

Greifen Sie auf dieses Kapitel zurück, wenn Sie wissen, welche Art von Operation Sie durchführen wollen, jedoch nicht die Befehlsörter kennen. Nach Zuordnung der Befehlsörter zu jeder Art von Operation finden Sie in Kapitel 5 eine ausführliche Beschreibung jedes einzelnen Befehlswortes sowie je ein Beispiel, wie es angewandt wird.

## Inhaltsverzeichnis

Allgemeine Hinweise zur Programmierung . . . . .	4-1
Zuordnung von Werten zu Variablen . . . . .	4-3
Anzeige von Informationen . . . . .	4-4
Numerische Operationen . . . . .	4-6
String-Operationen . . . . .	4-8
Datenfelder . . . . .	4-10
Vergleichs- und logische Operatoren . . . . .	4-11
Control Statements . . . . .	4-13
Teil- und Unterprogramme . . . . .	4-14
Fehlerverarbeitung . . . . .	4-16
Abspeichern und Aufruf von Programmen . . . . .	4-17
Abspeichern und Aufruf von Datendateien . . . . .	4-18
Senden von Informationen an einen Drucker . . . . .	4-19

## ALLGEMEINE HINWEISE ZUR PROGRAMMIERUNG

Dieser Abschnitt beschreibt in Kurzform einige der Regeln und Eigenschaften des BASIC der TI-70 Serie.

### Zeilennummern

Jede Zeile in einem Programm muß mit einer Zeilennummer beginnen. Als Zeilennummer ist jede ganze Zahl im Bereich von 1 bis 32766 zulässig. Mit folgenden Befehlswörtern können Sie neue Zeilennummern schaffen und bereits bestehende abändern.

NUM	erzeugt automatisch Programmzeilennummern bei Eingabe eines Programms
REN	numeriert die Zeilen eines Programms neu.

## Duplizieren einer Programmzeile

Sie können eine Programmzeile duplizieren, indem Sie eine bereits vorhandene Zeile aufrufen und die alte Zeilennummer überschreiben. Dadurch erhalten Sie eine neue Zeile, ohne dabei die Originalzeile zu löschen.

## Zeilen mit mehreren Statements

Eine Programmzeile darf im allgemeinen mehr als ein Statement enthalten. Sie müssen jedoch durch einen Doppelpunkt getrennt sein.

```
100 X=3:PRINT X:PAUSE
```

Folgende Einschränkungen gelten für Zeilen mit mehreren Statements.

- Die Statements IMAGE oder DATA dürfen in einer Zeile mit mehreren Statements nicht angewendet werden.
- Wollen Sie das SUB Statement benutzen, so muß es als erstes Statement in der Zeile aufgeführt sein.
- Benutzen Sie eine Bemerkung (REM) in einer Zeile mit mehreren Statements, so muß diese das letzte Statement der Zeile sein.
- Benutzen Sie ein DIM Statement, so muß es als letztes Statement in der Zeile aufgeführt sein. (Das DIM Statement kann nur einmal pro Zeile verwendet werden.)

## Bemerkungen

Durch die Verwendung von Bemerkungen können Sie Erklärungen in Ihre Programme einfügen. Eine Bemerkung dient als Gedächtnisstütze für den Programmierer und wird während des Programmablaufs vom Computer ignoriert.

REM zeigt an, daß der Rest der Programmzeile eine Bemerkung ist.

kann anstelle des REM Befehlswortes gesetzt werden.

## Speichereinteilung

Das BASIC der TI-70 Serie verfügt über zwei Befehlsörter, mit denen Sie den Speicherbereich ansprechen können.

ADDMEM hängt die Speicherkapazität eines RAM Speichererweiterungsmoduls an den eingebauten Arbeitsspeicher des Computers an. (Benutzen Sie, da es sich um ein eingebautes Unterprogramm handelt, den Befehl CALL ADDMEM.)

FRE gibt die Gesamtmenge des verfügbaren Speichers zur Programm- oder Datenspeicherung an oder gibt Aufschluß über den Speicherplatz, der gerade vom laufenden Programm belegt wird.

## ZUORDNUNG VON WERTEN ZU VARIABLEN

Sie können Werte zu Variablen zuordnen von Daten, die entweder im Programm enthalten sind oder während des Programmablaufs über die Tastatur eingegeben werden.

