Inhaltsverzeichnis zur Anleitung zum TOOL4
****

Erläuterungen zur synthetischen Programmierung . . . . Seite 4-2
CLEN . . . . . Seite 4-3
CONV . . . . . Seite 4-4
CONVL$ . . . . . Seite 4-5
CONVS$ . . . . . Seite 4-6
ELAST . . . . . Seite 4-7
ELINE . . . . . Seite 4-8
ELIST$ . . . . . Seite 4-9
ENTER . . . . . Seite 4-10
Beispiele zur synthetischen Programmierung . . . . . Seite 4-11

Erläuterungen zu statistischen Berechnungen . . . . Seite 4-14
AVGX / AVGY . . . . . Seite 4-15
CORR . . . . . Seite 4-16
GRA . . . . . Seite 4-17
SDNX / SDNY . . . . . Seite 4-18
SDVX / SDVY . . . . . Seite 4-19
SEG . . . . . Seite 4-20
STAT . . . . . Seite 4-21
STATCLR . . . . . Seite 4-22
STATIN . . . . . Seite 4-23
STATOFF . . . . . Seite 4-24
STATON . . . . . Seite 4-25
STATOUT . . . . . Seite 4-26
STX / STY . . . . . Seite 4-27
Beispiele zu statistischen Berechnungen . . . . . Seite 4-28

Liste der Instruktionen . . . . . Seite 4-30

Erläuterungen zur synthetischen Programmierung
****

Einführung

Unter "synthetischer Programmierung" ist die Erzeugung oder Veränderung eines ausführbaren Programmes durch ein anderes zu verstehen.
Die Befehle des TRAMsoft TOOL4 ermöglichen das Lesen und Speichern von BASIC-Programm-Zeilen während der Programmausführung.

BASIC-Schlüsselworte

Im SHARP PC-1500/A werden aus Platzgründen alle BASIC-Schlüsselworte wie PRINT, INPUT, GOSUB usw. in sogenannte 2-Byte-Token übersetzt und als solche im Speicher abgelegt. Variablen-Namen, Texte und Sonderzeichen werden dagegen im ASCII-Code gespeichert (siehe Anhang A-3/A-4). Wird ein Programm in die Anzeige gebracht oder auf einem Drucker ausgegeben, so werden die Token wieder in die entsprechenden Zeichenfolgen zurück übersetzt.
Eine in Token übersetzte Programm-Zeile kann, wie eine BASIC-Variable, maximal 80 Bytes lang sein. Eine zurück übersetzte Programm-Zeile kann dagegen wesentlich länger werden. Die Funktion ELIST$ liefert deshalb die unübersetzte Programm-Zeile, damit diese einer Variablen zugewiesen werden kann. Daneben stehen Funktionen zur Verfügung, um die Token in einer Programm-Zeile aufzufinden und in die zugehörige Zeichenfolge zu übersetzen.

Voraussetzungen

Die Anweisungen zur synthetischen Programmierung beziehen sich immer auf das "aktive Programm-Modul" (siehe Allgemeines, Seite iii).

Es können nur Programm-Zeilen verändert, gelöscht oder hinzugefügt werden, die sich im Speicher hinter der zur Zeit ausgeführten Programm-Zeile befinden. Diese Bedingung ist erfüllt, wenn

- sich die zu bearbeitende Programm-Zeile und die zur Zeit ausgeführte Programm-Zeile im gleichen Programm-Modul befinden und die zu bearbeitende Programm-Zeile eine höhere Zeilennummer aufweist als die ausgeführte Programm-Zeile

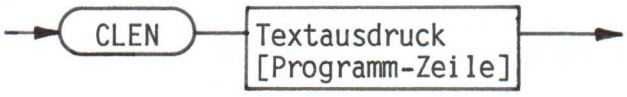
oder

- sich die zu bearbeitende Programm-Zeile in einem Programm-Modul befindet, das nachträglich mit MERGE, FLOAD P (TOOL 2), DLOAD P (Floppy) oder APPEND (TOOL 1) zum Programm mit der zur Zeit ausgeführten Programm Zeile hinzugefügt wurde.

CLEN

```
*************************************************************
*                                                           *
*  Liefert die Länge einer übersetzten Programm-Zeile.      *
*                                                           *
*************************************************************
```

Syntax



Ergebnis: numerischer Wert

Parameter-Beschreibung: - [Programm-Zeile]: Nicht übersetzte BASIC-Zeile.

