX(Bereich)**

Sucht den größten Wert in den Zellen im angegebenen Bereich.

**MIN(Bereich)**

Sucht den kleinsten Wert in den Zellen im angegebenen Bereich.

**MONTH(x)**

Ergibt den zur Zahl x gehörenden Monat.

**NPV(Bereich,Prozent,Periode)**

Berechnet den Netto-Gegenwartswert für die Daten im angegebenen Bereich. Prozent ist der jährliche Zinssatz, und Periode ist die Anzahl der Monate, auf die sich die Werte beziehen.

**PI()**

Die Konstante Pi (ungefähr 3.1415)

**RAD(x)**

Rechnet den im Gradmaß gegebenen Winkel x ins Bogenmaß um.

**REPT(Text,n)**

Belegt eine Zelle mit n mal dem ersten Zeichen des Textes.

**ROW()**

Ergibt die Nummer der angesprochenen Zeile.

### **SGN(x)**

Die Signumfunktion hat die Werte 1, -1 und 0, bei  $x > 0$ ,  $x < 0$  und  $x = 0$ .

### **SIN(x)**

Sinus des im Bogenmaß angegebenen Winkels.

### **STR(x,TYP,Dp)**

Konvertiert eine Zahl x in die dazugehörige Zeichenkette, wobei Dp die Stellen nach dem Dezimalpunkt sind, und mit Typ das Format angegeben wird:

- 0      Als Gleitpunktzahl
- 1      Wissenschaftliches Format
- 2      Integer
- 3      Allgemeines Format.

### **SQR(x)**

Quadratwurzel von x.

### **SUM(Bereich)**

Summe aller Zahlen im angegebenen Bereich.

### **TAN(x)**

Tangens von x (x im Bogenmaß).

### **TIME()**

Zeigt die Uhrzeit im Format HH:MM:SS. Die Uhrzeit muß vorher in SuperBASIC gesetzt worden sein.

### VAL(Text)

Konvertiert Text, der aus Ziffern besteht in die dazugehörende Zahl.

### WIDTH()

Ergibt die Breite der angesprochenen Zelle.

## 3.2 Anwendungsbeispiele

Vereinbarungen :

>A1 Zellenzeiger nach A1.

## I. Kassenbuch

Als Beispiel für die Aufstellung eines Arbeitsblattes wird folgende Aufgabenstellung gewählt. Es soll ein Kassenbuch angelegt werden, in welchem das Datum, die Belegnummer, eine Bezeichnung, die Einnahme oder Ausgabe, sowie der Kassenstand angezeigt wird. Das fertige Arbeitsblatt zeigt die Abbildung :

	A	B	C	D	E	F
1:	Datum	Beleg	Bezeichnung	Einnahmen	Ausgaben	Kassenstand
2:						
3:			Uebertrag			
4:						
5:25.09.85	15	Rechng Fa. Kohler		123.75		123.75
6:25.09.85	16	Euerbedarf			12.95	110.80
7:25.09.85	17	Porto			30.80	80.00
8:26.09.85	18	Postscheck		500.00		580.00
9:						
10:						

Spaltenbreite festlegen :

Die Spalte A soll 8 Zeichen, Spalte B 5 Zeichen, Spalte C 20 Zeichen die Spalten D und E 10 Zeichen und die Spalte F 12 Zeichen breit werden.

Der Befehl dafür ist GRID.

>A1 F3 G W 8 ENTER ENTER

grid>width,8 FROM A TO A

>B1	5
>C1	20
>F1	12

Die Spalten D und E sind bereits auf 10 Zeichen eingestellt.

Als nächstes werden die Bezeichnungen eingegeben.

```
>A1 "Datum
>B1 "Beleg
>C1 "Bezeichnung
>D1 "Einnahmen
>E1 "Ausgaben
>F1 "Kassenstand
>C3 "Uebertrag
>D3 rept("=",WIDTH())
>E3 rept("=",WIDTH())
oder F3 Echo cell D3 over range E3
>F3 0

>A2 row=rept("-",WIDTH()-1) * von A2 bis F2 *
>A4 row=rept("-",WIDTH()-1) * von A4 bis F4 *
```

Das Arbeitsblatt sieht nun so aus :

	A	B	C	D	E	F
1	Datum	Beleg	Bezeichnung	Einnahmen	Ausgaben	Kassenstand
2						
3			Uebertrag			
4						
5						
6						
7						
8						
9						
10						

Formeln :

In F5 wird die Eingabe zum Übertrag in F3 addiert, bzw. abgezogen.

>F5 F3+D5-E5

In der Zelle F(N) wird jeweils zum Inhalt der Zelle F(N-1) der Inhalt von D(N) hinzuaddiert oder der Inhalt von E(N) abgezogen. In die Zelle F6 müsste somit diese Formel eingetragen werden :

>F6 F5+D6-E6

Kopiert man diese Formel in die darunter liegenden Zellen, so wird in jeder Zelle ein Wert angezeigt. Bei einer Eingabe erscheint das Ergebnis in allen Zellen. Um dies zu vermeiden, wird ein bedingter Befehl verwendet.

>F5 IF(D5+E5,F3+D5-E5,0)  
>F6 IF(D6+E6,F5+D6-E6,0)

In F5 wird die Eingabe zum Übertrag in F3 addiert. Die Summe D5+E5 ist eine Bool'sche Variable, deren Wert entweder Null, oder ungleich Null ist. Wenn in eine der beiden Zellen ein Eintrag gemacht wird, dann ist der Wert dieser Variablen gleich Eins, die Formel wird berechnet und das Ergebnis wird angezeigt.

Die Formel in F6 wird nun in die Zellen F7 bis F25

kopiert.

>F6 F3 E F6 F7:F25

dabei wird angezeigt :

command> echo,cell F6,over range F7:F25

Mit dem Befehl DESIGN sollten Sie noch

"BLANK if zero "auf " YES "

setzen, damit keine überflüssigen Nullen angezeigt werden.

Das Format von zwei Dezimalstellen hinter dem Komma erreichen Sie durch die Eingabe von :

command> units,cells,decimal,decimal places 2

Mit dem Befehl SAVE wird das Blatt nach beendeter Eingabe gespeichert.

## II. Lineare Regression

Mit Hilfe der linearen Regression wird ein linearer Zusammenhang zwischen einer unabhängigen, und einer davon abhängigen Variablen hergestellt. Diese Variablen sind im allgemeinen Ergebnisse von Beobachtungen oder Experimenten.

Beispiel:

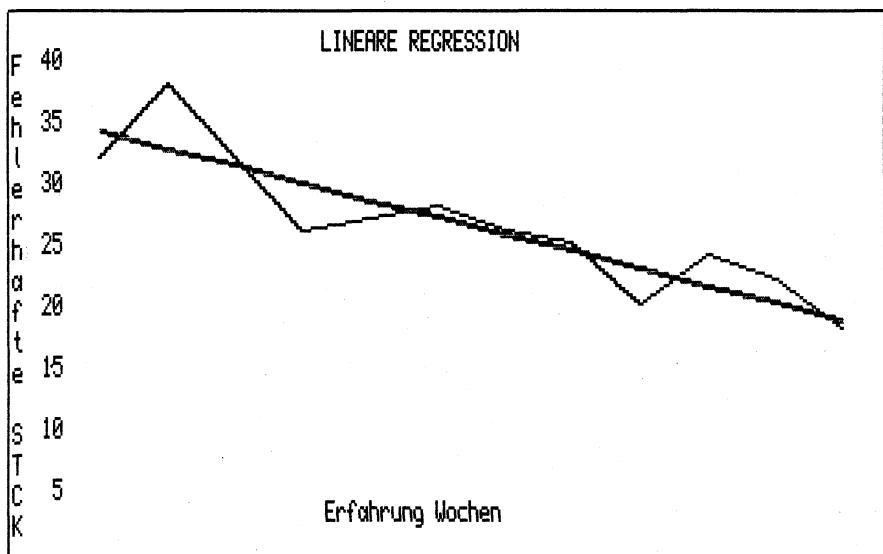
In einer Fabrik wird untersucht, in wieweit die Fehlerrate bei der Herstellung eines Werkstückes von der Erfahrung des Arbeiters abhängt. Das Arbeitsblatt in ABB

II.1 zeigt in der Spalte A die Zahl der Wochen, die ein Arbeiter Erfahrung bei der Herstellung eines Werkstückes hat. In Spalte B ist die Anzahl der fehlerhaften Stücke, die in einer bestimmten Zeit anfallen, eingetragen.

Es wurden folgende Werte ermittelt :

Erfahrung (Wochen)	7	9	6	14	8	12	10
Fehler (Stücke)	26	20	28	16	23	18	24
Erfahrung	4	2	11	1	8		
Fehler	26	38	22	32	25		

Trägt man diese Werte in ein Diagramm ein, so sieht man, daß bei wachsender Erfahrung die Fehlerhäufigkeit abnimmt.



Die Folge der Punkte kann durch die Gerade

$$y = a + bx$$

angenähert werden, wobei sich die Konstanten  $a$  und  $b$  wie folgt errechnen :

$$b = \frac{\sum XY - n\bar{X}\bar{Y}}{\sum X^2 - n\bar{X}^2}$$

und

$$a = \bar{Y} - b\bar{X}$$

Dabei ist  $n$  die Zahl der Beobachtungen, und Summe  $XY$  die abgekürzte Schreibweise für :

$$\sum_{i=1}^n X_i \cdot Y_i, \quad \sum X^2 = \sum_{i=1}^n X_i \cdot X_i$$

$$\bar{X} = \frac{1}{n} \cdot \sum_{i=1}^n X_i \quad \text{und} \quad \bar{Y} = \frac{1}{n} \sum_{i=1}^n Y_i$$

$X$ ,  $Y$  sind die Mittelwerte der Variablen und  $X$  ist das Quadrat des Mittelwertes der  $X$ -Werte.

Mit diesen Werten kann noch eine weitere Kennzahl berechnet werden, der Korrelationskoeffizient. Dieser ist durch

$$r = \frac{\sum XY - n \cdot \bar{X} \cdot \bar{Y}}{\sqrt{\sum X^2 - n\bar{X}^2} \cdot \sqrt{\sum Y^2 - n\bar{Y}^2}}$$

gegeben.

Mit  $r < 0$  besteht ein negativer Zusammenhang zwischen den Beobachtungen, wie in unserem Beispiel. Ist  $r = 0$ , so ist kein Zusammenhang zwischen den Beobachtungen. Ist  $r > 0$ , so besteht ein positiver Zusammenhang (größere Erfahrung, mehr Fehler).

Die Abbildung II.2 zeigt das Arbeitsblatt zur Berechnung der linearen Regression.

```

1| A      | B      | C      | D      | E      | F
1|Lineare Regression
2|
3| Erfahrung      Fehler
4| x-Werte      y-Werte      x*y      x*x      y*y      y-Werte ber
5|   7           26          182       49       676      25.5446194
6|   9           20          180       81       400      22.984375
7|   6           28          168       36       784      27.1445313
8|   14          16          224      196      256      16.0507813
9|   8           23          184      64       529      24.3710938
10|  12          18          216      144      324      18.8242188
11|  10          24          240      100      576      21.5976563
12|  4           26          104      16       676      29.9179688
13|  2           38          76       4       1444     32.6914063
14|  11          22          242      121      484      20.2109375
15|  1           32          32       1       1024     34.078125
16|  8           25          200      64       625      24.3710938
17| Summen :          2048      876      7798
18|Mittelw. x      7.66666667
19|Mittelw. y      24.8333333
20|
21|      r = -0.9084544      n = 12
22|      a = 35.4648438      b = -1.3867188

```

Der Bereich XW (A5..A16) enthält die X-Werte, der Bereich YW (B5..B16) die Y-Werte. In den Bereichen X\*Y, X\*X und Y\*Y werden die entsprechenden Produkte gebildet.

```

>C5 A5*B5
>D5 A5*A5
>E5 B5*B5
>F5 $B22+A5*$E22

```

Diese Formeln werden bis Zeile 16 kopiert.

Die Anzahl der Werte (12) wird in E21 eingetragen:

>E21 12

Die Mittelwerte in C18 und C19 werden mit

>C18 AVE(A5:A16)

>C19 AVE(B5:B16)

berechnet.

In den Spalten C, D und E werden die Summen der eingebrachten Werte errechnet :

>C17 sum(C5:C16)

>D17 sum(D5:D16)

>E17 sum(E5:E16)

Der Korrelationskoeffizient r wird in B21 mit der Formel

>B21 (C17-E21\*C18\*C19)/(sqr(D17-E21\*C18^2)\*  
(sqr(E17-E21\*C19^2))

berechnet.

Der Faktor a wird mit

>B22 C19-E22\*C18

und der Faktor b mit

>E22 (C17-E21\*C18\*C19)/(D17-E21\*C18^2)

berechnet.

Wenn Sie wollen, können Sie den Text mit JUSTIFY noch rechts ausrichten.

### III. Funktionswerte berechnen

Die Aufgabenstellung :

Es sollen die Funktionswerte der Funktion  $Y = \text{ABS}(0.5+\text{SIN}(\text{ABS}(X)))$  im Bereich von -5 bis +5 berechnet werden. Mit dem Graphikprogramm EASEL soll die Kurve graphisch dargestellt werden.