## Variablennamen

Ein gültiger Variablenname muß folgenden Regeln entsprechen.

- Ein Variablenname kann aus bis zu 15 Zeichen bestehen.
- Das erste Zeichen muß ein Buchstabe, eine Unterstreichungslinie oder das AT-Zeichen sein. Die weiteren Zeichen können Buchstaben, Zahlen, Unterstreichungslinien oder AT-Zeichen sein.
- Der Name einer String-Variablen muß mit einem Dollarsymbol \$ enden. (Das \$-Symbol kann nicht als Teil des Namens einer numerischen Variablen verwendet werden.)

## In einem Programm enthaltene Daten

Mit folgenden Statements können Sie Werte zu Variablen zuordnen unter Verwendung von Daten, die in einem Programm enthalten sind.

READ	ordnet Werte zu Variablen zu durch sequentielles Lesen von Werten aus den DATA Statements.
DATA	bietet eine Reihe von Werten, die durch READ Statements Variablen zugeordnet werden.
RESTORE	entscheidet, welches DATA Statement als nächstes gelesen wird.

### Über Tastatur eingegebene Daten

Folgende Statements unterbrechen den Ablauf eines Programms und erlauben die Eingabe von Daten über die Tastatur.

INPUT	erlaubt die Eingabe von einem oder mehreren Werten. INPUT kann einen Dialog anzeigen.
LINPUT	erlaubt die Eingabe eines String-Wertes. Mit LINPUT werden alle Ihre Eingaben, einschließlich Kommas und Anführungszeichen unverändert akzeptiert.  LINPUT kann einen Dialog anzeigen.
ACCEPT	erlaubt die Eingabe eines Wertes. Das ACCEPT Statement bietet verschiedene Möglichkeiten für die Eingabe, die mit INPUT oder LINPUT nicht möglich sind.

## ANZEIGE VON INFORMATIONEN

Mit den Befehlswörtern in diesem Abschnitt können Sie Informationen in die Anzeige holen.

### Nicht formatierte Anzeige

Folgende Befehlswörter verwenden Sie, um numerische und String-Werte in die Anzeige zu schreiben.

PRINT	bringt eine Konstante, den Wert einer Variablen oder das Ergebnis eines Ausdrucks in die Anzeige. (Normalerweise verwenden Sie eine PAUSE nach einem PRINT Statement.)
DISPLAY	ist ähnlich dem PRINT Statement, bietet jedoch einige Möglichkeiten, die es vielseitiger anwenden lassen als PRINT. (Normalerweise verwenden Sie eine PAUSE nach einem DISPLAY Statement.)
PAUSE	unterbricht den Ablauf eines Programms für eine gewisse Zeit, um Ihnen die Möglichkeit zu geben, die Information in der Anzeige anzusehen. Ohne PAUSE würden die Ergebnisse zu schnell in der Anzeige erscheinen, um sie genauer anzusehen.

Mit INPUT und LINPUT können Sie ebenfalls einen auf die gewünschte Eingabe bezogenen Dialog oder eine Nachricht anzeigen.

### Formatierte Anzeige

Mit PRINT und DISPLAY haben Sie einige Möglichkeiten zu überprüfen, wie Informationen angezeigt werden.

USING	definiert einen formatierten String für angezeigte Informationen oder spezifiziert den in einem separaten IMAGE Statement enthaltenen formatierten String.
IMAGE	Ein formatierter String wird als einziges in eine Zeile gestellt und kann so von mehr als einem PRINT USING oder DISPLAY USING Statement herangezogen werden.
TAB	Setzt Informationen an eine eigens bestimmte Stelle in der Anzeige.

## NUMERISCHE OPERATIONEN

Das BASIC der TI-70 Serie beinhaltet eine Vielzahl mathematischer Operatoren und Funktionen.

### Arithmetische Operatoren

Folgende Operatoren können Sie in einem numerischen Ausdruck anwenden.

$A + B$	Addiert A und B.
$A - B$	Subtrahiert B von A.
$A * B$	Multipliziert A und B.
$A / B$	Dividiert A durch B.
$A ^ B$	erhebt A zur Potenz von B.

+ und - können auch als Präfix-Operatoren (Vorzeichen) zur Angabe von positiven oder negativen Werten verwendet werden, z.B. bei +4 oder -6.