Hinweise: - Enthält die Programm-Zeile kein BASIC-Schlüsselwort, so liefern CLEN und LEN das gleiche Resultat.

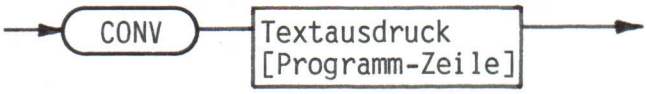
- CLEN kann manuell oder programmkontrolliert ausgeführt werden.

- Beispiel siehe Seite 4-11.

CONV

```
*************************************************************
*                                                           *
*  Liefert die Anzahl Elemente in einer Programm-Zeile.     *
*                                                           *
*************************************************************
```

Syntax



Ergebnis: numerischer Wert

Parameter-Beschreibung: - [Programm-Zeile]: Nicht übersetzte BASIC-Zeile.

Erläuterung

Ein Element ist:

- Ein einzelnes Zeichen eines Variablen-Namen oder Textes, sowie Sonderzeichen
- Ein Token eines BASIC-Schlüsselwortes

Hinweise: - Das Resultat der Funktion CONV kann als Schleifenzähler beim schrittweisen Uebersetzen einer Programm-Zeile verwendet werden.

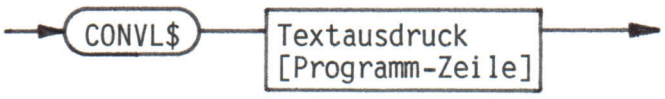
- CONV kann manuell oder programmkontrolliert ausgeführt werden.

- Beispiel siehe Seite 4-11.

CONVL$

```
*************************************************************
*                                                           *
*  Uebersetzt alle Token in einer Programm-Zeile in die     *
*  entsprechenden Zeichenfolgen.                            *
*                                                           *
*************************************************************
```

Syntax



Ergebnis: Text

Parameter-Beschreibung: - [Programm-Zeile]: Nicht übersetzte BASIC-Zeile.

Hinweise: - CONVL$ übersetzt alle Token in die entsprechnenden Zeichenfolgen, alle übrigen Zeichen bleiben unverändert.

- Token von nicht angeschlossenen Zusatzgeräten zum PC-1500 werden wie bei LIST, bzw. LLIST mit einer "Wellenlinie" übersetzt.

- CONVL$ kann manuell oder programmkontrolliert ausgeführt werden.

- Beispiel siehe Seite 4-11.

Besondere Fehlermeldung: - ERROR 67: Die übersetzte Programm-Zeile würde länger als 80 Zeichen.

CONVS$

```
*************************************************************
*                                                           *
*  Uebersetzt ein einzelnes Element in einer Programm-      *
*  Zeile.                                                   *
*                                                           *
*************************************************************
```

Syntax

Ergebnis: Text

Parameter-Beschreibung: - [Programm-Zeile]: Nicht übersetzte BASIC-Zeile.

- [Position]: Stelle des zu übersetzenden Elementes.

Hinweise:
- CONVS$ übersetzt ein einzelnes Element in einer Programm-Zeile.
- Ein Element ist ein einzelnes Zeichen oder ein Token.
- Ein einzelnes Zeichen bleibt unverändert.
- Ein Token wird in die entsprechende Zeichenfolge übersetzt.
- Token von nicht angeschlossenen Zusatzgeräten zum PC-1500 werden wie bei LIST, bzw. LLIST mit einer "Wellenlinie" übersetzt.
- CONVS$ kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-11.

ELAST

```
*************************************************************
*                                                           *
*  Liefert die Nummer der letzten Programm-Zeile.           *
*                                                           *
*************************************************************
```

Syntax

Ergebnis: numerischer Wert

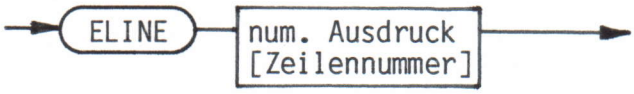
Hinweise: - ELAST bezieht sich immer auf das aktive Programm-Modul.

- ELAST kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-11.

ELINE

```
*************************************************************
*                                                           *
*  Liefert die Nummer der angegebenen oder nächst höheren   *
*  Programm-Zeile.                                          *
*                                                           *
*************************************************************
```

Syntax



Ergebnis: numerischer Wert

Parameter-Beschreibung: - [Zeilennummer]: num. Ausdruck im Bereich der möglichen Zeilennummern.