In Spalte C die Spaltenbreite auf 22 setzen:

```
grid>width 22, FROM C TO C
```

Text eintragen:

```
>C1 "Funktionsberechnung  
>C2 "y=abs(0.5+sin(abs(x)))  
>C4 "Schrittweite  
>C5 "Startwert  
>A1 "X-Wert  
>B1 "Y-Wert
```

Werte eintragen :

```
>D4 0.16  
>D5 -5
```

Formeln eintragen :

```
>A2 D5  
>A3 A2+$D4  
D4 muß hier absolut angesprochen werden, da immer der  
Wert in D4 für die Formeln verwendet wird.
```

Übertragen der Formel in A3 in den benötigten Bereich.

```
command>echo,cell A3,over range A4:A70
```

```
>B2 abs(0.5+sin(abs(A2)))
```

Übertragen der Formel in B2 nach B3 bis B70.

command>echo,cell B2,over range B3:B70

Der Anfang des Arbeitsblattes sieht dann so aus :

	A	B	C	D
1	X-Wert	Y-Wert	Funktionsberechnung	
2	-5	0.45892427	$Y=\text{abs}(.5+\sin(\text{abs}(x)))$	
3	-4.84	0.49186876		
4	-4.68	0.49947552	Schrittweite	0.16
5	-4.52	0.48155025	Startwert	-5
6	-4.36	0.43855086		
7	-4.2	0.37157577		
8	-4.04	0.28233591		
9	-3.88	0.17311093		
10	-3.72	0.04669105		
11	-3.56	0.0936943		
12	-3.4	0.2444589		
13	-3.24	0.40175141		
14	-3.08	0.56155372		
15	-2.92	0.71978361		
16	-2.76	0.87239904		
17	-2.6	1.01550137		
18	-2.44	1.145435		
19	-2.28	1.25888071		
20	-2.12	1.35294048		
21	-1.96	1.42521152		
22	-1.8	1.47384763		
23	-1.64	1.49760638		
24	-1.48	1.49588084		
25	-1.32	1.4687151		

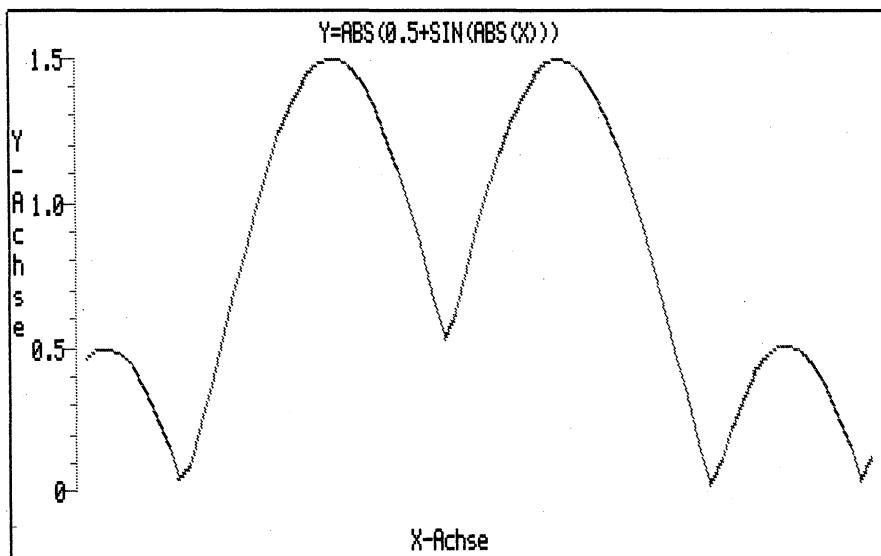
Durch das Ändern der Schrittweite und des Startwertes in den Zellen D4 und D5 können Sie für jeden gewünschten Wert den Funktionswert berechnen.

Exportieren der Funktionswerte in EASEL.

files>export to easel,range B1:B70,by columns,to Funktion1

Es werden nur die Funktionswerte spaltenweise exportiert.

Es ist zu beachten, daß in der ersten zu exportierenden Zeile oder Spalte, ein Wert eingetragen ist, sonst wird die ganze Zeile bzw. Spalte ignoriert.



## **Notizen**

# 4

# Das Datenbanksystem ARCHIVE

## 4.1 Befehle

QL ARCHIVE ist ein Datenbanksystem, mit dem Sie Daten in jeder Form eingeben, abspeichern, sortieren, selektieren und ausgeben können.

In einer speziellen Programmiersprache, die dem Super-BASIC ähnlich ist, können Sie eigene Programme schreiben, und dadurch in kürzester Zeit eigene Ideen verwirklichen.

Was ist denn eigentlich eine Datenbank ?

Eine Datenbank ist grundsätzlich eine Sammlung von Informationen, seien es Zahlen, Wörter, oder Fakten. In der einfachsten Form ersetzt sie einen Stapel Karteikarten, der zum Beispiel Namen und Adressen enthält. Der Unterschied zwischen einer Datenbank und einem Adressenverzeichnis ist nun, daß Sie mit einer Datenbank die gewünschten Informationen in kürzester Zeit erhalten, ohne lange suchen zu müssen, denn das erledigt der Computer für Sie. Die Daten werden in einer Datei gespeichert, und stehen jederzeit zur Verfügung. Sie können nach bestimmten Kriterien sortiert werden, und sind in jedem Format ausdruckbar.

## Starten des Programmes

Legen Sie die Kassette mit ARCHIVE in das Mikrolaufwerk 1 ein, und laden Sie das Programm mit

LRUN MDV1\_BOOT

Sie dürfen die Programm-Kassette nicht herausnehmen, da ARCHIVE von Zeit zu Zeit Programmteile nachladen muß. In Laufwerk 2 muß sich eine formatierte Kassette befinden, zum Speichern der Daten.

### Die Funktionstasten :

F1 = Help. Hilfe-Taste, mit der man zusätzliche Informationen abfragen kann.

F2 = Prompts. An- und Abschalten der Anzeige des Kontrollbereiches.

F3 = Umschalten zwischen den vier Befehlslisten. Im Bildschirmeditor und im Programmeditor: In den Befehlsmodus schalten.

F4 = Einfügen von Text. Nur im Programmeditor.

F5 = Editieren von Text. Nur im Programmeditor.

ESC . Löscht noch nicht vollständig eingebene Befehle.

### Befehle :

all	alter	append	back
backup	close	cls	continue
create	delete	dir	display
dump	edit	endall	endcreate
error	export	find	first
format	if	import	ink
input	insert	kill	last

let	llist	load	local
locate	look	lprint	merge
mode	new	next	open
order	paper	position	print
quit	rem	reset	return
run	save	screen	search
sedit	select	sinput	sload
spooloff	spoolon	sprint	ssave
stop	trace	update	use
while			

Da diese Befehle eine eigene Sprache bilden, müssen sie immer ganz eingegeben werden.

Syntax :

[ ] Parameter in eckigen Klammern sind wahlfrei.

ldn = logischer Dateiname.

Parameter, die zwischen \* \* stehen, können beliebig oft wiederholt werden.

var = Variablenname.

pn = Prozedurname.

## ALL [ldn]

Zeigt alle Datensätze der angegebenen Datei an. Ohne Angabe wird die aktuelle Datei angezeigt.

## ALTER

Zeigt die gegenwärtigen Werte der Variablen an. Die Variablen können geändert werden.

**APPEND [ldn]**

Fügt einen Datensatz zu einer Datei hinzu.

**BACK [ldn]**

Geht um einen Datensatz zurück.

**BACKUP Dateiname1 AS Dateiname2**

Macht eine Kopie von Dateiname1 als Dateiname2.

**CLOSE [ldn]**

Schließt die angegebene Datei.

**CLS**

Löscht die Bildschirmanzeige.

**CONTINUE**

Fährt mit der Verarbeitung des Befehls SEARCH oder FIND beim nächsten Datensatz fort.

**CREATE Name [LOGICAL ldn]: Feld \*[: Feld] \*:****ENDCREATE**

Eröffnet die Datei Name mit den angegebenen Feldern.

**DELETE [ldn]**

Löscht den aktuellen Datenatz.

**DIR [Laufwerk]**

Zeigt das Inhaltsverzeichnis des angegebenen Laufwerkes, oder falls keines angegeben ist, von MDV2\_.

## **DISPLAY**

Zeigt den logischen Dateinamen, eine Liste der Feldbezeichnungen, und die dazugehörigen Werte des aktuellen Datensatzes.

## **DUMP [;Feld] \*[,Feld]\***

Druckt die angegebenen Felder des mit SELECT ausgewählten Datensatzes. Ohne Angabe werden alle Felder ausgedruckt.

## **EDIT**

Aufruf für den Prozeduren-Editor.

## **ERROR pn[;Ausdruck]**

Markiert eine Prozedur für Fehlerfälle. Tritt ein Fehler auf, so wird die Prozedur abgebrochen. Mit der Funktion ERRNUM() kann der Fehler festgestellt werden.

## **EXPORT pdn [;Feld] \*[,Feld]\* [QUILL]**

Schreibt die unter var ausgewählten Felder in einem für ABACUS oder EASEL lesbaren Format auf KASSETTE. Soll die Datei mit QUILL gelesen werden, so ist der letzte Parameter anzugeben.

## **FIND Zeichenkette**

Durchsucht die Datei von Anfang an nach der angegebenen Zeichenkette. Die Suche wird mit CONTINUE fortgesetzt.

## **FIRST [ldn]**

Setzt den Dateizeiger auf den ersten Satz der spezifizierten Datei.

## **FORMAT "Name"**

Formatiert die Mikrokassette in MDV2\_ mit einem frei wählbaren Namen.

## **IF num.Ausdruck: ... [:ELSE : ...] : ENDIF**

Durchführung der IF-Anweisung :

a. Ohne else

Ist der numerische Ausdruck erfüllt, dann werden die folgenden Anweisungen ausgeführt.

b. Mit else

Ist der numerische Ausdruck erfüllt, dann werden die Anweisungen zwischen IF und ELSE ausgeführt. Falls nicht, die Anweisungen zwischen ELSE und ENDIF. Die Verarbeitung wird in beiden Fällen nach ENDIF fortgesetzt.

## **IMPORT Name1 AS Name2 [LOGICAL [dn]]**

Liest die von ABACUS oder ARCHIVE exportierte Datei Name1, und erstellt eine ARCHIVE-Datei Name2. Zusätzlich kann ein logischer Dateiname angegeben werden.

## **INK numerischer Ausdruck**

Setzen der Schriftfarbe.

Schwarz : 0,1

Rot : 2,3

Grün : 4,5

Weiß : 6,7

## **INPUT [var | Zk | Par \*[; var | Zk | Par]\*] [;]**

Eingabe von Tastatur für die angeführten Variablen. Als Par sind AT, TAB, INK, PAPER möglich. INK und PAPER gelten nur für diesen Befehl.

**INSERT**

Hinzufügen eines neuen Datensatzes zu einer Datei.

**KILL Dateiname**

Löscht eine Datei von Kassette.

**LAST [ ldn ]**

Setzt den Dateizeiger auf den letzten Datensatz in der spezifizierten Datei.

**LET Var = Ausdruck**

Zuordnung eines Wertes.

**LLIST**

Druckt alle im Speicher stehenden Prozeduren auf dem Drucker aus.

**LOAD Name**

Lädt die angegebene Prozedur von Kassette.

**LOCAL Var \*[,Var]\***

Definiert die folgenden Variablen als lokal in einer Prozedur.

**LOCATE Ausdruck \*[,Ausdruck]\***

Sucht - in einer vorher ortierten Datei - nach dem angegebenen Ausdruck.

**LOOK Dateiname [LOGICAL ldn]**

Öffnet die genannte Datei zum Lesen.

**LPRINT [Ausdruck| Par \*|[; Ausdruck | Par]\*] [;]**

Ausgabe der Ausdrücke auf Drucker.

**MERGE Dateiname**

Fügt die auf Kassette gespeicherte Prozedur zu der im Speicher stehenden.

**MODE Var1,Var2**

Ändert die Anzeige. Für Var1 können Sie 0 oder 1 angeben, wobei 0 die Bildschirmunterteilung in einzelne Bereiche aufhebt, und den ursprünglichen Zustand wieder herstellt. Var2 definiert die Anzahl der Zeichen pro Zeile (4, 6, oder 8 = 40, 64, oder 80 Zeichen). Der Anfangswert bei einem Monitor ist MODE 1,8.

**NEW**

Löscht alle Daten im Speicher und schließt alle noch offenen Dateien.

**NEXT [ldn]**

Setzt den Dateizeiger auf den nächsten Satz.

**OPEN Dateiname [LOGICAL ldn]**

Eröffnet eine Datei zum Lesen und Schreiben.

**ORDER Feld; ald \*[,Feld ; ald]\***

Sortiert die Sätze der Datei nach den angegebenen Feldern in aufsteigender (a) oder absteigender (d) Reihenfolge. Zum Sortieren werden nur die ersten acht Zeichen berücksichtigt.

**PAPER Zahl**

Setzen der Hintergrundfarbe.

**POSITION Zahl**

Nimmt den mit Zahl spezifizierten Datensatz als aktuellen Datensatz.

**PRINT [Ausdruck | Par ] \*[; Ausdruck| Par]\* [;]**

Zeigt die Werte der angegebenen Felder auf dem Bildschirm an.

**QUIT**

Schliesst alle Dateien und beendet ARCHIVE.

**REM**

Alles was nach REM steht wird als Kommentar betrachtet, und bei der Programmausführung ignoriert.

**RESET**

Stellt die Datensätze wieder her, die durch SELECT nicht benutzt wurden. Dabei wird die Sortierung zerstört.

## **RETURN**

Beendet eine Prozedur.

## **RUN Name**

Lädt die unter "Name" abgespeicherten Prozeduren und führt die "start" genannte Prozedur aus.

## **SAVE Name**

Schreibt alle Prozeduren im Speicher unter einem Namen auf Kassette.

## **SCREEN**

Zeigt den mit SLOAD geladenen Bildschirm-Layout an.

## **SEARCH Num.Ausdruck**

Sucht vom Anfang einer Datei nach einem Datensatz, in dem der numerische Ausdruck erfüllt ist.

## **SEdit**

Aufruf des Bildschirm-Editors.

## **SELECT Num.Ausdruck**

Durchsucht die ganze Datei nach Datensätzen, bei denen der numerische Ausdruck zutrifft. Die Datei verhält sich dann so, als ob nur die ausgewählten Sätze vorhanden wären.

**SINPUT Var \*[,Var]\***

Wartet auf Eingabe für die angegebenen Variablen in der spezifizierten Reihenfolge. Alle Variablen müssen im Bildschirm-Layout enthalten sein.

**SLOAD Name**

Lädt einen vorher definierten und abgespeicherten Bildschirm-Layout.

**SPOOLOFF**

Abschalten des Befehles SPOOLON.

**SPOOLON Name [EXPORT]**

Die Ausgabe der Befehle LPRINT, LLIST und DUMP wird auf Kassette gespeichert, und nicht gedruckt. Die Steuerzeichen für den Drucker werden auch übertragen. Durch EXPORT wird die Datei lesbar für QUILL.

**SPRINT**

Zeigt die Felder des aktuellen Datensatzes im definierten Bildschirm-Layout an.