Der Computer besitzt ebenfalls Vergleichs- und logische Operatoren, durch die der Vergleich zweier numerischer Ausdrücke möglich ist. Angaben zu diesen Operatoren finden Sie auf Seite 4-12.

### Hierarchie der Operatoren

Folgende Tabelle zeigt die Reihenfolge, nach der arithmetische Berechnungen durchgeführt werden.

Direktfunktionen + oder -

^

×, /

+, -

Ein in Klammern stehender Ausdruck hat Vorrang und wird vor den außerhalb der Klammern stehenden Operationen ausgeführt.

### Trigonometrische Funktionen

Mit folgenden Befehlswörtern werden die Winkleinheiten für trigonometrische Berechnungen ausgewählt. Alle Winkel werden entsprechend dem gesetzten Winkelmodus gemessen.

DEG	Wahl des Grad-Modus.
RAD	Wahl des Radiant-Modus.
GRAD	Wahl des Neugrad-Modus.

Beginnen Sie trigonometrische Berechnungen stets mit der Wahl des passenden Winkelmodus. Sie können dann folgende trigonometrische Funktionen anwenden.

ACOS	Berechnet den Arkuskosinus einer Zahl.
ASIN	Berechnet den Arkussinus einer Zahl.
ATN	Berechnet den Arkustangens einer Zahl.
COS	Berechnet den Kosinus eines Winkels.
SIN	Berechnet den Sinus eines Winkels.
TAN	Berechnet den Tangens eines Winkels.

### Hyperbel-Funktionen

Im BASIC der TI-70 Serie sind nachfolgende Hyperbel-Funktionen enthalten.

ACOSH	Berechnet einen hyperbolischen Arkuskosinus.
ASINH	Berechnet einen hyperbolischen Arkussinus.
ATANH	Berechnet einen hyperbolischen Arkustangens.

COSH	Berechnet einen hyperbolischen Kosinus.
SINH	Berechnet einen hyperbolischen Sinus.
TANH	Berechnet einen hyperbolischen Tangens.

## Andere numerische Funktionen

Weitere numerische Funktionen, die das BASIC der TI-70 Serie anbietet, sind nachfolgend angegeben.

ABS	Berechnet den Absolutwert.
EXP	Berechnet eine Potenz der Zahl e.
INT	Gibt den Ganzzahl-Anteil einer Zahl wieder.
LN	Berechnet den natürlichen Logarithmus.
LOG	Berechnet den Zehnerlogarithmus.
PI	Berechnet den Wert von pi als 3.141592654.
RANDOMIZE	Stellt sicher, daß eine Zufallszahl nicht vorhersehbar ist, indem dieses Statement einen Zufalls-Anfangswert für den Zufallszahlengenerator stellt.
RND	Gibt eine Zufallszahl wieder.
SGN	Überprüft eine Zahl, ob sie positiv, negativ oder Null ist.
SQR	Berechnet die Quadratwurzel.

## STRING-OPERATIONEN

Mit den String-Operationen Ihres Computers können Sie Dialoge und Nachrichten in Ihre Programme einbauen und Programme schreiben, die String-Informationen verarbeiten.

## string-Konstanten

Wenn Sie eine String-Konstante eintippen, so setzen Sie sie zwischen Anführungszeichen. Durch die Anführungszeichen wird der String identifiziert, die Zeichen werden jedoch nicht als Teil des Strings gelesen. Soll ein Anführungszeichen innerhalb einer String-Konstante vorkommen, tippen Sie zwei aufeinanderfolgende Anführungszeichen.

### Beispiel für String-Konstante

```
"Quote Test"  
"Quote""Test"  
""""Quote""""Test""""
```

### Darstellung beim Ausdruck

```
Quote Test  
Quote"Test  
"Quote""Test"
```

Bei Eingabe eines Null-Strings (der keine Zeichen enthält) tippen Sie lediglich das offene und geschlossene Anführungszeichen ohne Zeichen dazwischen.

## Verketten von Strings

Der Verkettungsoperator (&) verknüpft zwei Strings in einen, der allerdings nicht mehr als 255 Zeichen haben darf.