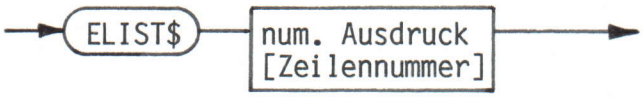
Hinweise:
- ELINE dient dazu, die Zeilennummer der nachfolgenden Programmzeile festzustellen.
- ELINE bezieht sich immer auf das aktive Programm-Modul.
- ELINE kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-11.

Besondere Fehlermeldung: - ERROR 11: Die angegebene Zeilennummer ist grösser als die höchste Zeilennummer im aktiven Programm-Modul.

ELIST$

```
*************************************************************
*                                                           *
*  Liefert die unübersetzte Programm-Zeile.                 *
*                                                           *
*************************************************************
```

Syntax



Ergebnis: Text

Parameter-Beschreibung: - [Zeilennummer]: Zeilennummer der aus dem Programm-Speicher auszulesenden BASIC-Zeile. Existiert die angegebene Zeilennummer nicht, so wird die Zeile mit der nächst höheren Zeilennummer ausgelesen.

Hinweise: - ELIST$ bezieht sich immer auf das aktive Programm-Modul.

- ELIST$ kann manuell oder programmkontrolliert ausgeführt werden.

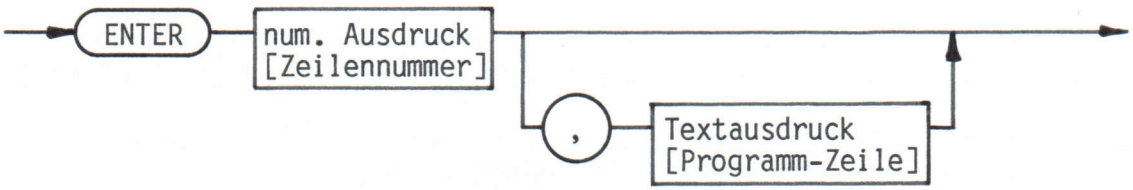
- Beispiel siehe Seite 4-11.

Besondere Fehlermeldung: - ERROR 11: Die angegebene Zeilennummer ist grösser als die höchste Zeilennummer im aktiven Programm-Modul.

ENTER

```
*************************************************************
*                                                           *
*  Speichert oder löscht eine Programm-Zeile.               *
*                                                           *
*************************************************************
```

Syntax



Parameter-Beschreibung:
- [Zeilennummer]: Zeilennummer der zu bearbeitenden Programm-Zeile.
- [Programm-Zeile]: BASIC-Zeile, die ins Programm eingefügt werden soll.
  Wird keine Programm-Zeile angegeben, so wird die entsprechende Zeile aus dem Speicher gelöscht.

Achtung
****

Es dürfen nur Programm-Zeilen verändert werden, die sich hinter der zur Zeit ausgeführten Programm-Zeile befinden (siehe Erläuterungen zur synthetischen Programmierung, Seite 4-2).

Hinweise:
- ENTER bezieht sich immer auf das aktive Programm-Modul.
- ENTER schaltet die WAIT-Automatik ab und setzt die Cursor-Position zurück, d.h. ENTER hat die Nebenwirkung WAIT :PRINT.
- ENTER kann nur programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-11.

Besondere Fehlermeldungen:
- ERROR 67: Zeilennummer und Programm-Zeile ergeben einen Ausdruck von mehr als 79 Bytes.
- ERROR 178: Zum Abspeichern der angegebenen Programm-Zeile steht nicht mehr genügend Speicherplatz zur Verfügung. Die Variabeln bleiben erhalten !

Beispiele zur synthetischen Programmierung
****

Beispiel 1

BASIC-Programm-Zeile:

```
10 "A"INPUT "X = ";X:Y=SIN X:GOTO 200
```

LEN ELIST$ 100 liefert den Zahlenwert 25.
CLEN ELIST$ 100 liefert den Zahlenwert 34.

Erklärung:

Die Programm-Zeile enthält 3 BASIC-Schlüsselworte: INPUT, SIN und GOTO.

Die Funktion LEN interpretiert jedes Token als 2 Zeichen, die Funktion CLEN übersetzt dagegen jedes Token und setzt die entsprechende Anzahl Zeichen für jedes Schlüsselwort ein.

Anmerkung:

Enthält eine Programm-Zeile kein BASIC-Schlüsselwort, so liefern LEN und CLEN das gleiche Resultat.