**SSAVE Name**

Speichert die momentane Bildschirmanzeige als Bildschirm-Layout auf Kassette ab.

**STOP**

Beendet die Ausführung aller Prozeduren.

## **TRACE**

Schaltet den Trace-Modus ein bzw. aus. Im Trace-Modus wird die ausgeführte Programmzeile angezeigt und kann mit der Leertaste angehalten werden. Die erneute Eingabe von Trace schaltet den Modus wieder ab.

## **UPDATE [ ldn ]**

Ersetzt den aktuellen Datensatz in der angegebenen Datei, durch die momentanen Werte der Feld-Variablen.

## **USE ldn**

Nimmt die angegebene Datei als aktuelle Datei.

## **WHILE Num.Ausdruck : ... : ENDWHILE**

Wiederholt die Anweisungen zwischen WHILE und ENDWHILE, solange bis der Wert des numerischen Ausdruckes ungleich Null ist.

## **FUNKTIONEN**

### **Syntax :**

**x** = Numerischer Ausdruck.

**a** = Ausdruck mit Zeichenkette (String).

### **ABS(x)**

Absolutbetrag von **x**.

**CHR(x)**

Gibt das ASCII-Zeichen mit der Nummer x an.

**CODE(a)**

Gibt die Nummer des ersten Zeichens von a an.

**DATE(x)**

Gibt das Tagesdatum (vorher in SuperBASIC setzen) in einem der drei Formate an:

x	Datum
0	JJJJ/MM/DD
1	TT/MM/JJJJ
2	MM/TT/JJJJ

**DAY(S(JJJJ/MM/TT))**

Gibt die Anzahl der Tage an, die vom 1. Januar 1583 bis zum angegebenen Datum vergangen sind.

**EOF([ldn])**

Zeigt an, ob Sie versucht haben, nach dem Dateiende noch weiterzulesen. Der Funktionswert ist dann 1, ansonsten ist er 0.

**ERRNUM()**

Gibt die Nummer des zuletzt aufgetretenen Fehlers an.  
Null heißt kein Fehler.

### **FIELDT(x[,ldn])**

Zeigt den Typ des durch die Nummer angegebenen Feldes an.  
Der Funktionswert ist 0, bei einem numerischen Feld,  
ansonsten ist er 1.

### **FIELDV(x[,ldn])**

Der Wert dieser Funktion ist der Inhalt des mit der  
Nummer angegebenen Feldes.

### **FOUND()**

Falls ein Datensatz mit SEARCH oder FIND gefunden wurde,  
ist der Wert dieser Funktion 1, ansonsten ist er 0.

### **GETKEY()**

Wartet auf einen Tastendruck. Der Funktionswert ist die  
gedrückte Taste.

### **INKEY()**

Falls zur Zeit des Funktionsaufrufes eine Taste gedrückt  
wurde, ist diese der Wert der Funktion.

### **INSTR(Text1,Text2)**

Gibt die Position von Text2 in Text1 an. Ist Text2 nicht  
in Text1 enthalten, dann ist der Wert 0.

### **INT(x)**

Ganzzahliger Anteil der Zahl x.

**LEN(a)**

Anzahl der Zeichen in a.

**LOWER(a)**

Konvertiert den Text a in Kleinbuchstaben.

**MEMORY()**

Anzahl der freien Bytes im Speicher.

**MONTH(x)**

Ergibt den Monat, der zur Zahl x gehört.

**NUMFLD([ldn])**

Anzahl der Felder in den Datensätzen der angegebenen Datei.

**RECNUM([ldn])**

Nummer des aktuellen Datensatzes der spezifizierten Datei (von 0 ab gezählt).

**REPT(a,x)**

Wiederholt die unter a angegebene Zeichenkette x mal.

**SGN(x)**

Die Signumfunktion hat die Werte -1, 0, +1 bei  $x < 0$ ,  $x = 0$ ,  $x > 0$ .

## **STR(x, Typ, Dp)**

Konvertiert die Zahl x in eine Zeichenkette. Typ gibt das Format der konvertierten Zahl an :

- 0 Gleitpunktzahl
- 1 Exponentielle Darstellung
- 2 Ganze Zahl
- 3 Allgemeines Format

Mit Dp wird die Anzahl der Stellen hinter dem Dezimalpunkt angegeben.

## **SQR(x)**

Quadratwurzel aus x.

## **TIME()**

Gibt die Uhrzeit im Format "HH:MM:SS" an.

## **UPPER(a)**

Wandelt die mit a angegebene Zeichenkette in Großbuchstaben um.

## **VAL(a)**

Wandelt den mit a gegebenen Text, der sich aus Ziffern zusammensetzt, in eine Zahl um.

## Der Programmeditor

Zum Eingeben von Prozeduren ist ein spezieller Editor vorhanden, der folgende Möglichkeiten bietet:

- Eine Prozedur eingeben
- Eine andere Prozedur bearbeiten
- Eine Zeile ändern (F5)
- Eine Zeile einfügen (F4)
- Eine Zeile löschen (C für Cut)
- Die gelöschte Zeile einfügen (P für Paste)
- Eine Prozedur löschen (D für Delete)
- Eine neue Prozedur eingeben (N für New)

Der Editor wird aufgerufen durch

EDIT

Auf dem Bildschirm erscheint ein neues Bild, mit einem roten Balken auf der linken Seite, und dem Wort

proc ""

In die Hochkommas müssen Sie den Programmnamen eintippen, zum Beispiel

proc "Test"

Nun wird

```
proc "Test"  
endproc
```

angezeigt.

Jede neue Zeile Ihrer Prozedur wird zwischen diese beiden Zeilen eingefügt.

## Eine andere Prozedur bearbeiten

Wenn Sie bereits mehrere Prozeduren eingegeben haben, und Sie möchten eine andere als die Aktuelle editieren, dann können Sie mit den Tasten TABULATE und SHIFT TABULATE aus der angezeigten Liste den Namen der gewünschten Prozedur auswählen. Diese wird dann die neue aktuelle Prozedur, und wird bei einem erneuten Aufruf mit EDIT angezeigt.

## Eine Zeile editieren

Wenn Sie eine Zeile ändern wollen, dann wählen Sie mit den Tasten "Cursor nach oben" und "Cursor nach unten" die Zeile aus, und Drücken F5. Nach Abschluß der Änderungen müssen Sie ENTER eingeben, und die Zeile wird übernommen.

## Eine Zeile einfügen

Mit der Taste F4 wird eine neue Zeile nach der aufgehellten Zeile eingefügt. Sie müssen also vor dem Drücken von F4 mit den Tasten "Cursor nach oben" und "Cursor nach unten" die Position auswählen.

## Zeilen löschen (Cut)

Wählen Sie mit den Cursor-Tasten die gewünschte Zeile aus, drücken Sie F3 und dann C. Wenn Sie ENTER drücken, dann wird diese Zeile gelöscht. Sie können aber auch durch Cursor-Bewegungen einen Abschnitt markieren, und dann ENTER drücken, um den ganzen Abschnitt zu löschen. Der gelöschte Text wird gespeichert, und kann an anderer Stelle mit "Paste" eingefügt werden.

## Zeilen einfügen (Paste)

Text, der mit Cut gelöscht wurde, wird mit P (Paste) wieder eingefügt. Dazu wählen Sie mit TABULATE und SHIFT TABULATE die Prozedur aus, in welche Sie einfügen wollen.

Dann müssen Sie nur die Zeile markieren (mit den Cursor-Tasten), nach welcher der Text eingefügt werden soll, und F3 und P drücken.

#### Eine Prozedur löschen (Delete)

Sie wählen mit TABULATE die gewünschte Prozedur aus, drücken F3, D und dann ENTER, und die Prozedur wird gelöscht. Falls Sie versehentlich etwas Falsches eingegeben haben, drücken Sie einfach statt ENTER eine andere Taste, und Sie kehren zurück ins Menü.

#### Eine neue Prozedur eingeben (New)

Um eine neue Prozedur einzugeben, drücken Sie F3 und N. Diese neue Prozedur erscheint dann bei der Eingabe von EDIT als aktuelle Prozedur.

Auf Kassette geschrieben werden alle im Speicher stehenden Prozeduren mit

SAVE "Prozedurname"

Laden können Sie die gespeicherten Prozeduren mit

LOAD "Prozedurname"

#### Der Bildschirm-Editor (Screen Editor)

Mit dem Bildschirm-Editor können Sie einen Bildschirm-Layout entwerfen, oder ändern. Der Editor wird aufgerufen mit

SEDITION

Ein Bildschirm-Layout besteht aus zwei Teilen, dem festen Text, und den variablen Werten. Mit dem Befehl SCREEN wird der Text angezeigt, und mit SPRINT werden die aktuellen Variablen eingetragen.

Durch Drücken der Taste F3 kommen Sie in den Befehlsmodus und können folgende Befehle eingeben:

- C Bildschirm löschen
- V Bereich für eine Variable markieren
- I Textfarbe festlegen
- P Hintergrundfarbe festlegen

Ein fertiggestellter Bildschirm-Layout wird mit SSAVE auf Kassette abgespeichert, und kann mit dem Befehl SLOAD wieder geladen werden.

## 4.2 Adressenverwaltung unter ARCHIVE

Als Beispiel für die Programmierung von Prozeduren möchte ich Ihnen zeigen, wie man eine Adressenverwaltung unter ARCHIVE programmiert.

Die benötigten Felder werden mit dem Befehl CREATE erstellt.

CREATE "Adressen"

Anrede\$  
Vorname\$  
Name\$  
Strasse\$  
PLZ  
Ort\$

ENDCREATE

Mit INSERT können Sie Adressen eingeben. Nach jeder fertig eingetippten Adresse müssen Sie F5 drücken, um diese zu speichern. Mit ESC wird der Befehl unterbrochen. Geben Sie am Ende CLOSE ein, damit die eingegebenen Datensätze auf Kassette gespeichert werden.

Rufen Sie den Programm-Editor auf (mit EDIT) und geben Sie die folgenden Prozeduren für die Adressenverwaltung ein. Am Ende verlassen Sie mit ESC den Editor, und sichern alle Prozeduren mit SAVE (z.B. SAVE "MENUE").

## I. Das Menü

Aus dieser Prozedur werden alle folgenden Prozeduren aufgerufen. Die Datei "Adressen" wird zu Beginn eröffnet, und beim Programmende wieder geschlossen.

```
proc MENUE
  let Y$="?"
  open "ADRESSEN"
  while Y$<>"6"
    cls
    print "MENUE ADRESSENVERWALTUNG"
    print
    print "1 ADRESSENLISTE DRUCKEN"
    print "2 ADRESSEN SORTIEREN"
    print "3 ADRESSEN SELEKTIEREN"
    print "4 DATENSAETZE ANZEIGEN UND AENDERN"
    print "5 ADRESSEN-AUFKLEBER DRUCKEN"
    print "6 ENDE"
    print
    print "SIE WUENSCHEN ?";
    let Y$=getkey()
    if Y$=="1": DRUCKEN: endif
    if Y$=="2": SORTIERE: endif
    if Y$=="3": SELEKT: endif
    if Y$=="4": ZEIGEN: endif
    if Y$=="5": AUFKLEB: endif
  endwhile
  close
endproc
```

Die Bildschirmanzeige sieht so aus :

MENUE ADRESSENVERWALTUNG

- 1 ADRESSENLISTE DRUCKEN
- 2 ADRESSEN SORTIEREN
- 3 ADRESSEN SELEKTIEREN
- 4 DATENSAETZE ANZEIGEN UND AENDERN
- 5 ADRESSEN-AUFKLEBER DRUCKEN
- 6 ENDE

SIE WUENSCHEN ?

II. Drucken der Adressenliste.

```
proc DRUCKEN
    lprint "ADRESSENLISTE"
    lprint
    lprint
    first
    let Z=0
    while not eof()
        lprint "Adresse Nr."; Z
        lprint
        lprint ANREDE$+" ";
        lprint VORNAME$+" "+NAME$+" ";
        lprint STRASSE$+" ";
        let PLZ$=str(PLZ, 2, 0)
        lprint PLZ$+" "+ORT$+
        lprint
        lprint
        next
        let Z=Z+1
    endwhile
endproc
```

## Ein Beispiel für die Adressenliste :

### ADRESSENLISTE

Adresse Nr.0

Herr Franz Muxeneder Aachener 12 8000 Muenchen 100

Adresse Nr.1

Frau Karla Wege Ziegenweg 21 7000 Stuttgart 1

Adresse Nr.2

Frl Maria Braun Waldweg 33 2250 Hameln

Adresse Nr.3

Herr Franz Karl Waldweg 22 8850 Holzkirchen

### III. Adressen sortieren.

Sie können alle Adressen nach einem Feld sortieren.

```
proc SORTIERE
  cls
  print "SORTIEREN NACH"
  print
  print "1 ANREDE"
  print "2 VORNAME"
  print "3 NAME"
  print "4 STRASSE"
  print "5 PLZ"
  print "6 ORT"
  print
  let X#=upper(getkey())
```

```
if X$=="1": order ANREDE$; A: endif
if X$=="2": order VORNAME$; A: endif
if X$=="3": order NAME$; A: endif
if X$=="4": order STRASSE$; A: endif
if X$=="5": order PLZ; A: endif
if X$=="6": order ORT$; A: endif
cls
display
endproc
```

Die Bildschirmanzeige :

SORTIEREN NACH

- 1 ANREDE
- 2 VORNAME
- 3 NAME
- 4 STRASSE
- 5 PLZ
- 6 ORT

#### IV. Selektieren von Adressen.

Diese Prozedur wählt nach Ihren Angaben bestimmte Adressen aus der gesamten Liste aus (zum Beispiel alle mit der Anrede "Herr"). Für alle folgenden Operationen sind die anderen Adressen nicht mehr ansprechbar. Wollen Sie wieder mit allen Adressen arbeiten, dann geben Sie RESET ein (R unter Menüpunkt 4.)

```

proc selekt
  cls
  print "Selektieren nach"
  print
  print "1 Anrede"
  print "2 Vorname"
  print "3 Name"
  print "4 Strasse"
  print "5 PLZ"
  print "6 Ort"
  let x$=upper(getkey())
  if x$=="1": cls : input "ANREDE :"; A$: select ANREDE$=A$: endif
  if x$=="2": cls : input "VORNAME :"; A$: select VORNAME$=A$: endif
  if x$=="3": cls : input "NAME :"; A$: select NAME$=A$: endif
  if x$=="4": cls : input "STRASSE :"; A$: select STRASSE$=A$: endif
  if x$=="5": cls : input "PLZ :"; A: select PLZ=A: endif
  if x$=="6": cls : input "ORT :"; A$: select ORT$=A$: endif
  cls
  display
endproc