Folgender Programmabschnitt zeigt ein Beispiel für die Verkettung von Strings. (Beachten Sie, daß das erste Zeichen des Ausdrucks STRING2\$ eine Leerstelle ist und so der verkettete String an der richtigen Stelle steht.)

```
STRING1$="THIS IS A"  
STRING2$=" CONCATENATED STRING"  
STRING3$=STRING1$ & STRING2$  
PRINT STRING3$
```

Folgender Text wird angezeigt:

```
THIS IS A CONCATENATED STRING
```

Der Computer besitzt ebenfalls Vergleichs- und logische Operatoren, was den Vergleich von zwei Stringausdrücken möglich macht. Angaben zu diesen Operatoren finden Sie auf Seite 4-12.

## String-Funktionen

Folgende String-Funktionen berechnen einen String-Ausdruck und geben einen numerischen Wert wieder.

ASC	Wandelt das erste Zeichen eines Strings in den entsprechenden ASCII Zeichenkode um.
NUMERIC	Überprüft einen String-Ausdruck auf seine Eigenschaft als numerische Konstante.
VAL	Gibt den numerischen Wert eines String-Ausdrucks wieder.
LEN	Berechnet die Länge eines Strings.
POS	Untersucht einen String auf das Auftreten eines Unterstrings.

Folgende String-Funktionen berechnen einen numerischen Ausdruck und geben einen String-Wert wieder.

CHR\$	Wandelt einen ASCII Zeichenkode in das entsprechende Zeichen in der Anzeige um.
STR\$	Wandelt eine Zahl in einen String um.

Folgende String-Funktionen berechnen einen String-Ausdruck und geben einen String-Wert wieder.

RPT\$	Schafft einen neuen String durch mehrfache Wiederholung eines Anfangs-Strings.
SEG\$	Gibt einen Unterstring eines Strings wieder.

## DATENFELDER

Mit dem BASIC der TI-70 Serie sind ein-, zwei- und dreidimensionale Datenfelder möglich.

## definition der Größe eines Datenfeldes

Ein Datenfeld erfordert Speicherplatz. Sie können ein Datenfeld mit dem DIM Statement definieren oder den vom Computer reservierten Speicherplatz akzeptieren.

DIM	Definiert die Anzahl der Dimensionen sowie die Anzahl der Elemente jeder Dimension eines Datenfeldes.
-----	---

Wurde ein Datenfeld nicht durch ein DIM Statement spezifiziert, wählt der Computer automatisch eine Datenfeldgröße von 11 Elementen (numerierte von 0 bis 10) für jede Dimension im Datenfeld.

Benutzen Sie z.B. folgendes Statement, ohne vorher das Datenfeld SOMENAME dimensioniert zu haben, wird das Datenfeld auf zwei Dimensionen spezifiziert mit Speicherplatz für je 11 Elemente.

```
SOMENAME(0,0)=123
```

Bezieht sich Ihr Programm auf ein Datenfeldelement außerhalb des für das Datenfeld festgesetzten Bereiches, gibt der Computer eine Fehlermeldung wieder. Geben Sie eine Bruchzahl als Indexwert ein, rundet der Computer die Zahl; geben Sie jedoch eine negative Ganzzahl ein, erhalten Sie eine Fehlermeldung.

## VERGLEICHS- UND LOGISCHE OPERATOREN

Ein Vergleichsausdruck vergleicht zwei numerische oder String-Werte und überprüft, ob eine spezifizierte Beziehung richtig oder falsch ist. Ein logischer Ausdruck verbindet zwei Vergleichsausdrücke.

## Vergleichsoperatoren

Folgende Tabelle zeigt Beispiele mit den sechs Vergleichsoperatoren. In jedem Beispiel vergleicht der Operator die Werte A und B, die entweder numerische oder String-Ausdrücke darstellen.

A < B	Richtig, wenn A kleiner B ist.
A <= B	Richtig, wenn A kleiner oder gleich B ist.
A > B	Richtig, wenn A größer B ist.
A >= B	Richtig, wenn A größer oder gleich B ist.
A = B	Richtig, wenn A gleich B ist.
A <> B	Richtig, wenn A ungleich B ist.

**Anmerkung:** In den meisten Programmen zählt nur, daß eine Bedingung richtig oder falsch ist. In manchen Programmen kann es jedoch von Vorteil sein, daß ein Vergleichsausdruck den Wert -1 hat, wenn die Bedingung richtig ist oder den Wert 0 hat, wenn die Bedingung falsch ist.