Beispiel 2

Ein Programm zur Erstellung von Wertetabellen soll neben dem Startwert xmin, dem Endewert xmax und der Schrittbreite dx auch die zu berechnende Formel y=f(x) interaktiv abfragen. Die eingegebene Formel soll in den Programmspeicher abgelegt werden. Anschliessend wird die Wertetabelle berechnet.

```
 10 "B"CLEAR :DIM FO$(0)*80
 20 INPUT "y=f(x): ";FO$(0)
 30 INPUT "xmin = ";U
 40 INPUT "xmax = ";O
 50 INPUT "dx   = ";D
 60 ENTER 200,FO$(0)+":RETURN"
 70 X=U
 80 GOSUB 200
 90 PRINT X,Y
100 X=X+D:IF X<=OTHEN 80
110 END
```

Wird die Frage "y=f(x):" mit "Y=SINX*COSX" beantwortet, so sieht das Programm anschliessend wie folgt aus:

```
 10 "B"CLEAR :DIM FO$(0)*80
 20 INPUT "y=f(x): ";FO$(0)
 30 INPUT "xmin = ";U
 40 INPUT "xmax = ";O
 50 INPUT "dx   = ";D
 60 ENTER 200,FO$(0)+":RETURN"
 70 X=U
 80 GOSUB 200
 90 PRINT X,Y
100 X=X+D:IF X<=OTHEN 80
110 END
200 Y=SIN X*COS X:RETURN
```

Beispiele zur synthetischen Programmierung (Fortsetzung)

Beispiel 3

Das nachfolgende Beispiel löscht alle mit "REM" beginnenden Zeilen im gleichen Programm-Modul.

```
  1 "C"CLEAR :L=10
  2 IF CONVS$ (ELIST$ L,1)="REM "ENTER L
  3 IF L<ELAST LET L=ELINE (L+1):GOTO 2
  4 END
 10 REM *** quadratische gleichung ***
 20 REM *** ====================== ***
 30 REM
 40 REM *** eingabe ***
 50 INPUT "A = ";A
 60 INPUT "B = ";B
 70 INPUT "C = ";C
100 REM berechnung der diskriminante
110 D=B*B-4*A*C
120 IF D=0THEN 310
130 IF D<0THEN 410
200 REM 2 reelle loesungen
210 X1=(-B+SQR D)/(2*A):X2=(-B-SQR D)/(2*A)
220 PRINT X1,X2:END
300 REM 1 doppel-loesung
310 X1=-B/(2*A)
320 PRINT X1:END
400 REM 2 komplexe loesungen
410 X1=-B/(2*A):X2=SQR -D/(2*A)
420 PRINT X1,X2:END
```

Nach Ausführung der Programm-Zeilen 1 ... 4 befindet sich noch folgendes im Programm-Speicher:

```
  1 "C"CLEAR :L=10
  2 IF CONVS$ (ELIST$ L,1)="REM "ENTER L
  3 IF L<ELAST LET L=ELINE (L+1):GOTO 2
  4 END
 50 INPUT "A = ";A
 60 INPUT "B = ";B
 70 INPUT "C = ";C
110 D=B*B-4*A*C
120 IF D=0THEN 310
130 IF D<0THEN 410
210 X1=(-B+SQR D)/(2*A):X2=(-B-SQR D)/(2*A)
220 PRINT X1,X2:END
310 X1=-B/(2*A)
320 PRINT X1:END
410 X1=-B/(2*A):X2=SQR -D/(2*A)
420 PRINT X1,X2:END
```

Erläuterungen zu statistischen Berechnungen
****

TOOL4 stellt verschiedene Anweisungen und Funktion zur Erfassung und Auswertung statistischer Werte zur Verfügung.
Um statistische Berechnungen durchführen zu können, müssen die Anzahl Wertepaare, die Summe, die Quadratsumme und die Summe der Produkte dieser Werte-Paare gespeichert werden. Zu diesem Zweck wird mit STATON eine spezielle DIM-Variable angelegt.
Diese Statistik-Variable wird so angelegt, dass sie aus dem BASIC nicht angesprochen werden kann, d.h. es besteht keine Gefahr, dass sie mit einer anderen DIM-Variablen in konflikt gerät. Die in der Statistik-Variablen gespeicherten Zahlenwerte können mit der Funktion STAT n jederzeit angezeigt werden.