```

Die Bildschirmanzeige :

```

Selektieren nach

1 Anrede
2 Vorname
3 Name
4 Strasse
5 PLZ
6 Ort

```

## V. Datensätze anzeigen und ändern.

Mit dieser Prozedur können Sie die wichtigsten Operationen durch Drücken einer einzigen Taste ausführen.

```

proc zeigen
  let x$="?"
  while x$<>"B"
    cls

```

```
print "Sie koennen eingeben :"
print
print "E Erster Datensatz"
print "N Naechster Datensatz"
print "V Vorheriger Datensatz"
print "L Letzter Datensatz"
print "* Inhaltsverzeichnis MDV2_"
print "A Datensatz aendern"
print "B Beenden der Prozedur"
print "R Reset - Aufheben der Selektion"
let x$=upper(getkey())
if X$=="E": first : display : endif
if X$=="N": next : endif
if X$=="V": back : endif
if X$=="L": last : endif
if X$=="*": dir "MDV2_": let Z$=getkey(): endif
if X$=="A": alter : endif
if X$=="R": reset : endif
display
sprint
let Z$=getkey()
endwhile
endproc
```

Sie haben folgende Möglichkeiten :

Sie koennen eingeben :

E Erster Datensatz  
N Naechster Datensatz  
V Vorheriger Datensatz  
L Letzter Datensatz  
\* Inhaltsverzeichnis MDV2\_  
A Datensatz aendern  
B Beenden der Prozedur  
R Reset - Aufheben der Selektion

## VI. Adressen-Aufkleber drucken.

Diese Prozedur druckt Ihnen die gewünschte Anzahl an Aufklebern. Es werden immer zwei Adressen nebeneinander gedruckt.

```
proc AUFKLEB
  cls
  print "AUFKLEBER DRUCKEN"
  print
  print "VORHER DIE GEWUENSCHTE ADRESSE AUSWAEHLEN"
  print
  input "ANZAHL ?"; A
  let X=1
  while X<=A
    let A=A-2
    lprint ANREDE$; : lprint tab 40; ANREDE$
    lprint VORNAME$+" "+NAME$; : lprint tab 40; VORNAME$+" "+NAME$
    lprint STRASSE$; : lprint tab 40; STRASSE$
    lprint PLZ; : lprint " "+ORT$; : lprint tab 40; PLZ; : lprint " "+ORT$
    lprint : lprint : lprint
  endwhile
endproc
```

Die Bildschirmanzeige :

AUFKLEBER DRUCKEN

VORHER DIE GEWUENSCHTE ADRESSE AUSWAEHLEN

ANZAHL ?

Beipielausdruck :

Herr  
Franz Karl  
Waldweg 22  
8850 Holzkirchen

## Entwerfen des Bildschirm-Layouts

Sie rufen den Bildschirmeditor auf mit SEDIT. Auf dem Bildschirm wird folgendes angezeigt :

```
logical name : main
Anrede$      : .....
Vorname$      : .....
Name$        : .....
Strasse$     : .....
PLZ          : .....
Ort$         : .....
```

Den neuen Text können Sie einfach an der gewünschten Stelle eintippen. Die Variablen müssen Sie, um sie an anderer Stelle eingeben zu können, vorher löschen. Bewegen Sie den Cursor auf einen der Punkte, an denen die Variable definiert ist, und drücken die Tasten :

F3  
V  
ENTER

Nun bewegen Sie den Cursor an den neuen Platz der Variablen, und geben ein :

F3  
V  
<Name der Variablen, z.B. Vorname\$ >  
<Länge der Variablen in Punkten .....>  
ENTER

Der fertige Bildschirm-Layout kann dann so aussehen:

```
ADRESSEN
Anrede : .....
Vorname : ..... Name : .....
Strasse : .....
PLZ : .... Ort : .....
```

Durch ESC verlassen sie den Bildschirmeditor, und mit  
SSAVE "AdreBild" speichern Sie den eingegebenen Layout  
ab.

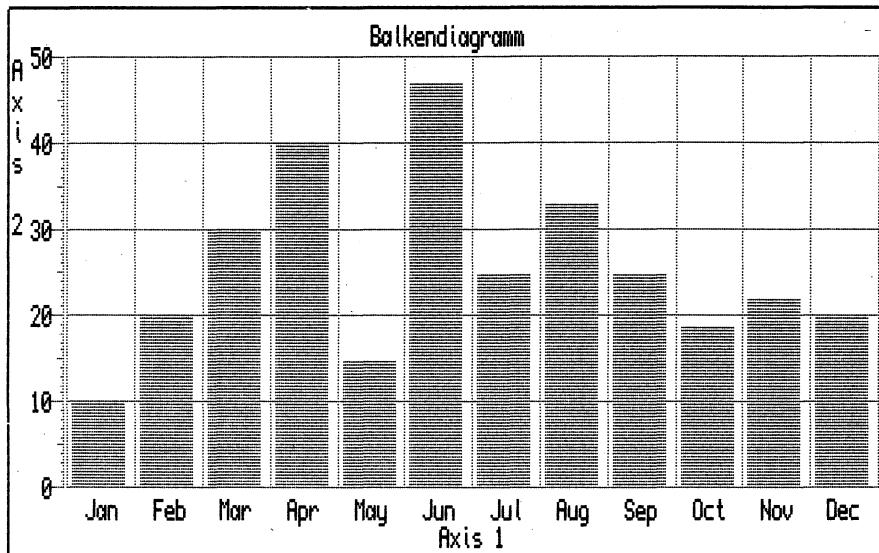
## Notizen

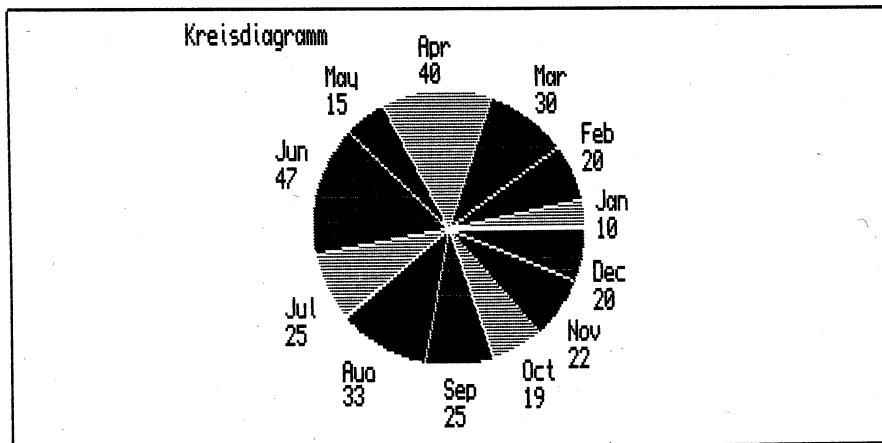
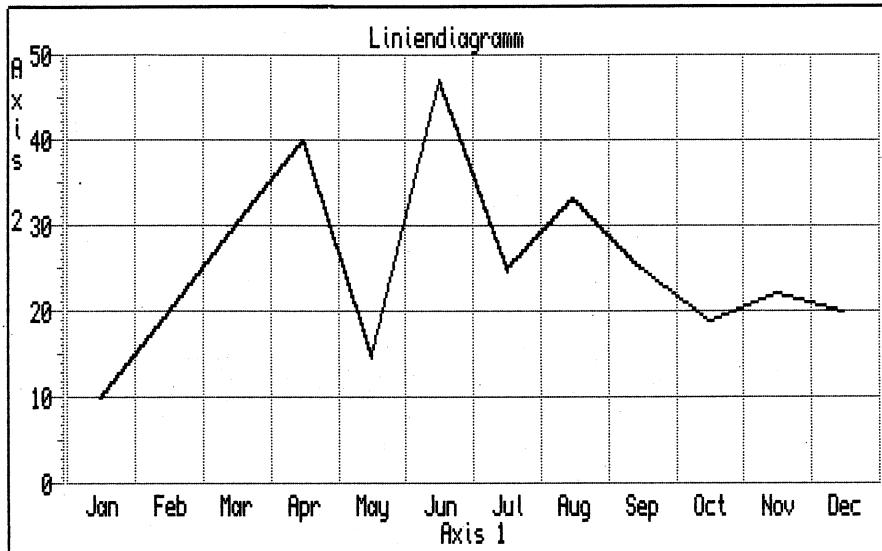
# 5

# Das Grafikprogramm EASEL

## 5.1 Befehle

EASEL ist ein Graphikpaket zur Darstellung von Zahlen in Diagrammen. Es ist interaktiv, leicht zu bedienen und sehr vielseitig. Sie können wählen zwischen Balken-, Torten- und Liniendiagrammen. Hintergrund, Beschriftung und Farben sind veränderbar.





Legen Sie die Kassette mit EASEL in Laufwerk 1 und laden Sie das Programm mit

LRUN MDV1\_BOOT

Bitte die Programm-Kassette nicht aus dem Laufwerk nehmen, da EASEL immer wieder nachlädt. Zum Abspeichern von Programmen und Bildern eine Kassette in Laufwerk 2 einlegen.

Sobald das Programm fertig geladen ist, können Sie Zahlen oder Formeln eingeben. Die Werte werden sofort in ein Diagramm eingezeichnet. Dabei werden Maßstab und Skala automatisch den Werten angepaßt. Außerdem können Zahlen, die mit ABACUS oder ARCHIVE erstellt wurden, über IMPORT eingelesen werden.

#### Die Funktionstasten :

F1 = HELP. Hilfsfunktion. Durch Drücken von F1 erhalten Sie weitere Informationen.

F2 = Prompts. An- und Abschalten der Anzeige des Kontroll-Bereichs (oberer Bildschirmteil).

F3 = COMMANDS. Schaltet in den Befehlsmodus um.

F4 = Löschen einer Zelle oder eines anderen Objektes (Text, Achsen,...). Das Objekt, auf dem sich das Fadenkreuz befindet, wird gelöscht.

F5 = Einfügen einer Zelle. Rechts vom Fadenkreuz wird eine leere Zelle eingefügt.

ESC = Korrektur bei noch nicht vollständig eingegebenen Befehlen und Abbrechen der Druck-Ausgabe.

TABULATE und SHIFT & TABULATE bewegen das Fadenkreuz.

#### Befehle :

Da alle notwendigen Parameter im Dialog vom System abgefragt werden, genügt es, nur den Befehl, ohne weitere Angaben einzugeben.

Sie kommen in den Befehlsmodus durch Drücken der Taste

### F3.

Das Eingeben des Anfangsbuchstabens der Befehle genügt.

C für Change = Ändern.

Änderungsmöglichkeiten :

A Axis.

Ändern der Achsenfarbe und der Zahlen an der Y- Achse.  
Die Achsen können auch weggelassen werden.

B Bar.

Das Balkendiagramm kann aus den 16 vordefinierten ausgewählt oder neu erstellt werden.

F Format.

Mit Format kann man zwischen acht verschiedenen Anzeigarten wählen (0-7). Zu Beginn ist Format 0 geladen und es wird ein Balkendiagramm angezeigt.

G Graph Paper.

Ändern der Hintergrundfarbe und der Rasterfarbe. Es stehen 7 Möglichkeiten zur Auswahl (0-6), Sie können aber auch selbst einen neuen Hintergrund entwerfen.

L Line.

Sie können zwischen 16 verschiedene Arten von Linien wählen, und auch weitere selbst entwerfen.

S Segment.

Farbe eines Segmentes ändern; nur beim Tortendiagramm möglich.

T Text.

Farbe für Text und Hintergrund auswählen.

Beispiel: Ändern des Formates.

F3  
C  
F

und eine Zahl eingeben, wenn Sie die Nummer des gewünschten Formates bereits kennen. Ansonsten drücken Sie einfach ENTER (damit wird das Fragezeichen eingegeben), und alle Formate werden angezeigt.

## **EDIT (Editieren):**

**Text.**

Stellen Sie das Fadenkreuz auf den zu ändernden Text und benutzen Sie den Zeileneditor, um den gewünschten Text einzugeben. Nach Drücken von ENTER kann der Text noch mit den Cursor-Tasten verschoben werden. Nochmaliges Drücken von ENTER fixiert den Text.

**Labels.**

Das Fadenkreuz wird auf den Text der Zelle gestellt und kann mit dem Editor bearbeitet werden. Zum Beenden ENTER drücken.

**Key.**

Das Feld mit den Kurvenangaben kann mit den Cursor-Tasten verschoben werden. Durch ENTER wird die Position festgelegt.

**Axis.**

Durch Drücken von V oder H wird entweder die Beschriftung der vertikalen oder der horizontalen Achse ausgewählt und kann, wie bei "Text" beschrieben, geändert werden.

**Weitere Befehle :**

**Backup**

Erstellt eine Sicherungskopie einer EASEL-Datei.

## **Delete**

Löscht eine Datei von Kassette.

## **Export**

Speichert alle im Speicher enthaltenen Kurven in einer für ABACUS oder ARCHIVE lesbaren Form.

## **Import**

Liest die Daten von ABACUS oder ARCHIVE und stellt sie als Graphiken dar.

## **Format**

Formatiert Kassette in MDV2\_ (oder einem anderen Laufwerk).

## **HIGHLIGHT**

Hervorheben eines bestimmten Wertes in der Graphik. Durch Eingabe von N werden alle negativen Werte hervorgehoben. Wenn Sie V eingeben, können Sie mit TABULATE und SHIFT & TABULATE das Fadenkreuz verschieben und damit den Wert bestimmen, der hervorgehoben werden soll.

## **KILL**

Löscht eine oder mehrere der eingegebenen Kurven.

## **LOAD**

Lädt eine gespeicherte Graphik mit allen abgespeicherten Parametern von Kassette.

## **NEWDATA**

Erstellt einen neuen Satz Graphiken, dem Sie einen Namen geben müssen. Die alten Daten bleiben gespeichert.

## **OLDDATA**

Zeigt einen "alten" Satz Daten wieder an.