## String-Vergleiche

Vergleicht ein Vergleichsausdruck zwei Stringwerte, so nimmt der Computer gleichzeitig ein Zeichen aus jedem String und vergleicht ihre ASCII Codes. Führende und abschließende Leerstellen werden mit berücksichtigt. (Eine Liste der ASCII Codes finden Sie in Anhang B.)

- Unterscheiden sich die ASCII Codes, so ist der String mit dem niedrigeren Code kleiner als der String mit dem höheren Code.
- Stimmen alle ASCII Codes überein und haben die gleiche Länge, sind die Strings gleich.
- Ist einer der Strings länger, wird der Vergleich für alle Zeichen, die im kürzeren String enthalten sind, ausgeführt. Sind alle ASCII Codes gleich, so gilt der längere String als der größere.
- Der Null-String ("") ist kleiner als jeder andere String.

## Logische Operatoren

Folgende Tabelle zeigt Beispiele mit den vier logischen Operatoren.

A AND B	Nur richtig, wenn sowohl A als auch B richtig sind.
---------	---

A OR B	Richtig, wenn entweder A oder B richtig ist, oder wenn beide, A und B, richtig sind.
A XOR B	Richtig, wenn entweder A oder B richtig ist, aber falsch, wenn beide, A und B, richtig sind.
NOT A	Nur richtig, wenn A falsch ist.

Die logischen Operatoren können auch dazu benutzt werden, einzelne Bits eines numerischen Wertes zu verarbeiten. Nähere Angaben lesen Sie bitte in Anhang F nach.

## CONTROL STATEMENTS

Control Statements werden verwendet, um den Ablauf, nach dem Programm-Statements ausgeführt werden, zu steuern. Mit Control Statements können Sie bedingte oder nicht bedingte Verzweigungen oder Schleifen erzeugen.

### Nicht bedingte Verzweigung

Dieses Statement überträgt stets die Steuerung auf eine bestimmte Zeilennummer.

GOTO	Überträgt die Steuerung des Programmablaufs auf eine einzige bestimmte Zeilennummer.
------	--

### Bedingte Verzweigung

Hier wird es ermöglicht, einen von mehreren möglichen Wegen der Programmsteuerung auszuwählen, je nach bestimmten Bedingungen innerhalb des Programms.

IF/THEN/ELSE	Verwendet Vergleichs- und logische Operatoren, um eine Bedingung in einem Programm zu überprüfen und zu entscheiden, welche Statements ausgeführt werden.
--------------	---

ON GOTO	Überträgt je nach Wert eines numerischen Ausdrucks die Programmsteuerung auf eine von mehreren Zeilennummern.
---------	---

ON GOSUB

Überträgt je nach Wert eines numerischen Ausdrucks die Programmsteuerung einem von mehreren Teilprogrammen.

## Schleifen

Eine FOR/NEXT Schleife wiederholt die Statements in der Schleife eine bestimmte Anzahl von Durchläufen. Mit folgenden Befehlswörtern können Sie eine FOR/NEXT Schleife bilden.

FOR/TO/STEP

Bezeichnet den Beginn einer FOR/NEXT Schleife.

NEXT

Bezeichnet das Ende einer FOR/NEXT Schleife. Wird das NEXT Statement ausgeführt, geht die Programmsteuerung auf das dem FOR/TO/STEP unmittelbar folgende Statement zurück.

## TEIL- UND UNTERPROGRAMME

Da Ihre Programme immer komplizierter werden, werden Sie feststellen, daß Sie dieselbe Gruppe von Zeilen an verschiedenen Stellen benötigen. Anstatt nun diese Zeilen jedesmal wieder neu zu schreiben, ist es wesentlich einfacher, sie als Teil- oder Unterprogramm einzugeben. Ein Teil- oder Unterprogramm kann an jede Stelle eines Programms eingebaut werden.

### Anwendung eines Teilprogramms

Folgende Befehlswörter werden für Teilprogramme verwendet.

GOSUB

Überträgt die Programmsteuerung vom Hauptprogramm auf ein Teilprogramm, das bei einer bestimmten Zeilennummer beginnt.

ON GOSUB

Überträgt basierend auf dem Wert eines numerischen Ausdrucks die Programmsteuerung auf eines von mehreren Teilprogrammen.

RETURN

Bezeichnet das Ende des Teilprogrammes und überträgt die Programmsteuerung wieder dem

Statement, das auf GOSUB oder ON GOSUB folgt.