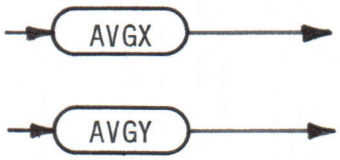
Für statistische Auswertungen stehen folgende Funktionen zur Verfügung:

- Berechnung der Durchschnittswerte von x, bzw. y (AVGX, bzw. AVGY)
- Berechnung der Standard-Abweichung mit dem Ganzheitsparameter n (SDNX, bzw. SDNY)
- Berechnung der Standard-Abweichung mit dem Ganzheitsparameter n-1 (SDVX, bzw. SDVY)
- Berechnung der Korrelation (CORR)
- Berechnung der Steigung der linearen Regression (GRA)
- Berechnung des y-Achsen-Abschnittes der linearen Regression (SEG)
- Berechnung eines statistischen Wertes auf Grund einer vorgegebenen Koordinate (STX, bzw. STY)

AVGX / AVGY

```
*************************************************************
*                                                           *
*  Berechnet den Durchschnittswert von x, bzw. y.           *
*                                                           *
*************************************************************
```

Syntax



Ergebnis: numerischer Wert

Formel: (wobei E = Summe)

$$AVGX = \frac{Ex}{n} \qquad AVGY = \frac{Ey}{n} \qquad \text{für } n \neq 0$$

Hinweise: - AVGX und AVGY können manuell oder programmkontrolliert ausgeführt werden.

- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung: - ERROR 6: Die Statistik-Variable existiert nicht.

CORR

```
*************************************************************
*                                                           *
*  Berechnet die Korrelation der eingegebenen Wertepaare    *
*  gegenüber der linearen Regression.                       *
*                                                           *
*************************************************************
```

Syntax

Ergebnis: numerischer Wert

Formel: (wobei E = Summe)

$$CORR = \frac{Exy - \frac{Ex*Ey}{n}}{SQR((Ex^2 - \frac{(Ex)^2}{n} *(Ey^2 - \frac{(Ey)^2}{n}))} \qquad \text{für } n \neq 0$$

Erläuterung

Der Absolutwert der Korrelation ist ein Mass für die Abweichung der linearen Regression von den tatsächlichen Werten. Sie ermöglicht deshalb eine Aussage über die Genauigkeit der entsprechenden Statistik.

1.0: Absolute Genauigkeit
0.8 .. 1.0: Gute Genauigkeit
0.6 .. 0.8: Mässige Genauigkeit
0.6 .. 0.0: Schlechte Genauigkeit

Eine negative Korrelation bedeutet, dass sich die beiden Grössen umgekehrt Porportional zueinander verhalten (negative Regression).

Hinweise: - CORR kann manuell oder programmkontrolliert ausgeführt werden.

- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung: - ERROR 6: Die Statistik-Variable existiert nicht.

GRA

```
*************************************************************
*                                                           *
*  Berechnet die Steigung der linearen Regression.          *
*                                                           *
*************************************************************
```

Syntax

Ergebnis: numerischer Wert

<u>Formel:</u> (wobei E = Summe)

$$GRA = \frac{Exy - \frac{Ex*Ey}{n}}{Ex^2 - \frac{(Ex)^2}{n}} \quad \text{für } n \neq 0$$

Hinweise: - GRA kann manuell oder programmkontrolliert ausgeführt werden.

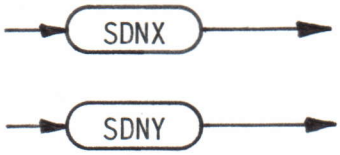
- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung: - ERROR 6: Die Statistik-Variable existiert nicht.

SDNX / SDNY

```
*************************************************************
*                                                           *
*  Berechnet die Standardabweichungen für x, bzw. y mit     *
*  dem Ganzheits-Parameter n.                               *
*                                                           *
*************************************************************
```

Syntax



Ergebnis: numerischer Wert

<u>Formel:</u> (wobei E = Summe)

$$SDNX = SQR\left(\frac{Ex^2 - n*AVGX^2}{n}\right) \qquad SDNY = SQR\left(\frac{Ey^2 - n*AVGY^2}{n}\right) \quad \text{für } n \neq 0$$

Hinweise: - SDNX / SDNY kann manuell oder programmkontrolliert ausgeführt werden.

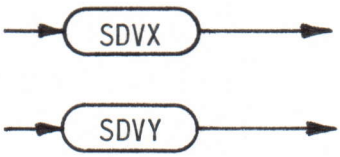
- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung: - ERROR 6: Die Statistik-Variable existiert nicht.