## **PRINT**

Druckt die auf dem Bildschirm angezeigte Graphik auf einem Drucker aus, den Sie aus einer Liste auswählen können, oder speichert in druckbarer Form auf Kassette ab.

## **QUIT**

Beenden von EASEL und Rückkehr zu SuperBASIC. Mit ENTER bestätigen Sie den Befehl. Mit ESC können Sie in EASEL weiterarbeiten.

## **RENAME**

Gibt einem bestehenden Satz von Graphiken einen neuen Namen.

## **SAVE**

Speichert alle im Speicher vorhandenen Graphiken mit allen Spezifikationen auf Kassette ab.

## **VIEW**

Zeigt alle, oder einzelne Werte im gewünschten Format auf dem Bildschirm an.

## **ZAP**

Löscht alle eingegebenen Daten und zeigt wieder die ursprüngliche Monatsbeschriftung an. Die geänderten Parameter werden nicht in den Ursprungszustand zurückversetzt.

## **FUNKTIONEN :**

**ABS(X)** Absolutbetrag von x.

**ATN(x)** Arcustangens von x im Bogenmaß.

- COS(x) Cosinus von x im Bogenmaß.
- EXP(x) E hoch x
- INT(x) Integer (ganzzahliger Anteil) von x
- LN(x) Natürlicher Logarithmus von x (Basis e).
- PI() Konstante Pi = 3.141592.
- SGN(x) Signumfunktion : 1 für  $x > 0$ , 0 für  $x = 0$  und -1 für  $x < 0$ .
- SQR(x) Quadratwurzel von x. ( $x > 0$ )
- TAN(x) Tangens von x im Bogenmaß.

#### Ausdrucken :

Mit EASEL ist es möglich, auch auf dem Drucker Delta-10 von STAR Graphiken ausdrucken zu lassen. Dabei ist folgendes zu beachten :

Die Übertragungsgeschwindigkeit beträgt 9600 Baud, die Datenlänge 8 Bit, und es wird ohne Paritätsprüfung und ohne Handshaking übertragen.

Die Schalter von Dip-Schalter 3 erhalten damit die Stellungen :

1 2 3 4 5 6 7 8

OFF OFF OFF OFF ON ON ON ON

Das Verbindungs kabel zum Drucker steckt in SER1.

## 5.2 Anwendungen

### 5.2.1 Importieren einer mit ABACUS erstellten Datei

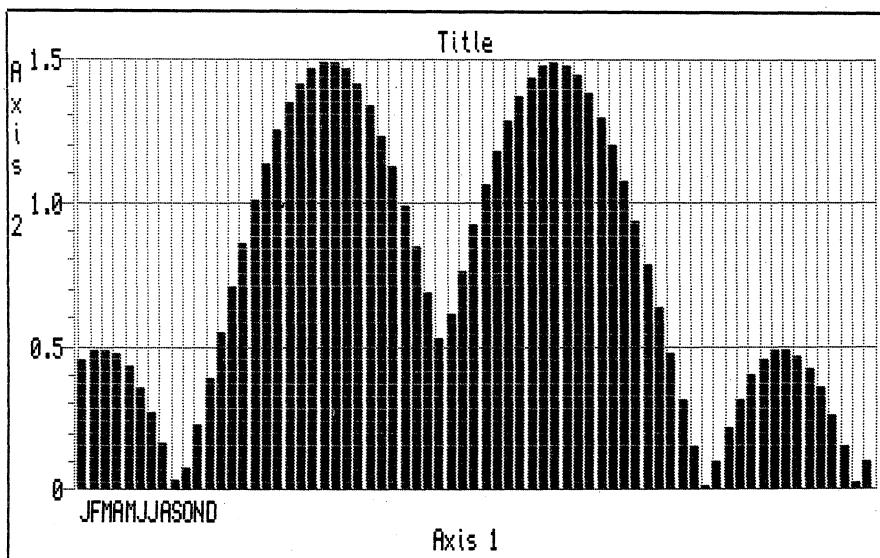
Mit ABACUS haben wir die Funktionswerte der Funktion  $y = \text{ABS}(0.5 + \text{SIN}(\text{ABS}(x)))$  im Bereich von -5 bis ungefähr +5 berechnen lassen, und in einer für EASEL lesbaren Form unter dem Namen Funktion1 abgespeichert (mit Export). Diese Datei wollen wir nun mit EASEL anzeigen und ausdrucken.

Das Einlesen der Datei erfolgt mit dem Befehl IMPORT.

F3 F I Funktion1

command> Files: import from Funktion1

Die Datei wird nun geladen, und auf dem Bildschirm wird folgendes Bild angezeigt :



Da statt eines Balkendiagrammes ein Liniendiagramm dargestellt werden soll, geben wir den Befehl ein:

F3 C L 1

command> Change to LINE 1

Das störende Gitter verschwindet mit

F3 C G 5

command> Change to GRAPH PAPER 5

Die Monatsbeschriftung wird gelöscht :

F3 E L F4 zwölf mal drücken, bis alle Namen gelöscht sind.

command> Edit Label

Edit label: Jan F4 drücken.  
Feb F4 drücken usw.

Die Überschrift eintragen :

F3 E T

Das Achsenkreuz mit den Cursor-Tasten auf das T von Title bewegen, ENTER drücken.

add text "Title

überschreiben mit  $Y=ABS(0.5+\sin(ABS(X)))$

ENTER drücken, und die Schrift mit der Taste "Cursor nach links" etwa drei Zentimeter nach links verschieben.

Nochmals ENTER drücken, und der Text ist fixiert.

Zuletzt werden noch die Achsen beschriftet.

F3 E A V

command> Edit axis name V

add text "Axis 2

mit "Y-Achse " überschreiben.

Die ENTER-Taste drücken, den Text etwas nach unten verschieben, und ENTER drücken zum fixieren.

F3 E A H

command> Edit axis name H

add test "Axis 1

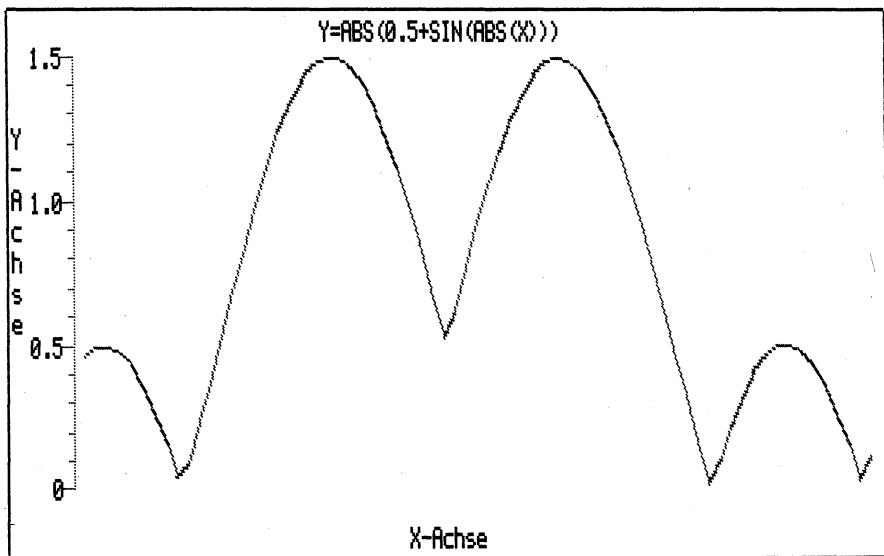
mit "X-Achse überschreiben, und zwei mal die ENTER-Taste drücken.

Der fertige Graph kann nun ausgedruckt werden.

F3 P P

command> print graph

Den Ausdruck des fertigen Graph finden Sie auf der nächsten Seite (138).



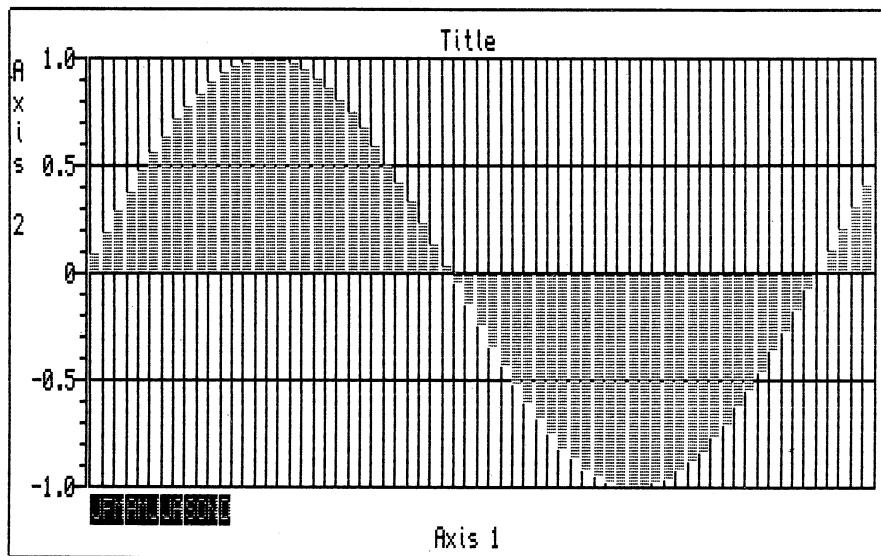
**Fertiger Graph**

## 5.2.2 Eingeben einer Funktion

Ich will Ihnen anhand der Sinusfunktion zeigen, wie man schnell Funktionen mit EASEL zeichen kann. Dazu ist es nötig, die Anzahl der dargestellten Zellen zu vergrössern. Mit der Taste "TABULATE" den Zellenanzeiger solange nach rechts bewegen, bis die Zellenzahl fünfmal vergrössert wurde. Durch den Wert "cell", der die Nummer der aktuellen Zelle enthält, lassen sich einfach Funktionen eingeben :

```
a = sin(cell/10)
```

Ihr Bild sieht dann so aus :



Das hier störende Gitter wird unsichtbar mit  
command> Change to GRAPH PAPER 5

Anstatt des Balkendiagrammes soll ein Liniendiagramm dargestellt werden.

command> Change to LINE 0

Das Löschen der Monatsbeschriftung geschieht durch

command> Edit Labels

und Drücken der Taste F4 zum Löschen.

Mit "Edit Text" und "Edit Labels" werden die Überschrift und die Achsennamen geändert.

Die Kurve "a" wird abgespeichert, und neue Daten können eingegeben werden mit NEWDATA.

Als neuer Name wird "Nu" eingetippt, für Nulllinie.

Eine Null-Linie wird gezeichnet durch

Nu = 0\*cell

Um diese Linie sichtbar zu machen, kann es nötig sein, mit "Change to LINE 1" die Linie zu ändern.

Erneut den Befehl NEWDATA eingegeben, und als neuen Namen "b" wählen.

Die zweite Funktion wird eingetragen mit

b=sin(cell/3)

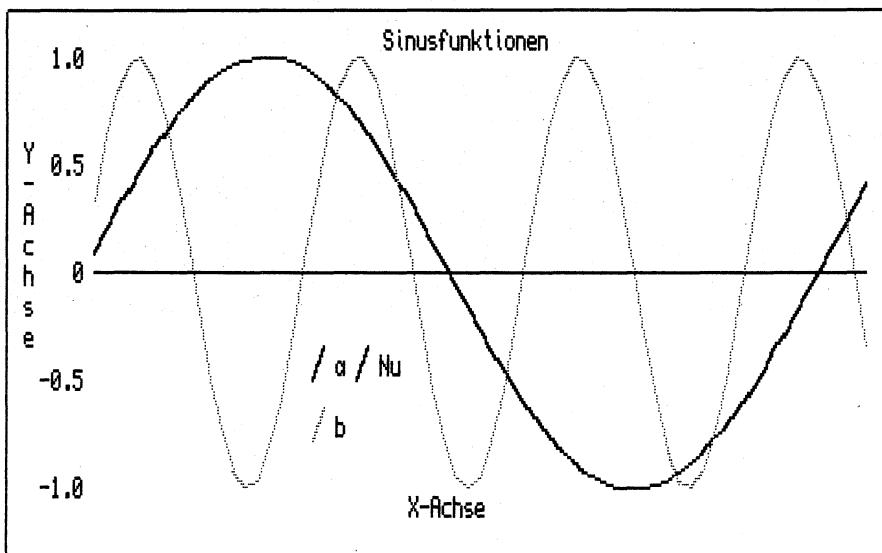
Die Darstellung wird auf "LINE 2" geändert.

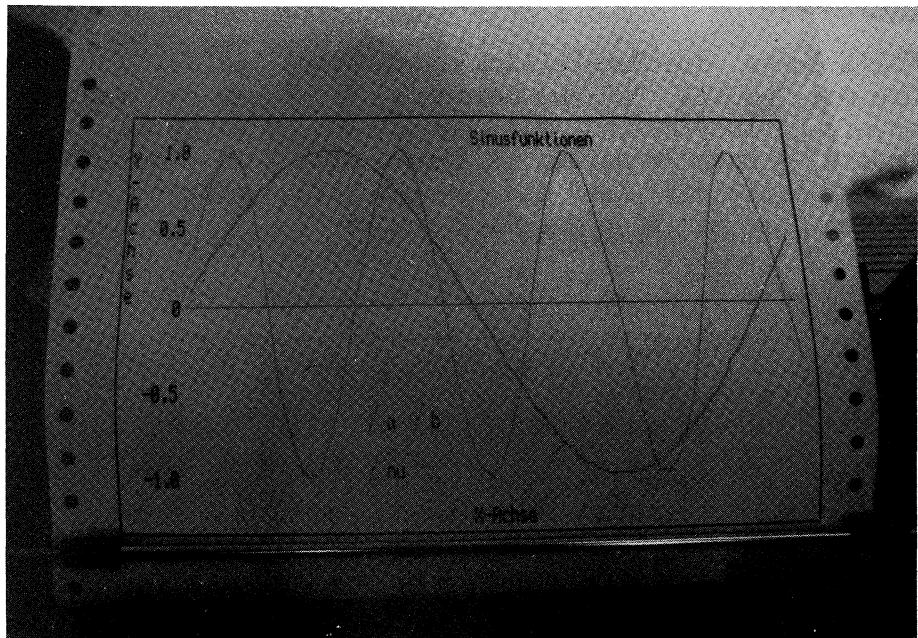
Mit dem Befehl

View a,b,Nu with format 3

werden alle Graphen angezeigt.