Es ist allgemein üblich, Teilprogramme nach der Hauptsequenz eines Programmes zu setzen, es ist jedoch unzulässig, ein Teilprogramm nach dem END Statement aufzurufen.

## Verwendung eines Unterprogrammes

Mit folgenden Befehlswörtern wird ein Unterprogramm definiert und aufgerufen.

CALL

Überträgt die Steuerung des Programmablaufs vom Hauptprogramm auf einen bestimmten Unterprogrammnamen.

SUB

Bezeichnet den Beginn eines Unterprogramms.

SUBEND

Bezeichnet das Ende eines Unterprogramms.

SUBEXIT

Beendet den Ablauf eines Unterprogramms. Sie verwenden dieses Statement, wenn Sie vor dem SUBEND Statement aus dem Unterprogramm herausgehen wollen.

Ein Unterprogramm muß stets nach dem letzten Statement des Hauptprogramms folgen.

## Eingebaute Unterprogramme

Der Computer besitzt sechs eingebaute Unterprogramme – ADDMEM, ERR, GET, IO, KEY und PUT. Zugang zu diesen Unterprogrammen haben Sie über das CALL Statement. Verwenden Sie z.B. CALL ADDMEM, wenn Sie das ADDMEM Unterprogramm benutzen wollen.



## FEHLERVERARBEITUNG

Der Computer hat verschiedene Statements, mit deren Hilfe man Fehler im Programm ausfindig machen kann. Mit einigen Statements können Sie das Programm austesten, mit anderen das Programm auf alle Fehler untersuchen, die während des Programmablaufs auftreten können.

### Austesten eines Programms

Zum Austesten eines Programms können Sie folgende Befehlswörter zur Unterbrechung des Programms verwenden, so daß Sie den Wert der Variablen im Programm testen können.

BREAK	Setzt Stoppstellen innerhalb eines Programms.  <b>Anmerkung:</b> Sie können die (BREAK) Taste auch zur Unterbrechung des Programmablaufs drücken.
CON	Startet den Programmablauf nach einer Stoppstelle erneut.
ON BREAK	Läßt Ihnen die Wahl der Reaktion nach Auftreten einer Stoppstelle.
UNBREAK	Hebt die mit dem BREAK Statement gesetzten Stoppstellen wieder auf.

### Verarbeitung von Fehlern und Warnungen

Bei Auftreten von Fehlern oder Warnungen kann ein Programm oftmals diese Probleme beheben und verarbeiten, bevor sie in den Programmablauf eingreifen.

ON ERROR	Bietet Ihnen die Möglichkeit, Maßnahmen zu ergreifen, wenn ein Fehler während des Programmablaufs auftritt.
ON WARNING	Bietet Ihnen die Möglichkeit, Maßnahmen zu ergreifen, wenn eine Warnung während des Programmablaufs auftritt.

## ABSPEICHERN UND AUFRUF VON PROGRAMMEN

Wollen Sie eine Kopie Ihres Programmes für künftigen Gebrauch anlegen, so können Sie es abspeichern auf ein Peripheriegerät wie z.B. einen Kassettenrekorder oder ein RAM Modul. Nachdem Sie das Programm abgespeichert haben, können Sie es jederzeit wieder aufrufen und in den Computer laden. (Ausführliche Hinweise zur Benutzung von Kassetten finden Sie in Kapitel 6 dieses Buches.)

### Abspeichern eines Programms

Mit Hilfe folgender Befehlswörter können Sie eine Kopie des gerade im Speicher befindlichen Programms anlegen.

FORMAT	Initialisiert das im externen Speichergerät enthaltene Speichermedium. (Kassetten brauchen nicht formatiert zu werden.)  <b>Achtung:</b> Wird ein Medium, das bereits gespeicherte Programme enthält, neu formatiert, werden diese gelöscht.
SAVE	Kopiert das Programm auf ein externes Speichergerät, wie z.B. einen Kassettenrekorder.
VERIFY	Überprüft die auf das externe Gerät gespeicherte Kopie und stellt so sicher, daß ordnungsgemäß aufgenommen wurde.
PUT	Kopiert das Programm auf ein RAM Modul! (Da es sich um ein eingebautes Unterprogramm handelt, verwenden Sie CALL PUT.)

### Aufruf eines Programms

Mit folgenden Befehlswörtern können Sie eine Kopie eines auf ein externes Speichergerät gespeicherten Programmes wieder aufrufen.