SDVX / SDVY

```
*************************************************************
*                                                           *
*  Berechnet die Standardabweichungen für x, bzw. y mit     *
*  dem Ganzheits-Parameter n-1.                             *
*                                                           *
*************************************************************
```

Syntax



Ergebnis: numerischer Wert

<u>Formel:</u> (wobei E = Summe)

$$SDVX = SQR\left(\frac{Ex^2 - n*AVGX^2}{n-1}\right) \qquad SDVY = SQR\left(\frac{Ey^2 - n*AVGY^2}{n-1}\right) \quad \text{für } n \neq 0$$

Hinweise:
- SDVX / SDVY kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung: - ERROR 6: Die Statistik-Variable existiert nicht.

SEG

```
*************************************************************
*                                                           *
* Berechnet den y-Achsenabschnitt der linearen Regression.  *
*                                                           *
*************************************************************
```

Syntax

Ergebnis: numerischer Wert

Formel:

$$SEG = AVGY - GRA * AVGX \quad \text{für } n \neq 0$$

Hinweise: - SEG kann manuell oder programmkontrolliert ausgeführt werden.

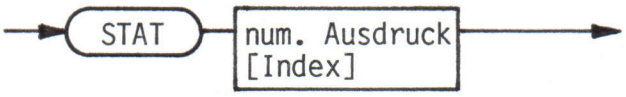
- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung: - ERROR 6: Die Statistik-Variable existiert nicht.

STAT

```
*************************************************************
*                                                           *
*  Liefert den Inhalt der Statistik-Variablen.              *
*                                                           *
*************************************************************
```

Syntax



Ergebnis: numerischer Wert

Parameter-Beschreibung: - [Index]: numerischer Ausdruck im Bereich 0 ... 5.
0: n = Anzahl Elemente
1: Ex = Summe x
2: Ex^2 = Summe x^2
3: Exy = Summe x*y
4: Ey = Summe y
5: Ey^2 = Summe y^2

Hinweise: - Mit STAT können die Werte der Statistik-Variablen nur gelesen, aber nicht verändert werden.

- STAT kann manuell oder programmkontrolliert ausgeführt werden.

- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung: - ERROR 6: Die Statistik-Variable existiert nicht.

STATCLR

```
*************************************************************
*                                                           *
*  Setzt die Statistik-Variable null.                       *
*                                                           *
*************************************************************
```

Syntax

Hinweise: - STATCLR kann manuell oder programmkontrolliert ausgeführt werden.

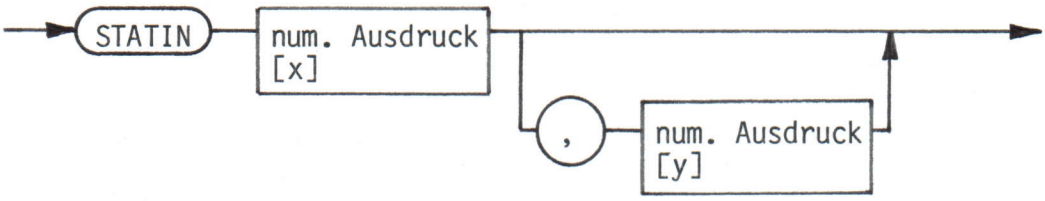
- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung: - ERROR 6: Die Statistik-Variable existiert nicht.

STATIN

```
*************************************************************
*                                                           *
*  Fügt der Statistik-Variablen ein Wertepaar bei.          *
*                                                           *
*************************************************************
```

Syntax



Parameter-Beschreibung:
- [x]: Zahlenwert der x-Koordinate.
- [y]: Zahlenwert der y-Koordinate.
  Wird nichts angegeben, so wird null eingesetzt.

Hinweise:
- STATIN führt die Statistik-Variable nach (Zähler und Summen).
- Ist einer der beiden num. Ausdrücke fehlerhaft, so wird die Verarbeitung abgebrochen, bevor die Statistik-Variable nachgeführt wurde.
- Tritt beim nachführen der Summen ein Fehler auf, so ist die Statistik-Variable nur teilweise nachgeführt.
- STATIN kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung:
- ERROR 6: Die Statistik-Variable existiert nicht.

STATOFF

```
*************************************************************
*                                                           *
*  Löscht die Statistik-Variable.                           *
*                                                           *
*************************************************************
```

Syntax

Hinweise: - STATOFF kann manuell oder programmkontrolliert ausgeführt werden.

- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung: - ERROR 6: Die Statistik-Variable existiert nicht.

STATON

```
*************************************************************
*                                                           *
*  Legt die Statistik-Variable an.                          *
*                                                           *
*************************************************************
```

Syntax

Hinweise:
- Die Statistik-Variable entspricht der DIM-Variablen sT(5) und und benötigt 55 Byte RAM.
- Die Statistik-Variable wird mit CLEAR, NEW oder PURGE DIM (TOOL3) gelöscht.
- STATON kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-28.