Falls Ihnen die Darstellung noch nicht gefällt, können Sie mit OLDDATA "a" zum Beispiel die Kurve "a" nochmals anzeigen lassen, und mit dem Befehl "Change to LINE" ändern. Nach erneutem Anzeigen mit dem View-Befehl, können mit "Edit Key" die Kurvenbezeichnungen verschoben werden, und man kann das Bild drucken.





**Ausdruck auf dem Drucker**

# 6

# Das Textverarbeitungs- programm QUILL

Quill (zu deutsch Feder) ist ein einfach zu bedienendes Textverarbeitungsprogramm mit den Grundfunktionen Text eingeben, abspeichern, korrigieren, ausdrucken und formatieren. Zusätzlich ist es möglich, die Schrifttypen zu ändern, Buchstaben zu suchen und zu ersetzen, und das Druckformat frei zu gestalten.

## Starten des Programmes

Legen Sie die Kassette mit Quill in MDV1\_ und eine formatierte Kassette zum Abspeichern von Texten in MDV2\_. Laden Sie das Programm mit

LRUN MDV1\_BOOT

Bitte lassen Sie beide Kassetten in den Laufwerken, da Quill immer wieder darauf zugreift.

Die Funktionstasten :

F1 = Help. Hilfe-Taste, die zusätzliche Informationen liefert.

F2 = Prompts. An- und Abschalten der Anzeige des Kontrollbereichs.

F3 = Commands. Schaltet in den Befehlsmodus um.

F4 = Auswahl zwischen verschiedenen Schrifttypen.

SHIFT F4 = Umschalten zwischen Einfügen und Überschreiben.

ESC = Korrektur bei noch nicht vollständig eingegebenen Befehlen.

TABULATE und SHIFT & TABULATE bewegen den Tabulator vor und zurück.

### Schriftarten (Typefaces)

Normal

Fettdruck (Bold)

Unterstrichen (Underline)

Hochgestellt (High)

Tiefgestellt (Low)

Dazu kommt noch die Möglichkeit, die Schrift eines bereits eingegebenen Textes zu ändern (Paint).

Die verschiedenen Schrifttypen können Sie auswählen durch Drücken von F4 und Eingeben des Anfangsbuchstabens der gewünschten Schrift. In der letzten Bildschirmzeile wird die augenblicklich gültige Schrift hinter dem Wort TYPEFACE angezeigt. Natürlich sind auch Kombinationen von verschiedenen Schriften möglich. Das Abschalten der speziellen Schriftart erfolgt wie das Einschalten - durch die Eingabe des Anfangsbuchstabens der Schrift.

## Auswahl von Fettdruck :

F4

B

## Fettdruck und Unterstreichen :

F4

B

F4

U

## Die Befehle:

Um einen Befehl einzugeben, müssen Sie zuerst mit F3 in den Befehlsmodus umschalten. Im oberen Bildschirmteil wird eine von zwei Listen mit den Befehlen angezeigt. Auf die zweite Liste können Sie mit 0 ( für Other) umschalten. Es genügt, wenn Sie den Anfangsbuchstaben des Befehles eingeben. Das Programm fragt Sie dann im Dialog nach weiteren Angaben.

## Befehle der 1. Befehls-Seite :

	Erase	Header	Margins	Save
Copy	Footer	Justify	Print	Tabs
Design	Goto	Load	Quit	
OTHER				

## Befehle der 2. Befehls-Seite :

	Merge	Search
Files	Page	Zap
Hyphenate	Replace	
OTHER		

## COPY

Verschiebt oder kopiert Text.

Bewegen Sie den Cursor an den Anfang des zu kopierenden Textes und Drücken Sie ENTER. Stellen Sie den Cursor an das Ende des zu kopierenden Textes und drücken Sie ENTER. Der Abschnitt wird dann hervorgehoben. Jetzt gehen Sie mit dem Cursor an die Stelle, an die der Text eingefügt werden soll und drücken C. Mit D (für Delete = Löschen) wird dann der markierte Abschnitt gelöscht, mit ENTER bleibt er erhalten. Wenn Sie den markierten Text noch an anderen Stellen einfügen wollen, bewegen Sie den Cursor wieder an die gewünschte Stelle und geben Sie C ein, sonst drücken Sie ENTER.

## DESIGN

Dieser Befehl kontrolliert mehrere Optionen, die die Druckausgabe betreffen.

B Bottom Margin = Unterer Rand. Gibt die Anzahl der Leerzeilen am Ende jeder Seite an. Vorgegeben sind 6 Zeilen.

D Display Width = Anzahl der Zeichen pro Zeile. Mit 4, 6 oder 8 kann man zwischen 40, 64 oder 80 Zeichen auswählen.

G Gaps = Zwischenraum zwischen den Zeilen. Es sind 0, 1 oder 2 Leerzeilen möglich.

P Page Size = Seitengröße. Gibt die Gesamtzahl der Zeilen einer Seite an. Der Standardwert ist 66.

S Start Page Number = Nummer der ersten Seite. Der Standardwert ist 1.

U Upper Margin = Oberer Rand. Gibt die Anzahl der Leerzeilen am Anfang jedes Textes an. Der Standardwert ist 6.

Am Ende müssen Sie ENTER eingeben, wenn Ihre Änderungen übernommen werden sollen. Mit ESC werden die Änderungen wieder gelöscht.

## ERASE

Löscht einen Teil Ihres Textes. Der zu löschenende Abschnitt wird mit den Cursor-Tasten angegeben, wie beim Befehl COPY.

## FILES

Erlaubt Ihnen, Befehle zum Umgang mit den Mikrolaufwerken einzugeben.

B Backup = Datensicherung. Erstellt eine Sicherungskopie von einer Ihrer Text-Dateien.

D Delete = Löschen. Löscht eine Text-Datei von Kassette.

F Format = Formatieren. Formatiert die Kassette in MDV2\_.

I Import = Importieren. Einfügen einer mit ABACUS oder ARCHIVE exportierten Datei oder einer anderen Text-Datei.

## FOOTER

Position der Fußnoten im Text:

- |        |                  |
|--------|------------------|
| None   | Keine Fußnoten.  |
| Left   | Am linken Rand.  |
| Centre | In der Mitte.    |
| Right  | Am rechten Rand. |

Mit der Leertaste wählen Sie die gewünschte Option aus, drücken ENTER und geben die Fußnote ein. Zusätzlich können Sie angeben, ob die Seitenzahlen im arabischen, römischen oder alphabetischen Format angegeben werden, und wieviele Leerzeilen zwischen dem Textende und der Fußnote liegen sollen.

## GOTO

Bewegt den Cursor mit

T für Top an den Anfang des Textes.

B für Bottom an das Ende des Textes.

einer Seitenzahl an den Anfang dieser Seite.

## HEADER

Mit diesem Befehl können Sie die Stelle und den Text einer Überschrift bestimmen. Mit der Leertaste wählen Sie zwischen folgenden Optionen aus :

- None Keine Überschrift.
- Left Am linken Rand.
- Centre In der Mitte.
- Right Am rechten Rand.

Das Format der Seitenzahlen können Sie wie bei FOOTER ändern.

## **HYPHENATE**

Trennen eines Wortes am Ende einer Zeile und automatisches Einfügen eines Trennungsstriches. Bewegen Sie den Cursor auf das erste Zeichen im Wort, das auf der neuen Zeile stehen soll und drücken Sie H.

## **JUSTIFY**

Erlaubt Ihnen, die Art der Ausrichtung Ihres Textes zu ändern. Der Befehl wird vom Beginn des Abschnittes gültig, in dem sich der Cursor befindet.

L Left = Ausrichtung nur links.

C Centre = Ausrichtung in der Mitte.

R Right = Ausrichtung Rechts und Links.

## **LOAD**

Lädt einen Text von MDV2\_. Durch die Eingabe eines ? wird das Inhaltsverzeichnis angezeigt.

## **MARGINS**

Legt die Position der Ränder fest. Mit der Leertaste wählen sie aus, ob der linke Rand (LEFT), das Einrücken (INDENT) oder der rechte Rand (RIGHT) verändert werden soll. Bewegen Sie den Cursor auf die gewünschte Position und Drücken Sie ENTER, wenn alle Änderungen abgeschlossen sind.

## MERGE

Kopiert einen Text von Kassette an die momentane Cursor-Position.

## OTHER

Schaltet auf die zweite Seite der Befehle um.

## PAGE

Mit Page können Sie einem Seitenvorschub im Text angeben. Bewegen Sie den Cursor an die gewünschte Stelle und drücken Sie ENTER. Durch SHIFT und => wird ein Seitenvorschub im Text gelöscht.

## PRINT

Dieser Befehl dient zum Ausdrucken des aktuellen Textes, oder eines gespeicherten Textes, ganz oder nur zum Teil. Der Ausdruck kann auch auf Kassette abgespeichert werden.

## QUIT

Beenden von QUILL und Rückkehr zu SuperBASIC. Sie haben drei Eingabemöglichkeiten :

ENTER Speichert den Text ab, wobei sie einen anderen Namen dafür angeben können.

A für Abandon = Aufgeben. Speichert den Text nicht ab.

ESC macht den QUIT-Befehl rückgängig.

## REPLACE

Ersetzt die angegebene Buchstabenfolge durch eine andere. Bei jedem Vorkommen entscheiden Sie, ob ersetzt werden soll (Y eingeben), oder nicht (andere Taste drücken).

## SAVE

Sichert den aktuellen Text unter einem frei wählbaren Namen auf die Kassette in MDV2\_. Wenn nichts anderes angegeben, wird \_doc angehängt.

## SEARCH

Sucht eine Buchstabenfolge in Ihrem Text.

## TABS

Legt die Position der Tabulatorsprünge fest. Die Änderung ist von dem Paragraph ab wirksam, in dem sich der Cursor gerade befindet.

Möglichkeiten :

L Left.

Der Text wird ab der Tabulator-Position fortgesetzt.

C Centred.

Der Text wird auf der Tabulator-Position zentriert.

R Right.

Der Tabulator bildet den rechten Rand des Textes.

D Decimal.

Richtet Zahlen so aus, das der Dezimalpunkt an der Stelle des Tabulators steht.

Durch Eingabe von T wird eine Tabulatorposition fixiert,  
mit X wird diese gelöscht.

## ZAP

Löscht den ganzen aktuellen Text. Nach dem Löschen  
können Sie die Kassette in MDV2\_ wechseln.

### Beispiele zu den Befehlen:

Linker Rand 1, keine Einrückung, rechter Rand 60:

F3

M

L(left) ist hervorgehoben.

Mit dem Cursor nach links bis zur 1. Stelle.

Leertaste drücken.

I(ndent) ist hervorgehoben. Mit dem Cursor nach links  
bis zur 1. Stelle.

Leertaste drücken.

R(right) ist hervorgehoben. Mit dem Cursor auf Stelle 60.

### Ändern der Seitengröße von 66 auf 62:

F3

D

P Page Size wird invers dargestellt.

62 eingeben.

ENTER.

## Eine Überschrift zentriert drucken:

F3

H

Zweimal Leertaste drücken, dann ist Centre erhellt. rrr für römische Ziffern eintippen, und dann die gewünschte Überschrift eingeben.

3 als Abstand zum Text eingeben.

N für normale Schrift.

## Ein Wort unterstreichen:

F4

U

Das Wort eintippen

F4

U

Oder :

Das Wort eingeben und den Cursor an den Wortanfang bewegen.

F4

U

P

Cursor an das Ende des Wortes stellen.

ESC

**Ausdrucken des ganzen eingegebenen Textes:**

F3  
P  
ENTER  
ENTER  
ENTER

**Ausdrucken von Seite 2 und 3 des eingegebenen Textes:**

F3  
P  
ENTER  
2  
ENTER  
3  
ENTER  
ENTER

**Den linken Tabulator auf 5, den rechten auf 55 setzen, mit einer Ausrichtung für den Dezimalpunkt auf 40:**

F3  
T  
Cursor auf Position 5 und T eingeben.  
Leertaste zweimal drücken, dann wird R(ight) heller gezeigt.  
Cursor auf Position 55 und T eingeben.  
Leertaste dreimal betätigen, dann ist D(ecimal) aufgeheilt.  
Cursor auf Position 40 und T eingeben.  
ENTER

**Abspeichern unter dem Namen BRIEF1:**

F3  
S  
BRIEF1  
ENTER

## **Notizen**

# 7

## Druckeranschluß an Star Delta-10

Der Drucker Delta-10 von Star läßt sich mit folgenden Spezifikationen am QL betreiben :

Datenlänge 8 Bit  
ohne Paritätsprüfung  
ohne Handshaking  
Bit-Rate 1200 Baud

1. Umschalten auf seriellen Betrieb :

Dip-Schalter 2-2 auf ON

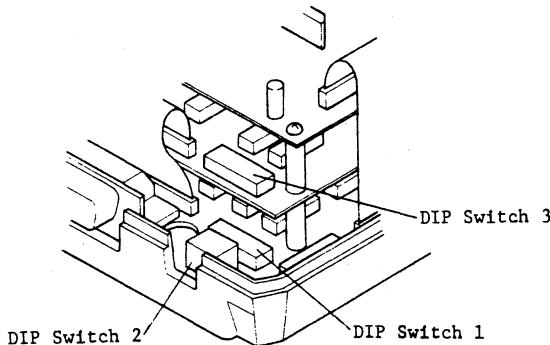
2. Interface auf 8 Bit schalten:

Dip-Schalter 2-3 auf OFF

Die Schalter von Dip-Schalter 3 erhalten folgende Stellungen :

1 2 3 4 5 6 7 8

OFF OFF OFF OFF ON ON OFF OFF



Um ein Programm auszudrucken ist folgendes anzugeben :  
 (Anstatt der Kanalnummer 4, die hier verwendet wird, können Sie auch andere Nummern benutzen.)