**Achtung:** Wenn Sie ein Programm von einem externen Gerät aufrufen, wird jedes Programm, das gerade im Speicher des Computers abgelegt

ist, gelöscht. Wollen Sie das momentane Programm schützen, speichern Sie es, bevor Sie das neue Programm aufrufen.

**OLD** Ruft ein Programm aus einem externen Speichergerät wie z.B. einem Kassettenrekorder auf.

**GET** Ruft ein Programm aus einem RAM Modul auf. (Da es sich um ein eingebautes Unterprogramm handelt, verwenden Sie CALL GET.)

### ABSPEICHERN UND AUFRUF VON DATENDATEIEN

Wird eine große Anzahl von Daten in einem Programm verarbeitet, so ist es normalerweise zweckdienlich, die Daten auf ein externes Speichergerät zu übertragen, wie z.B. einen Kassettenrekorder.

#### Benennung einer Datendatei

Der gültige Name für eine in einem externen Speichergerät eröffnete Datei hängt von der Beschaffenheit des Peripheriegeräts ab. Bei Dateien auf Kassette kann eine Datei aus 1 bis 18 Zeichen bestehen. Der Name muß mit einem Buchstaben beginnen, der restliche Name kann aus beliebigen Zeichen mit Ausnahme von Komma oder Punkt bestehen. Dateinamen für Kassette können eine .NM Erweiterung beinhalten, um Nachrichten wegzulassen bzw. zu übergehen.

#### Verwendung von Datendateien

Mit folgenden Befehlswörtern können Sie Datensätze speichern und aufrufen. Sie können Datendateien auch löschen.

**FORMAT** Initialisiert das Speichermedium im externen Speichergerät. (Kassetten brauchen nicht formatiert zu werden.)

**Achtung:** Wird ein Medium, das bereits gespeicherte Programme enthält, neu formatiert, werden diese gelöscht.

**OPEN #** Stellt die Kommunikation zwischen dem Computer und dem externen Speichergerät her.

**CLOSE #** Hebt die Kommunikation zwischen Computer und externen Speichergerät auf.

**PRINT#** Speichert einen Datensatz auf eine eröffnete Datendatei.

**INPUT #** Ruft einen Datensatz aus einer eröffneten Datendatei auf.

**EOF** Überprüft die Datendatei, ob noch Datensätze in der Datei verblieben sind.

**RESTORE** Wählt den nächsten auszuführenden Datensatz aus. (RESTORE kann nicht in Verbindung mit einem Kassettenrekorder benutzt werden.)

**DELETE** Löscht eine bestimmte Datendatei aus einem externen Speichergerät.

### SENDEN VON INFORMATIONEN AN EINEN DRUCKER

Der Anschluß eines geeigneten Druckers ermöglicht es Ihnen, einen Ausdruck des gerade im Speicher befindlichen Programmes zu übertragen. Sie können auch Bemerkungen in Ihr Programm einbauen und so Informationen an den Drucker senden.

#### Auflisten eines Programms

Mit einem einzigen Befehlswort können Sie ein Programm auflisten. Zum Ausdrucken müssen Sie Ihren Drucker weder mit OPEN noch mit CLOSE ein- oder ausschalten.

**LIST** "Gerätename" führt den Ausdruck eines Programms Zeile für Zeile durch. LIST muß als Befehl verwendet werden; es kann nicht in ein Programm eingebaut werden.

### Druckeraufruf innerhalb eines Programms

Folgende Befehlswörter können als Statements innerhalb eines Programms verwendet werden.

OPEN #	Stellt eine Verbindung zwischen Computer und Drucker her.
CLOSE #	Hebt die Verbindung zwischen Computer und Drucker auf.
PRINT #	Schickt eine Zahl oder einen String an den Drucker.
IO	Führt zusätzliche Kontrolloperationen durch, die nicht in das BASIC der TI-70 Serie eingebaut sind. (Da IO ein eingebautes Unterprogramm ist, verwenden Sie CALL IO.) Die mit dem IO Unterprogramm möglichen Operationen hängen von der Art des Druckers ab.

Informationen über die Verwendung des PC-324 Thermodruckers lesen Sie bitte in Kapitel 6 dieses Handbuchs unter "Druckerkontrolle durch BASIC" nach.