Hinweis für TOOL3-Benützer:

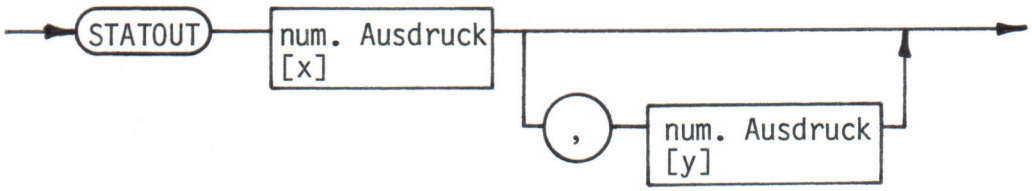
- Die Statistik-Variable kann nach CLEAR, NEW oder PURGE DIM mit VKEEP wieder adressierbar gemacht werden.

Besondere Fehlermeldungen:
- ERROR 5: Die Statistik-Variable existiert bereits.
- ERROR 10: Zum Anlegen der Statistik-Variable steht nicht mehr genügend Speicherplatz zur Verfügung.

STATOUT

```
*************************************************************
*                                                           *
*  Entfernt ein Wertepaar aus der Statistik-Variablen.      *
*                                                           *
*************************************************************
```

Syntax



Parameter-Beschreibung:
- [x]: Zahlenwert der x-Koordinate.
- [y]: Zahlenwert der y-Koordinate.
  Wird nichts angegeben, so wird null eingesetzt.

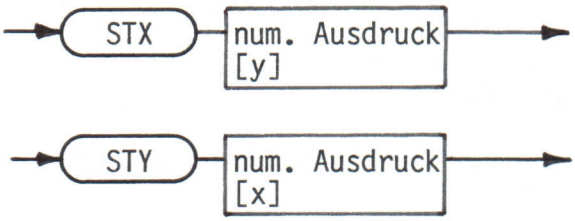
Hinweise:
- Wurde nur ein Zahlenwert eines Wertepaares falsch eingegeben, so müssen trotzdem beide Werte angegeben werden.
- STATOUT führt die Statistik-Variable nach (Zähler und Summen).
- Ist einer der beiden num. Ausdrücke fehlerhaft, so wird die Verarbeitung abgebrochen, bevor die Statistik-Variable nachgeführt wurde.
- Tritt beim nachführen der Summen ein Fehler auf, so ist die Statistik-Variable nur teilweise nachgeführt.
- STATOUT kann manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung:
- ERROR 6: Die Statistik-Variable existiert nicht.

STX / STY

```
*************************************************************
*                                                           *
*  Berechnet zu einer vorgegebenen Koordinate den ent-      *
*  sprechenden statistischen Wert.                          *
*                                                           *
*************************************************************
```

Syntax



Ergebnis: numerischer Wert

Parameter-Beschreibung:
- [y]: y-Koordinate, zu der die entsprechende x-Koordinate berechnet werden soll.
- [x]: x-Koordinate, zu der die entsprechende y-Koordinate berechnet werden soll.

Formel:

$$STX = \frac{y\text{-}SEG}{GRA} \qquad STY = GRA * x + SEG$$

Hinweise:
- STX / STY können manuell oder programmkontrolliert ausgeführt werden.
- Beispiel siehe Seite 4-28.

Besondere Fehlermeldung: - ERROR 6: Die Statistik-Variable existiert nicht.

Beispiele zu statistischen Berechnungen
****

Beispiel 1

Es soll die durchschnittliche Niederschlagsmenge pro Monat ermittelt werden.
Es steht für jeden Monat die gefallene Niederschlagsmenge zur Verfügung.

Monat	Jan	Feb	Mär	Apr	Mai	Jun	Jul	Aug	Sep	Okt	Nov	Dez
Niederschlag	45	57	52	88	42	45	37	29	32	39	42	48

Eingabe der Werte:

STATON (enter)	Statistik-Variable anlegen
STATIN 45 (enter)	1. Wert eingeben
STATIN 57 (enter)	2. Wert eingeben
...	...
STATIN 48 (enter)	12. Wert eingeben

Anzahl eingegebene Werte abfragen:

STAT 0 (enter) liefert 12

Total gefallene Niederschlagsmenge:

STAT 1 (enter) liefert 556

Durchschnittliche Niederschlagsmenge pro Monat:

AVGX (enter) liefert 46.33

Beispiele zu statistischen Berechnungen

Beispiel 2

Bei 12 Personen wurde Körpergrösse und Schuhnummer aufgenommen. Es soll nun der statistische Zusammenhang untersucht werden.