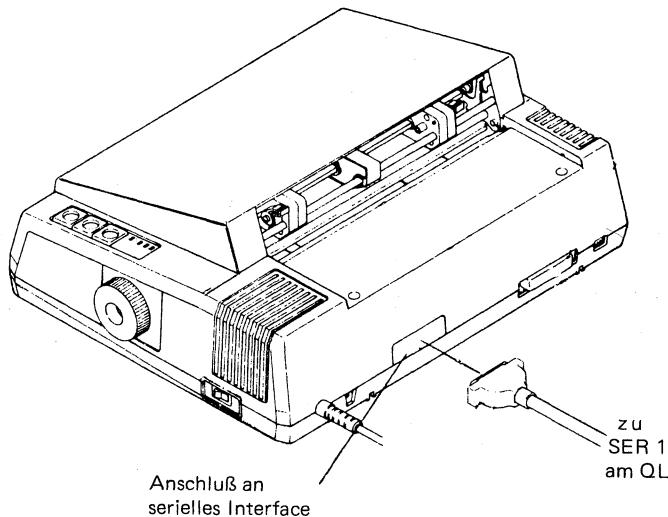
BAUD 1200

OPEN #4,SER1I (I bedeutet ohne Handshake)

LIST #4

CLOSE #4

Das Kabel läuft vom RS-232-Anschluß SER1 des QL zum parallelen Interface des Delta 10.



Es ist auch eine andere Übertragungsgeschwindigkeit möglich, wenn Sie den BAUD-Befehl ändern und die Dip-Schalter 3-6, 3-7 und 3-8 umschalten.

Z.B. Übertragungsrate von 9600 Baud

BAUD 9600

Dip-Schalter 3-6 3-7 3-8

ON ON ON

Steuerzeichen :

Steuerzeichen werden im Dezimalcode übermittelt.

Beispiel : Sie wollen ein Programm in Elite-Schrift ausdrucken. Der Code dafür ist ESC B 2 oder CHR\$(27) CHR\$(66) CHR\$(1)

Die Anweisungen lauten :

```
BAUD 1200
OPEN #4,SER1I
PRINT #4,CHR$(27)&CHR$(66)&CHR$(1)
LIST #4
CLOSE #4
```

Das Erstellen eines Ausdruckes in einem Programm erfolgt auch mit dem Befehl PRINT #4.

Beispiel :

```
10 REMark TEST
20 BAUD 1200
30 OPEN #4,SER1i
40 PRINT #4,"Dies ist ein Test. "
50 CLOSE #4
```

## **Notizen**

# Anhang- Befehlsübersicht

AUTO 10,10

AUTO 100

Beenden mit CTRL - Leertaste.

BAUD 1200

BAUD 2400

CALL 79422,...

CLEAR

CLOSE #3

CONTINUE

COPY MDV1\_ TO MDV2\_

COPY\_N MDV2\_ TO SER1

DELETE MDV1\_Prog1

DIR MDV2\_

DLINE 20 oder einfach 20  
DLINE TO 20  
DLINE 20 TO  
DLINE 10 TO 20

EDIT 10  
EDIT 10,5

EXEC Test1,Test2

EXEC\_N Test3,Test4

FORMAT MDV1\_

KEYROW(2)

LIST  
LIST #1  
LIST TO 50  
LIST #1, TO 50  
LIST 10 TO 50  
LIST #1,10 TO 50

LOAD MDV1\_Test  
LOAD MDV2-Test

LBYTES MDV1\_88724

LRUN Test

MERGE MDV1\_Test2

MRUN Test2

NET 2

NEW

OPEN #4,SER1

OPEN\_IN #5,SER2

OPEN\_NEW #5,MDV2\_sort

PEEK (64728)

PEEK\_W(64728)

PEEK\_L(64728)

POKE 64728,33

POKE\_W 64328,12

POKE\_L 64124,2234

RENUM 10

RENUM 10,10

RENUM 50;10,10

RENUM 50 TO 200;100,10

RETRY

RUN  
RUN 100

SAVE MDV1\_Test  
SAVE MDV2\_Test  
SAVE SER1\_Test

SEXEC MDV1\_64210,512,56

# SuperBASIC

ADATE 7200

BEEP 10000,40,50,10,12,5,10,20000

BEEPING

DATA 1,5,8,13,24  
DATA MONTAG,DIENSTAG

DATE

DATE\$

DAY\$(01/01/1985)

DEFIne FuNction Berech (Hilf)  
  LOCAL a,b  
  IF a > b THEN Berech (a)  
  ...  
  RETURN b  
END DEFIne

DEFIne PROCedure Berech (Hilf)  
  LOCAL c,d  
  ...  
END DEFIne

```
DEFIne PROCedure Test1 (xx)
    LOCAL e,f
    ...
    RETURN e
END DEFIne
```

```
DIM a$(10,10,5)
DIM tt(12,3,10)
```

```
DIMN(a$,2)
DIMN(tt,3)
```

```
EXIT Schleife
```

```
FOR i = 1 TO 100 : PRINT i
FOR i = 2,4,8 TO 18 STEP 1.5
...
END FOR i oder NEXT i
```

```
FOR i = 2 TO 15 STEP 3
...
IF .. THEN EXIT i
END FOR i
```

```
GO TO 320
```

```
GOSUB 2000
```

```
INKEY$(#5)
INKEY$(0)
INKEY$(100)
INKEY$(-1)
```

```
INPUT #4
INPUT "a eingeben ";a
INPUT a
```

```
ON xx GOTO 20,60,125,200
```

```
ON xx GOSUB 30,145,60
```

```
IF b < x THEN ...
```

```
IF a > 0 THEN ...
ELSE ...
END IF
```

```
LET a = 5
LET a(1) = 2
LET a$ = "Sonntag"
```

```
PAUSE
PAUSE 100
```

```
PRINT #5,a,b,c
PRINT "Ergebnis :";a;b
PRINT ! a
PRINT A \ B
```

```
READ a,b,c,d
```

```
REPeat Liste
...
IF .. THEN EXIT Liste
...
END REPeat
```

RESTORE

```
SElect ON x
  ON x = 1,2,3
    PRINT x
  ON x = 4
    PRINT y
  ...
  ON x = REMAINDER
    PRINT "Fehler"
END SElect
```

STOP

# ZEICHENBEHANDLUNG

AT 10,15  
AT #2,10,15

CSIZE 3,1  
CSIZE #2,3,1

CURSOR 10,30  
CURSOR #2,1,10  
CURSOR 10,20,A,B  
CURSOR #2,10,20,5,5

FLASH 1  
FLASH 0

OVER -1  
OVER 0  
OVER 1

SDATE 1985,24,02,12,10,0

STRIP 3  
STRIP #2,4

UNDER 0  
UNDER 1

# GRAPHIK

ARC 10,10 TO 50,40,90  
ARC #2,10,10 TO 50,40,90  
ARC TO 100,90,45

ARC\_R #2,0,0 TO 40,40,30

BLOCK 8,6  
BLOCK #2,8,6  
BLOCK #2,8,6,A,B  
BLOCK #1,8,6,10,30

BORDER 2,7  
BORDER #2,5,1

CIRCLE 50,50,20  
CIRCLE #2,50,50,20  
CIRCLE 50,50,20,0.5,10  
CIRCLE #2,50,50,20,0.5,10

CIRCLE\_R #2,50,50,20

CLS  
CLS #1  
CLS #0,0  
CLS 1  
CLS 2  
CLS 3  
CLS 4

FILL 0  
FILL 1

INK 2  
INK #2,5

LINE 10,50  
LINE #2,10,50  
LINE TO 60,100  
LINE #0 TO 80,110  
LINE 10,10 TO 30,70  
LINE #0,10,20 TO 40,70

LINE\_R #2,0,0 TO 10,10

MODE 4 oder MODE 512  
MODE 8 oder MODE 256

MOVE 20  
MOVE #2,20

PAN 10  
PAN #2,10  
PAN 10,3  
PAN -10,4

PAPER 3  
PAPER #0,1

PENUP  
PENUP #2

PENDOWN  
PENDOWN #1

POINT 30,40  
POINT #0,40,60

POINT\_R #2,10,10

RECOL 7,3,1,5,2,4,6,0  
RECOL #2,5,3,2,7,0,6,1,4

REMark

SCALE 120,40,40  
SCALE #0,120,40,40

SCROLL 20  
SCROLL -30  
SCROLL #2,20

TURN 30  
TURN #2,30

TURNT0 0  
TURNT0 #0,0

WINDOW 10,20,0,0  
WINDOW #2,50,100,20,10

# MATHEMATISCHE FUNKTIONEN

ABS(x)

ACOT(x)

ATAN(x)

COS(x)

COT(x)

DEG(x)

DIV

EXP(x)

INT(x)

LN(x)

LOG10(x)

MOD

PI

RAD(x)

RANDOMISE 12345

RND(x)

RND(x TO y)

SGN(x)

SIN(x)

SQRT(x)

TAN(x)

## Zeichen-Funktionen

CHR\$(127)

CODE(a)

FILL\$(aa,4)

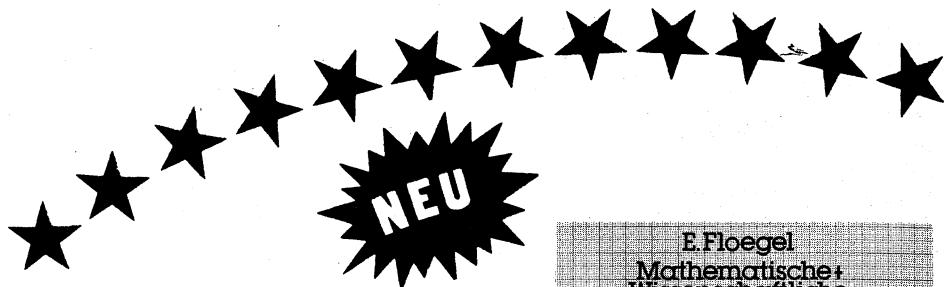
test INSTR es

a & b

## OPERATOREN

+	Addition
-	Subtraktion
*	Multiplikation
/	Division
^	Potenzierung
DIV	Ganzzahlige Division
MOD	Divisionsrest
OR	ODER
AND	UND
XOR	Exklusives ODER
NOT	Negierung
	Bitweise ODER
&&	Bitweise UND
^^	Bitweise exklusives ODER
~~	Bitweise NICHT
=	Gleich
==	Fast gleich
>	Größer
<	Kleiner
>=	Größer gleich
<=	Kleiner gleich
<>	Ungleich

## **Notizen**



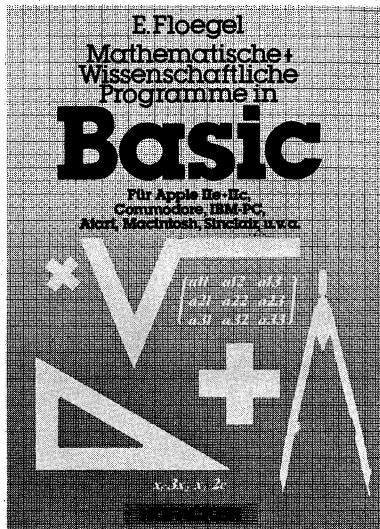
## **Mathematische + Wissenschaftliche Programme in BASIC, E. Flögel**

Die Programme sind in Standard BASIC verfasst, so daß sie auf allen Personalcomputern mit Microsoft BASIC arbeiten. Ideal für Apple IIe und IIc, sowie für IBM PC und Commodore.

Aus dem Inhalt: Elementare Mathematik, Funktionen und Polynome, Komplexe Zahlen, Vektoren, Matrizen, Lineare Gleichungssysteme, Ausgleichsrechnung, numerische Integration von Differentialgleichungen, Lösung des Dirichletproblems durch Differenzenverfahren, Lineare Regression, Lineare Planungsrechnung, Algebra.

**Best.-Nr. 102**

**29,80 DM**



## **Statistik in BASIC**

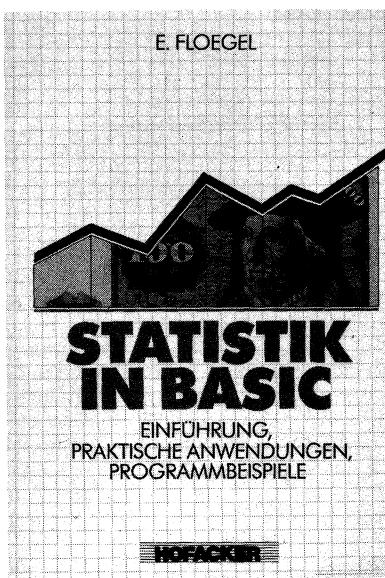
### **Einführung und praktische Anwendung**

Hier finden Sie das Statistik Buch auf das Sie schon lange gewartet haben. Es werden nicht nur Programme u. deren Ergebnisse mitgeteilt, sondern auch das ganze warum und drumherum wird mit beleuchtet. Die Programme sind vorwiegend in Microsoft BASIC geschrieben, so daß Sie auf nahezu allen Personalcomputern arbeiten müssten. Bei Grafikprogrammen wurden Versionen für den IBM PC und den Apple IIe oder IIc mit angegeben.

Ganz besonders ausführlich wurde der Teil Grafik behandelt. Aus dem Inhalt: Wahl der Daten, graphische Darstellung, Berechnung der statistischen Kenngrößen, Wahrscheinlichkeitsrechnung, statistische Testverfahren, Markov-Analyse, lineare Regression, Autokorrelation, Statistik und Glücksspiel, usw. Von diesem Buch werden Sie begeistert sein.

Verfasser: E. Flögel. ca. 220 Seiten  
**Best.-Nr. 188**

**39,- DM**



```
SCR # 151
0 BUSINESS ADDR INPUT
1 VOCABULARY ADRESSEN IMMEDIATE
2 ADRESSEN DEFINITIONS
3 DESCRI (VH) 0, 28,
4 DESCRI (COL) 28, 23,
5 DESCRI (ST) 51, 23,
6 DESCRI (PLZ) 74, 7,
7 DESCRI (ORT) 81, 21,
8 MESSEN (LNU) 075,
```

E. Floegel

# FOR TH

0. SER.  
1. TR.  
2. E.  
3. E.  
4. E.  
5. E.  
6. E.  
7. E.  
8. E.  
9. E.  
10. E.  
11. E.  
12. E.  
13. E.  
14. E.  
15. E.  
16. E.  
17. E.  
18. E.  
19. E.  
20. E.  
21. E.  
22. E.  
23. E.  
24. E.  
25. E.  
26. E.  
27. E.  
28. E.  
29. E.  
30. E.  
31. E.  
32. E.  
33. E.  
34. E.  
35. E.  
36. E.  
37. E.  
38. E.  
39. E.  
40. E.  
41. E.  
42. E.  
43. E.  
44. E.  
45. E.  
46. E.  
47. E.  
48. E.  
49. E.  
50. E.  
51. E.  
52. E.  
53. E.  
54. E.  
55. E.  
56. E.  
57. E.  
58. E.  
59. E.  
60. E.  
61. E.  
62. E.  
63. E.  
64. E.  
65. E.  
66. E.  
67. E.  
68. E.  
69. E.  
70. E.  
71. E.  
72. E.  
73. E.  
74. E.  
75. E.  
76. E.  
77. E.  
78. E.  
79. E.  
80. E.  
81. E.  
82. E.  
83. E.  
84. E.  
85. E.  
86. E.  
87. E.  
88. E.  
89. E.  
90. E.  
91. E.  
92. E.  
93. E.  
94. E.  
95. E.  
96. E.  
97. E.  
98. E.  
99. E.  
100. E.