Körpergrösse	184	171	179	160	170	192	195	165	182	165	195	180
Schuhnummer	44	36	37	37	41	44	43	37	44	38	43	41

Eingabe der Wertepaare:

STATCLR (enter)	Statistik-Variable null setzen
STATIN 184,44 (enter)	1. Wertepaar eingeben
STATIN 174,36 (enter)	2. Wertepaar eingeben
STATOUT 174,36 (enter)	fehlerhaftes Wertepaar löschen
STATIN 171,36 (enter)	2. Wertepaar korrekt eingeben
.....	.....
STATIN 180,41 (enter)	12. Wertepaar korrekt eingeben

Anzahl Eingaben kontrollieren:
STAT 0 (enter) liefert 12

Korrelation der Werte berechnen:
CORR (enter) liefert 0.787 mässige Korrelation, d.h. von der Körpergrösse kann nur bedingt auf die Schuhnummer geschlossen werden und umgekehrt.

Berechnung der Steigung und des y-Achsen-Abschnittes der linearen Regression der eingegebenen Wertepaare:
GRA (enter) liefert 0.209 Steigung
SEG (enter) liefert 3.226 y-Achsen-Abschnitt

Welche Schuhnummer hat ein Mensch mit Körpergrösse 180 ?
STY 180 (enter) liefert 41

Welche Körpergrösse hat ein Mensch mit Schuhnummer 38 ?
STX 38 (enter) liefert 167

Liste der Funktionen und Instruktionen
****

TOOL4 bietet zusätzlich folgende Funktionen und Instruktionen an:

BEFEHL	Abkürzung	Kurzbeschreibung	Seite
AVGX	AV.	Berechnet den Durchschnittswert von x.	4-14
AVGY	--	Berechnet den Durchschnittswert von y.	4-14
CLEN	--	Liefert die Länge einer übersetzten Programm-Zeile.	4-3
CONV	--	Liefert die Anzahl Elemente in einer Programm-Zeile.	4-4
CONVL$	CONV.	Uebersetzt alle Token in einer Programm-Zeile in die entsprechende Zeichenfolgen.	4-5
CONVS$	CONVS.	Uebersetzt ein einzelnes Element in einer Programm-Zeile.	4-6
CORR	COR.	Berechnet die Korrelation der eingegebenen Wertepaare gegenüber der linearen Regression.	4-16
ELAST	EL.	Liefert die Nummer der letzten Programm-Zeile.	4-7
ELINE	ELI.	Liefert die Nummer der angegebenen oder nächst höheren Programm-Zeile.	4-8
ELIST$	ELIS.	Liefert die unübersetzte Programm-Zeile.	4-9
ENTER	ENT.	Speichert oder löscht eine Programm-Zeile.	4-10
GRA	--	Berechnet die Steigung der linearen Regression.	4-17
SDNX	SD.	Berechnet die Standardabweichung für x mit dem Ganzheits-Parameter n.	4-18
SDNY	--	Berechnet die Standardabweichung für y mit dem Ganzheits-Parameter n.	4-19
SDVX	SDV.	Berechnet die Standardabweichung für x mit dem Ganzheits-Parameter n-1.	4-19
SDVY	--	Berechnet die Standardabweichung für y mit dem Ganzheits-Parameter n-1.	4-19
SEG	SE.	Berechnet den y-Achsenabschnitt der linearen Regression.	4-20

BEFEHL	Abkürzung	Kurzbeschreibung	Seite
STAT	--	Liefert den Inhalt der Statistik-Variablen.	4-21
STATCLR	STATC.	Setzt die Statistik-Variable null.	4-22
STATIN	STATI.	Fügt der Statistik-Variablen ein Wertepaar bei.	4-23
STATOFF	STATOF.	Löscht die Statistik-Variable.	4-24
STATON	STATO.	Legt die Statistik-Variable an.	4-25
STATOUT	STATOU.	Entfernt ein Wertepaar aus der Statistik-Variablen.	4-26
STX	--	Berechnet zu einer vorgegebenen y-Koordinate den entsprechenden x-Wert.	4-27
STY	--	Berechnet zu einer vorgegebenen x-Koordinate den entsprechenden y-Wert.	4-27