## Anwendungsbeispiele

2 : CO (CO) 20 : ST (ST) 20 ;  
3 : OHT (DRT) 20 : C1 (NR) 20 ;  
4 : PLZ (PLZ) 20 : C2 (AH) 20 ;  
5 : TEL (TEL) 20 : IC (IC) 20 ;  
6 :  
7 : CVN 6-4 CU ;

```
8 : CCO 8 4 CU ; : CST 10 4 CU ;
9 : CPL 12 4 CU ;
10 : COR 12 12 CU ;
11 : CCI 14 4 CU ;
12 : CEC 14 10 CU ;
13 : CTE 14 18 CU ;
14
15
```

- Datenverwaltungsprogramme
- Geschäftsprogramme
- Programmiertechniken
- Unterhaltungsprogramme
- Künstliche Intelligenz

OK

## FORTH Anwendungsbeispiele

von E. Flögel

Datenverwaltungs-, Geschäftsprogramme, Programmietechniken, Unterhaltungsprogramme, Künstliche Intelligenz.

Dies ist das erste Buch (206 Seiten) weltweit, welches Anwendersoftware in der Programmiersprache FORTH enthält.

FORTH hat sich in den letzten Jahren immer mehr Freunde geschaffen. In erster Linie wegen seiner hohen Portabilität. Aus dem Inhalt: Ein- u. Ausgabe von Daten, Zeichenketten in FORTH, verkettete Listen, ein kompl. Macroassembler, ein kleines Spiel, usw. sowie auch ein kompl. Geschäftsprogr., bestehend aus Adressen-, Lagerverwaltung und Fakturierung. Dieses Paket dürfte eine echte Sensation sein, da der Anwender hier den kompl. Quelltext eines Programmes erhält, dessen Entwicklungskosten eine sechsstellige Zahl betragen haben.

**Best.-Nr. 200**

49, - DM

## FORTH Handbuch

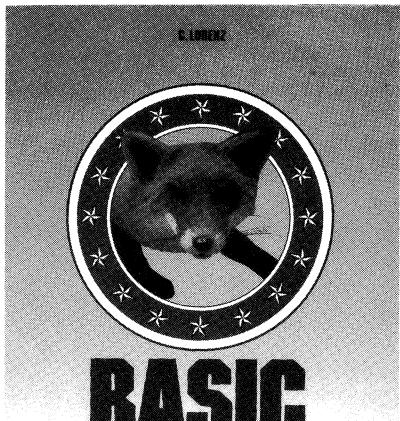
von E. Flögel

Dies ist das führende deutschsprachige Werk über diese sehr leistungsfähige Programmiersprache. Nach einer sehr guten Einführung bringt der Autor Beispiele aus der Praxis. Adressenverwaltung in FORTH mit virtueller Speicherwaltung u. v. a. mehr. Ein Muß für jeden, der FORTH kennenlernen will. FORTH ist eine extrem leistungsfähige Programmiersprache mit Interpreter- u. Compiler-eigenschaften. Die Codierung ist sehr effektiv u. Programme lassen sich in einem Bruchteil der Zeit erstellen, die man in jeder anderen, höheren Programmiersprache benötigen würde. FORTH-Programme lassen sich auf fast allen Rechnern, trotz unterschiedlicher CPU, nahezu ohne Änderungen implementieren. Die Sprache, die auch Sie unbedingt erlernen sollten. Dieses Buch (172 Seiten) ist Ihre Fahrkarte für den Einstieg in diese leistungsfähige Sprache.

Best.-Nr. 137

49. - DM

OK



# BASIC

FÜR FORTGESCHRITTENE

Werkfeinste Programmieranleitungen.  
Interessante Anwendungen aus dem Geschäftsbereich, Wissenschaft  
und Technik, IBM PC/CP/M, Commodore 64,  
Apple, Macintosh

**HOFACKER**

## BASIC für Fortgeschrittene

Die Hohe Schule für BASIC. Jeder der seinen PC besser nutzen will braucht dieses Buch. Commodore BASIC, MBASIC, Macintosh BASIC, TANDY BASIC, IBM PC BASIC viele Programmbeispiele. Künstliche Intelligenz-Anwendungsbeispiele, Grundlagen Floppy Disk Technik, Dateien auf Diskette und Cassette, in Commodore BASIC, CP/M und MS-DOS, ATARI und APPLE. Strategisches Programmieren, der binäre Baum und seine Anwendung, Sortierverfahren. Das sind nur einige Themen aus diesem sehr umfangreichen u. breitgestreuten Anwendungsfeld. Dieses Buch ist ein Muss für jeden BASIC Programmierer, gleich welchen Computer er besitzt oder noch kaufen will. (226 S.)

**Best.-Nr. 122**

**39,-- DM**

## 68000 Microcomputer Einführung

Verfasser: E. Flögel

198 Seiten

Eine leicht verständliche Anleitung zur Programmierung des leistungsfähigen 16 Bit Microcomputers von Motorola. Ein Lehr-, Lern- und Arbeitsbuch wie es eigentlich immer neben Ihrem Computer liegen sollte. Sehr viele anschauliche Beispiele und plausible, echt verständliche Erklärungen der Zusammenhänge. Eine Befehlsübersicht erleichtert das Nachschlagen von Befehlen und deren Funktion und Anwendung. Ideal für den PC Besitzer und den Industrieanwender. Geeignet für: Macintosh und Sinclair QL. Am Schluß des Buches finden Sie eine übersichtliche Zusammenstellung der Befehle mit Beispielen. Eines der besten und preiswertesten Bücher über 68000 weltweit !

**Best.-Nr. 25**

**39,-- DM**

**E. Flögel**

**=68000=**

## Programmier Handbuch

**Einführung,  
Programmbeispiele**

**HOFACKER**

# Weitere interessante Bücher von Hofacker:

Best.-Nr.	Titel	Preis DM	Best.-Nr.	Titel	Preis DM			
<b>BÜCHER in deutscher Sprache</b>								
1	Transistor Berechnungs- und Bauanleitungsbuch-1	29,80	146	Hardware Erweiterungen für den Commodore-64	39,00			
2	Transistor Berechnungs- und Bauanleitungsbuch-2	19,80	147	Beherrschen Sie Ihren Commodore-64	19,80			
3	Elektronik im Auto	9,80	148	Programmierhandbuch für SHARP	49,00			
4	IC-Handbuch, TTL, CMOS, Linear	19,80	149	Programme für TI 99/4A	49,00			
5	IC-Datenbuch, TTL, CMOS, Linear	9,80	175	Astrologie auf dem ATARI 800	49,00			
6	IC-Schaltungen, TTL, CMOS, Linear	19,80	187	Mehr als 29 Programme für den Commodore-64	29,80			
7	Elektronik Schaltungen	19,80	188	Statistik in BASIC	39,00			
8	IC-Bauanleitungs-Handbuch	19,80	189	Editor + Assembler, Macrofire	19,80			
9	Feldleitfettransistoren	9,80	190	Das große Spielebuch für ATARI 600XL/800XL, I	29,80			
10	Elektronik und Radio	19,80	200	FORTH-Anwendungsbeispiele	49,00			
12	Beispiele integrierter Schaltungen (BIS)	19,80	202	UNIX – Grundlagen und Anwendungen	39,00			
13	HEH, Hobby Elektronik Handbuch	9,80	204	Grafik und Ton mit dem Commodore-64	29,80			
16	CMOS Teil 1, Einführung, Entwurf, Schaltbeispiele	19,80	205	Das große Spielebuch für ATARI 600XL/800XL, II	29,80			
17	CMOS Teil 2, Entwurf und Schaltbeispiele	19,80	210	Superprogramme für IBM-PC	29,80			
18	CMOS Teil 3, Entwurf und Schaltbeispiele	19,80	212	Geschäftsprogramme für Commodore-64	39,00			
19	IC-Experimentier Handbuch	19,80	213	Technische Gleichungssysteme in BASIC	49,00			
20	Operationsverstärker	19,80	217	Künstliche Intelligenz	19,80			
21	Digitaltechnik Grundkurs	19,80	218	Produktivität rauft mit OPEN-ACCESS	39,00			
22	Mikroprozessoren, Eigenschaften und Aufbau	19,80	220	Tabellenkalkulation für blutige Laien (I.V.)	19,80			
23	Elektronik Grundkurs, Kurzlehrgang Elektronik	9,80	221	SYMPHONY-Anwendungen	29,80			
24	Programmieren in MaschinenSprache mit Z80, II	29,80	222	Praktische Anwendungen mit dem Sinclair QL	29,80			
25	68000 Microcomputer Einführung	39,00	223	MODULA-2 Anwender Handbuch	49,00			
26	Mikroprozessor, Teil 2	19,80	224	Anwenderprogramme für Apple II c, und II e	19,80			
27	BASIC-M für 68000/09/68000 (Motorola)	29,80	227	Die große Starparade	39,00			
28	Lexikon u. Wörterbuch f. Elektr. u. Mikroprozessor	29,80	8029 Z-80 Assembler-Handbuch	29,80				
30	Floppy Disk Selbstbau-Handbuch (I.V.)	49,00	<b>BÜCHER in englischer Sprache</b>					
31	57 Praktische Programme in BASIC	39,00	29	Hardware Handbook	49,00			
32	ATARI BASIC, für Selbststudium und Praxis	39,00	52	Small Business Programs for the IBM PC	29,80			
33	Microcomputer Programmierbeispiele	19,80	151	8K Microsoft BASIC Reference Manual	9,80			
34	TINY-BASIC Handbuch	19,80	152	Expansion Handbook for 6502 and 6800	19,80			
35	Der freundliche Computer	29,80	156	Small Business Programs	29,80			
102	Mathematische + Wissenschaftliche Progr. I. BASIC	29,80	158	The Second Book of Ohio Scientific	19,80			
103	Oszilloskop-Handbuch	19,80	159	The Third Book of Ohio Scientific	29,80			
104	Portable Computer Handbuch (I.V.)	39,00	160	The Fourth Book of Ohio Scientific	29,80			
108	Rund um den Spectrum (Progr., Tips + Tricks)	29,80	161	The Fifth Book of Ohio Scientific	19,80			
109	6502 Microcomputer Programmierung	29,80	162	ATARI Games in BASIC	19,80			
111	Programmieren mit TRS-80 (GENIE)	29,80	164	ATARI-BASIC Learning by Using	19,80			
112	PASCAL-Programmier-Handbuch	29,80	166	Programming in 6502 Machinelanguage PET/CMB	49,00			
113	BASIC-Programmier-Handbuch (mit BASIC-Kurs)	19,80	169	How the Progr. y. ATARI i. 6502 Machinelanguage	29,80			
114	Der Microcomputer im Kleinbetrieb	39,80	170	FORTH on the ATARI – Learning by Using	29,80			
116	Einführung 16-Bit Microcomputer	29,80	171	See the Future with your ATARI (Astrology)	49,00			
118	Programmieren in MaschinenSprache mit dem 6502	49,00	172	Hackerbook I (Tricks + Tips for your ATARI)	29,80			
119	Programmieren in MaschinenSprache mit Z80, I	39,00	173	PD-Program Descriptions (ATARI)	9,80			
120	Anwenderprogramme für TRS-80 und GENIE	29,80	174	ZX-81/TIMEX Progr. i. BASIC a. Machine Lang.	5,00			
121	Microsoft BASIC-Handbuch	29,80	176	Programs + Tricks for VIC's	29,80			
122	BASIC für Fortgeschrittene	39,00	177	CP-M – MBASIC and the OSBORNE	5,00			
123	IEC-Bus Handbuch	19,80	178	The APPLE in Your Hand	39,00			
124	Progr. in MaschinenSprache mit Commodore-64	29,80	182	The Great Book of Games Vol. I – Games f. the C-64	29,80			
127	Einführung i.D. Microcomputer-Progr. mit 6800	49,00	183	More on the Sixtyfour (Commodore-64)	39,00			
128	Programmieren mit dem CBM	29,80	184	How to Progr. your C-64 i. 6502/10 Machinelang.	29,80			
130	Programmierbeispiele für CBM	9,80	185	Commodore-64 Tune-up	39,00			
132	CP-M Handbuch	19,80	186	Small Business Programs for the Commodore-64	29,80			
133	Handbuch für MS-DOS, IBM-PC	29,80	Der HOFAKER-Verlag produziert und vertreibt neben einer sehr großen Auswahl an Fachbüchern für Elektronik und Microcomputertechnik noch:					
136	Funktionsanalyse	79,00	Leerplatten und Bauanleitungen für Zusatzeinrichtungen für Ihren Personalcomputer, sowie					
137	FORTH – Grundlagen, Einführung, Beispiele	49,00	Programme (Software) und Leercassetten (C-10) für die bedeutenden Personalcomputer.					
139	BASIC für blutige Laien (speziell f. TRS-80, Genie)	19,80	(i.V. bedeutet: Buch ist in Vorbereitung)					
140	Progr. i. BASIC u. Maschinencode mit dem ZX81	29,80						
141	Progr. f. VC-20 (Spiele, Utilities, Erweiterungen)	29,80						
143	35 Programme für den ZX81	29,80						
144	33 Programme für den ZX-Spectrum	29,80						
145	64 Programme für den Commodore-64	39,00						

**HOFAC** 536  
SINGAPORE DM 29.80  
COMPUTER STUDIO LOS ANGELES

ISBN 3-88963-222-X

HOLZKIRCHEN